



Swansea University
Prifysgol Abertawe



Swansea University E-Theses

Visual Analysis of Large, Time-Dependent, Multi-Dimensional Smart Sensor Tracking Data

Walker, James

How to cite:

Walker, James (2017) *Visual Analysis of Large, Time-Dependent, Multi-Dimensional Smart Sensor Tracking Data*.

Doctoral thesis, Swansea University.

<http://cronfa.swan.ac.uk/Record/cronfa36342>

Use policy:

This item is brought to you by Swansea University. Any person downloading material is agreeing to abide by the terms of the repository licence: copies of full text items may be used or reproduced in any format or medium, without prior permission for personal research or study, educational or non-commercial purposes only. The copyright for any work remains with the original author unless otherwise specified. The full-text must not be sold in any format or medium without the formal permission of the copyright holder. Permission for multiple reproductions should be obtained from the original author.

Authors are personally responsible for adhering to copyright and publisher restrictions when uploading content to the repository.

Please link to the metadata record in the Swansea University repository, Cronfa (link given in the citation reference above.)

<http://www.swansea.ac.uk/library/researchsupport/ris-support/>

Visual Analysis of Large, Time-Dependent, Multi-Dimensional Smart Sensor Tracking Data

James S. Walker

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Doctor of Philosophy.



Swansea University
Prifysgol Abertawe

Department of Computer Science
Swansea University

2017

Redacted Version of Original

Uploaded October 2017

A selection of third party content is redacted or is partially redacted from this thesis.

Pages:

Figs. 2.10 & 2.11 page 31.

Figs. 2.12 & 2.13 page 32.

Declaration

This work has not previously been accepted in substance for any degree and is not being currently submitted for any degree.

Signed: (candidate)

Date:

Statement 1

This thesis is being submitted in partial fulfilment of the requirements for the Degree of Doctor of Philosophy.

Signed: (candidate)

Date:

Statement 2

This thesis is the result of my own independent work/investigation, except where otherwise stated. Other sources are specifically acknowledged by clear cross referencing to author, work, and pages using the bibliography/references. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure of this thesis and the degree examination as a whole.

Signed: (candidate)

Date:

Statement 3

I hereby give consent for my thesis to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed: (candidate)

Date:

ABSTRACT

Technological advancements over the past decade have increased our ability to collect data to previously unimaginable volumes [Kei02]. Understanding temporal patterns is key to gaining knowledge and insight. However, our capacity to store data now far exceeds the rate at which we are able to understand it [KKEM10]. This phenomenon has led to a growing need for advanced solutions to make sense and use of an ever-increasing data space. Abstract temporal data provides additional challenges in its, representation, size, and scalability, high dimensionality, and unique structure.

One instance of such temporal data is acquired from smart sensor tags attached to freely-roaming animals recording multiple parameters at infra-second rates which are becoming commonplace, and are transforming biologists understanding of the way wild animals behave. The excitement at the potential inherent in sophisticated tracking devices has, however, been limited by a lack of available software to advance research in the field. This thesis introduces methodologies to deal with the problem of the analysis of the large, multi-dimensional, time-dependent data acquired. Interpretation of such data is complex and currently limits the ability of biologists to realise the value of their recorded information.

We present several contributions to the field of time-series visualisation, that is, the visualisation of ordered collections of real value data attributes at successive points in time sampled at uniform time intervals. Traditionally, time-series graphs have been used for temporal data. However, screen resolution is small in comparison to the large information space commonplace today. In such cases, we can only render a proportion of the data. It is widely accepted that the effective interpretation of large temporal data sets requires advanced methods and interaction techniques. In this thesis, we address these issues to enhance the exploration, analysis, and presentation of time-series data for movement ecologists in their smart sensor data analysis.

The content of this thesis is split into two parts. In the first volume, we provide an overview of the relevant literature and state-of-the-art methodologies. In the second part, we introduce a research development of techniques which address particular challenges unsolved in the literature, emphasising on their application to solving challenging domain level tasks faced by movement ecologists using smart tag data.

Firstly, we comparatively evaluate existing methods for the visual inspection of time-series data, giving a graphical overview of each and classifying their ability to explore and interact with data. Analysis often involves identifying segments of a time-series where specific phenomena occur and comparing between time segments for interesting patterns which can be used to form, prove, or refute a hypothesis. After analysis, findings are communicated to a wider audience. Navigating and communicating through a large data space is an important task which is not fully supported by existing techniques. We propose new visualisations and other extensions to the existing approaches. We undertake and report an empirical study and a field study of our approach on smart sensor data.

The reality of researchers faced with perhaps 10 channels of data recorded at sub-second rates spanning several days, is it is time consuming and error-prone for the domain expert to manually decode behaviour. Primarily this is dependent on the manual inspection of multiple time-series graphs. Machine learning algorithms have been considered, but have been difficult to introduce because of the large numbers of training sets required and their low discriminating precision in practice. We introduce TimeClassifier, a visual analytic system for the classification of time-series data to assist in the labelling and understanding of smart sensor data. We deploy our system with biologists and report real-world case studies of its use.

Next, we encapsulate TimeClassifier into an all-encompassing software suite, Framework4, which operates on smart sensor data to determine the four key elements considered pivotal for movement analysis from such tags. The software transforms smart sensor data into (i) dead-reckoned movements, (ii) template-matched behaviours, (iii) dynamic body acceleration derived energetics and (iv) position-linked environmental data before outputting it all into a single file. Biologists are given a software suite which enables them to link

behavioural actions and environmental conditions across time and space through a user-friendly software protocol. Its use enhances the ability of biologists to derive meaningful data rapidly from complex data. We make this software publicly available via the world-wide web and report over 60 users world-wide, across 12 countries.

Traditional analytical methods for smart sensor data concentrate on either behaviour or power, even though, because behaviour is manifest by movement which requires energy, both are inextricably linked. We introduce a proof-of-concept visualisation, the g-sphere, which capitalises on a suite of different acceleration-derived metrics (coupled with behaviour derived using framework4) and allow multiple dimensions to be visualised simultaneously. This approach, which highlights patterns that are not readily accessible using traditional time-series visualisation, can be used to inform our understanding and identification of processes in areas as disparate as sports practice, disease and wild animal ecology.

Finally, we conclude our findings and outline fruitful areas for future work.

ACKNOWLEDGEMENTS

Firstly, I would like to give special thanks to my academic supervisors, Prof. Mark W. Jones, and Dr. Robert S. Laramée (Bob) who's expertise, support, and patience made this thesis possible. Bob provided my first experience into the world of academia, supervising my BSc dissertation. During this time I was deliberating pursuing a career in industry. Bob persuaded me to stay in the university and undertake an MRes in collaboration with local business, which proved to be the perfect compromise. Without his encouragement I would not have considered a post-graduate degree. However, I thoroughly enjoyed my experience, and had the taste for undertaking academic research. Mark showed promise in me and offered a position to stay in the university to continue my research undertaking a PhD. Mark taught me the true meaning of academic research and guided me along the right path through the highs and lows. To both of these people I owe a great amount of gratitude which I will hold with me for the rest of my career.

This project has been a collaboration between the Computer Science and Biology departments in Swansea University. I would like to thank all those involved, but truly there is not enough space or time to list every person individually. Particular thanks go to Prof. Rory Wilson for his creativity, willingness, and belief in our research. Dr. Emily Shepard, Dr. Owen R. Bidder, Gwendoline I. Wilson, and Hannah Williams provided us with endless amounts of insightful feedback and showcased our work to the wider movement ecological community, without this exposure we would not have understood or gained insight into the domain level challenges faced, for this I am thankful.

I would also like to thank my family who have always been behind me every step of the way. Their love and support has been relentless and truly heart-warming. Thank you Victoria, Julie, and Steve Walker. My partner, Charlotte Sharples, has endured the endless evenings glued to the computer, and sacrifices made to get to this position today. Her love and support has seen me through this thesis.

Finally, this research would not have been possible without the financial assistance of ESPRC and Swansea University who have supported me throughout my PhD.

PUBLICATIONS

The work reported in the thesis has led to the following publications:

- James S. Walker, Mark W. Jones, Robert S. Laramée, Owen R. Bidder, Hannah J. Williams, Rebecca Scott, Emily L.C. Shepard, and Rory P. Wilson. **TimeClassifier: a visual analytic system for the classification of multi-dimensional time series data.** *The Visual Computer* (2015): 1–12.
- Gwendoline Ixia Wilson, Mark D. Holton, James S. Walker, Mark W. Jones, Ed Grundy, Ian M. Davies, David Clarke, Adrian Luckman, Nick Russill, Vianney Wilson, Rosie Plummer, and Rory P. Wilson. **A new perspective on how humans assess their surroundings; derivation of head orientation and its role in framing the environment.** *PeerJ* 3 (2015): e908.
- Owen R. Bidder, James S. Walker, Mark W. Jones, Mark D. Holton, Patricia Urg, Nikki M. Marks, Elizabeth A. Magowan, Iain E. Maguire, Rory P. Wilson. **Step by step: reconstruction of terrestrial animal movement paths by dead-reckoning.** *Movement ecology*, vol. 3, no. 1 (2015).
- James S. Walker, Mark W. Jones, Robert S. Laramée, Mark D. Holton, Emily LC Shepard, Hannah J. Williams, D. Michael D. Scantlebury, Nikki, J. Marks, Elizabeth A. Magowan, Iain E. Maguire, Owen R. Bidder, Agustina Di Virgilio and Rory P. Wilson. **Prying into the intimate secrets of animal lives; software beyond hardware for comprehensive annotation in Daily Diary tags.** *Movement ecology*, vol. 3, no. 1 (2015).
- Gwendoline Ixia Wilson, Brad Norman, James S. Walker, Hannah J. Williams, M. D. Holton, D. Clarke, and Rory P. Wilson. **In search of rules behind environmental framing; the case of head pitch.** *Movement ecology*, vol. 3, no. 1 (2015).
- James S. Walker, Rita Borgo, and Mark W. Jones. **TimeNotes: A Study on Effective Chart Visualisation and Interaction Techniques for Time-Series Data.** *IEEE Transactions on Visualisation and Computer Graphics (Proceedings of Information Visualisation 2015)*, vol. 22, no. 01, pp. 549–558, Jan. 2016
- Rory P. Wilson, Mark D. Holton, James S. Walker, Emily L. C. Shepard, D. Mike Scantlebury, Vianney L. Wilson, Gwendoline I. Wilson, Brenda Tysse, Rebecca Cliffe, Mike Gravenor, Javier Ciancio, Melitta McNarry, Kelly Mackintosh, Lama Qasem, Frank Rosell, Patricia Graf, Mark W. Jones. **A spherical-plot solution to linking acceleration metrics with animal performance, state, behaviour and lifestyle.** *Movement ecology*, vol. 4, no. 1 (2016)

CONTENTS

I	Background and Motivation	3
1	Introduction and Motivation	5
1.1	Information Visualisation	6
1.1.1	Time-Oriented Data	7
1.1.2	Brief History of Temporal Visualisation	9
1.2	Smart Sensor Technology	10
1.2.1	Ecology Work-flows	12
1.2.2	Visualisation of Smart Sensor Data	16
1.3	Research Challenges	17
1.4	Contributions	18
1.5	Thesis Structure	20
2	A Survey of Visualisation Techniques for Temporal Knowledge Discovery	21
2.1	Introduction and Motivation	21
2.1.1	Contributions	22
2.2	Knowledge Discovery	22
2.3	Data Mining	24
2.4	Information Visualisation	27
2.4.1	Layout Based	28
2.4.2	Histogram Based	29
2.4.3	Lens Based	30
2.5	Visual Analytics	32
2.6	Literature Classification	33
2.6.1	Classification	33
2.6.2	Clustering	35
2.6.3	Indexing and Retrieval	36
2.6.4	Pattern Discovery	39
2.6.5	Prediction	44
2.6.6	Segmentation	45
2.7	Discussion and Summary	47
II	Research Development	49
3	A Study on Effective Chart Visualisation and Interaction	51
3.1	Introduction	51
3.2	Background	52
3.3	Tasks and Design	53
3.3.1	Task-Based Evaluation	53
3.3.2	System Development	55
3.4	TimeNotes	57
3.4.1	Layout	59
3.4.2	Node Interaction and Rendering	60
3.4.3	Overlay	60
3.4.4	Annotation	61
3.5	User Study	61
3.5.1	Tasks and Stimula Design	62
3.5.2	Study Hypothesis	63
3.5.3	Study Analysis	64
3.5.4	Experimental Setting	64
3.5.5	Analysis Results	65

3.5.6	User Study Discussion	67
3.6	Field Study Discussion	67
3.7	Summary	68
4	Visual Analytics for Multi-Dimensional Time-Series Classification	70
4.1	Introduction	70
4.2	Background	71
4.3	Precision and Recall	72
4.4	Tasks and Design	73
4.4.1	Problem Statement	73
4.4.2	Domain Requirements	74
4.4.3	Tasks	74
4.4.4	Algorithm	74
4.4.5	Interface Design	78
4.5	Evaluation	83
4.5.1	Case Study 1 - Turtles	83
4.5.2	Case Study 2 - Soaring Birds	84
4.5.3	Formal User-Study	85
4.5.4	Domain Expert User Study	85
4.5.5	Evaluation Against Traditional Data Mining	86
4.6	Discussion	87
4.7	Summary	88
5	Software Beyond Hardware for Comprehensive Smart Sensor Annotation	89
5.1	Introduction	89
5.2	Background	91
5.3	Implementation	92
5.4	Framework4	92
5.4.1	Loading Data	92
5.4.2	Derivation of Animal Trajectory by Dead-Reckoning	93
5.4.3	Derivation of Animal Behaviour	107
5.4.4	Derivation of Animal Energy Expenditure	107
5.4.5	Derivation of the Physical Characteristics of the Environment	109
5.4.6	Exporting data	109
5.5	Evaluation	109
5.5.1	Software System	109
5.5.2	Impact of Framework4	110
5.5.3	Usage and Statistics	112
5.6	Summary	114
6	Visualisation for Linking Behaviour and Energetics for Diagnostics	116
6.1	Introduction	116
6.2	Background	118
6.3	Online Methods	118
6.3.1	Data Dimensions	119
6.3.2	Raw Plot	120
6.3.3	Spherical Plot	120
6.3.4	Binning in Three-Dimensions	121
6.3.5	Spherical Histogram	123
6.3.6	G-Urchin	123
6.3.7	Differential G-Sphere	125
6.4	User interface	127
6.4.1	Data G-Sphere Construction	127
6.4.2	Operation G-Sphere Construction	130
6.5	Results	130
6.5.1	The Basic G-Sphere	130
6.5.2	The Spherical Histogram Plot	131
6.5.3	The G-Urchin	132
6.5.4	The Differential G-Urchin	133
6.5.5	Contrasting Patterns of Acceleration of Different Body Parts	133
6.6	Discussion	133

6.7 Summary	137
7 Conclusion and Future Work	138
7.1 Conclusion	138
7.2 Future Work - Smart Sensor Data Visualisation	141
7.3 Future Work - Time-Series Visualisation	142
Appendices	145
A User Study Training Material	147

LIST OF FIGURES

1.1	The visual analytics process pipeline [KKEM10]. The human engages with analysis models through interaction visualisation. The user refines the parameters of models through combining visualisation and user knowledge in an iterative user-in-the-loop process.	7
1.2	This visualisation illustrates the traditional time-series graph (left) as conceived by Playfair in 1786 [Pla86]. A linear mapping from the temporal domain to the time (x) axis is applied (right). This image utilises the mapping notation presented by Isenberg et al. [IBDF11].	8
1.3	This visualisation illustrates Playfair's chart of exports and imports to and from Denmark and Norway, overlaid with that of England. The result shows a clear correlation between them. Image Credit: [Pla86]	9
1.4	This Infographic illustrates Charles Joseph Minard's depiction of the Napoleon French invasion. The gold line identifies the armies advance through Lithuania to Russia, while the black line follows the retreat. The width of the line highlights the devastating loss of life throughout the stages of the journey. Image Credit: Charles Joseph Minard [Cor12]	10
1.5	An accelerometer can be conceptually considered as an object suspended in-between two springs. When the object is subject to acceleration it is shifted. The displacement measured is proportional to the acceleration due to gravity (g -force).	11
1.6	A leatherback turtle performing a rolling manoeuvre through 90 degree rotations in the heave axis, and 180 degree rotations in the sway axis. Image Credit: Shepard et al. [SH08]	12
1.7	A conceptualisation of the biologists work-flows which we have observed over the duration of our collaboration.	13
1.8	Commercial graphing software utilised by the movement ecologists at the SLAM research centre to explore, analyse, and present their smart sensor data. Image Credit: Jensen [Sys].	15
1.9	A visualisation of dead reckoned animal movement trajectories representing the kinematic patterns from Whales in the wild. Image credit: Ware et al. [WAPW06]	16
2.1	Clustering (left) is used to identify natural groupings of similar data items. The data in the image is partitioned into three groups (as represented by the coloured regions). Classification (centre) uses a set of pre-defined classes (partitioned in the image) to determine the resulting classification of an unlabelled input (red circle). Indexing and retrieval (right) is the task of querying the data with an input sequence (red circle) to retrieve similar matches from a corpus of indexed sequences (non-red circles).	22
2.2	Prediction (left) is used to forecast future data values based on previous historical data. Pattern discovery (centre) is the task of finding interesting patterns, such as, anomalies (grey), and repetitive patterns (yellow). Segmentation (right) is used to divide the data into discrete segments (represented by the black vertical lines throughout the series).	23
2.3	Distance measures are used to quantify the similarity between two data series (in this case, the top and bottom time-series graphs). Euclidean distance (left) calculates the sum of the absolute difference between adjacent data items. Dynamic time warping (right), stretches and compresses time to identify related items which vary in temporal duration. Image Credit: [KR05a]	25
2.4	Zoom plot and temporal to position map. The zoomed in view-port is displayed as a linear function on the map of the visible data range.	28

2.5	Overview and detail plot [PCS95] and temporal to position map. The overview is represented as a linear mapping of the entire data range. The detail is represented as a disjoint region overlaid on the map for the data range selected.	29
2.6	Stack Zoom [JE10] and temporal to position map. Each node in the hierarchy is represented as a disjoint region overlaid on the map for the data range occupied.	29
2.7	Pixel Plot [KL06] and linear temporal to position map.	30
2.8	Two-dimensional Pixel Plot [HDKS07] and linear temporal to position map.	30
2.9	River Plot [BPS ⁺ 07] and linear temporal to position map.	30
2.10	SignalLens [Kin10] and temporal to position map. The temporal to position map represents this region as a higher linear gradient.	31
2.11	Smoothed SignalLens [Kin10] and temporal to position map. The temporal to position map represents this region as a higher linear gradient.	31
2.12	RiverLens and temporal to position map. The zoomed in region features a lower-density of data items per pixel. The temporal to position map represents this region as a higher linear gradient.	32
2.13	ChronoLenses [ZCPB11] and temporal to position map. Each lens is represented as a disjoint region overlaid on the map for the data range occupied.	32
2.14	The VAET system highlights interesting events in e-transaction data. Initially all data items are displayed, as the user highlights interesting transactions in the visualisation the relevant information space utilised is adjusted accordingly. An overview of saliency values are encoded in a pixel-based display. A knot view encodes each transaction, with lines draw between to represent clusterings. Image Credit: [XCH ⁺ 14]	34
2.15	VizTree provides a graphical overview of time-series. A suffix tree encodes local and global structure with branch width representing amplitude frequency, chained together to show temporal trends. Image Credit: [LKL05]	40
2.16	TimeSeer utilises scagnostics to characterise scatter plots of data attributes at each time index. The statistical summaries allow the user to explore pairs of variables to identify anomalies in the data. Image Credit: [DAW13]	41
2.17	OpinionFlow depicts social media trends through a river type visual metaphor. Divergence and convergence in the flow illustrate how opinions propagate over time. This instance depicts opinion diffusion on the Xbox when the Xbox One was announced. Image Credit: [WLY ⁺ 14]	42
2.18	The TiMoVA system enables comparison of prediction models. The user is guided through selecting appropriate models and parameters using several visual aides. Image Credit: [BAF ⁺ 15]	43
2.19	This visualisation supports the user in understanding the influence of parameter selection on resulting segmentations for activity recognition. Vertically aligned strips encode parameter settings with the respective segmentation results shown alongside. Compound views provide statistical summaries. Image Credit: [RLK ⁺ 15]	45
3.1	The embedded zoom visualisation is constructed using multiple zoom handles (vertical lines) which allow the user to interactively define time-axis distortions to the time-series graph. Snapshots of the visualisation during exploration and analysis are shown below the main display for later access during presentation and reporting.	56
3.2	The stack lens visualisation is a layout based method comprising of multiple layers of time-series graphs. Zoom handles span between adjacent layers to define the time-axis distortion. The start of each handle defines time index it represents, and the end of the corresponding handle defines the position to map the corresponding temporal position to on the display. Layers create a natural graphical history of interaction with the visualisation.	57
3.3	Layout techniques commonly applied when visualising hierarchical data. Node link diagram (left), space filling tree map (center), and a space filling layout as used in the Stack Zoom technique (right)	58

3.4	TimeNotes displaying data from a Condor, one of the largest birds of prey. The Condor flies (flapping its wings - left tree) to find thermals. The Condor glides in in a circular motion in the thermal to gain altitude (central tree heading is plotted, showing repeated cycles through 360 degrees). The Condor rises in altitude then glides through it to conserve energy. Once the bird loses altitude it flies to the next thermal and repeats (right tree).	59
3.5	This figure shows two zoom regions overlapped. A semi-transparent blue smooth connection is drawn between parent and child nodes to enable temporal positioning to be perceived even when data bounds overlap. Dragging and repositioning the temporal position bounds on the parent node (top) allow the user to modify the data bounds visualised in each child.	60
3.6	Intermediate node removed in the TimeNotes display (left). All child nodes are shifted upwards to the superior node (right).	61
3.7	In this figure we illustrate the overlay functionality of TimeNotes. Five nodes are snapped together to form an overlaid time-series graph view. A unique colour is applied to each line which corresponds to the connection drawn to the parent node to perceive the temporal location of each line. As we can see, the selected temporal sequences contain a low variance in both frequency and amplitude across time.	62
3.8	Analysis of performance results for Tasks A, D and E, (mean, median) values are indicated below each bar. Error bars show 95% confidence intervals. . .	63
3.9	Analysis of performance results for Tasks B and C, (mean, median) values are indicated below each bar. Error bars show 95% confidence intervals. . .	64
4.1	This image shows an early prototype of the TimeClassifier user interface. Initially the software comprised of a focus and context display which only allowed users to search for similar patterns across the series. There was no facility to validate classified behaviour.	72
4.2	This figure shows the classification workflow typically utilised when performing supervised learning. A model (classifier) learns from extensively labelled data to generalise relationships. The model operates on new unseen data to classify behaviour.	73
4.3	This figure shows the main user interface of TimeClassifier operating on data from a deployment on an Imperial Cormorant. Top: A stacked time-series graph of the whole data set with overlaid coloured regions illustrating labelled animal behaviour. The user can apply zooming to see more detail. Bottom-left: The results widget is used for searching for behaviour instances and displaying results. Bottom-right: The classification widget encapsulates classified results. Each tab represents a classification group with each classified behaviour instance represented using embedded time-series and overlaid plot views. . .	79
4.4	This figure shows our proposed system work-flow. Firstly, the user provides one instance of a behaviour from the time-series or from our template database. The classification algorithm then finds all instances of specified behaviour at the selected frequencies on the unlabelled data. The results are presented to the user in the results view. Using our feedback loop, precision and recall can be boosted by providing secondary examples to feed into the classifier or by manually accepting / rejecting matches. Accepted behaviours are moved to the classification panel. The user can manually label data samples to further improve precision and recall is necessary.	80
4.5	This figure shows the classification wizard. (a) illustrates the wizard parameters for dynamically adjusting the threshold. (b) shows the density based visualisations to gain an understanding of where matches occur in the data series. (c) shows the overlaid signals visualisation of all the extracted matches in a stacked graph format, with one graph for each attribute of the template. The template signal is overlaid in red to show a direct comparison. A yellow to blue colour scheme is used, yellow representing low similarity matches, while blue encodes high similarity matches.	81
4.6	The overlaid plot view overlays the classified instances in a time-series graph. The user can accept or reject results by rubber-band selection (yellow) on the time-series.	82

4.7	This figure shows a subset of approximately 1 hour of labelled ascent (green), descent (blue), swimming (black), and burst swimming (yellow) patterns extracted from data obtained from a Magellanic Penguin by a domain expert using TimeClassifier.	84
4.8	This figure shows the three graphs for each of our participants performing behaviour labelling using TimeClassifier. We report the f-score (a combination of precision and recall) achieved by each participant over time, as they label each of the behaviour groupings of swimming, burst swimming, descent and ascent. The X axis encodes time, while the Y axis encodes the F-Score percentage. Line colour corresponds to the F-Score of the specified behaviour detailed in the graph legend.	86
5.1	A spreadsheet developed by movement ecologists in the SLAM research group for deriving animal trajectories. The spreadsheet consists of several configuration parameters. The coupling of how the configuration maps to the resulting trajectory is not clear.	91
5.2	This figure shows the Framework4 analysis pipeline. The user inputs their sensor data in CSV or tab delimited format. After this, the user applies any combination of our behaviour analysis, energy expenditure, trajectory and environmental condition derivation tool kits. Finally, the data is exported into a unified CSV file with the derived data channels time aligned with the original input data.	92
5.3	The data import wizard provides an intuitive, but flexible user interface for importing data from CSV and tabular delimiter file formats. The user specifies the start row of the data, along with the file delimiter. A preview of the imported data is shown in a tabular layout, the user can rename headers using a pre-set or by extracting them from the imported file. The specified attribute names are used to reference the data channels in the software.	93
5.5	Framework4 performs dead reckoning in the following way. Firstly GPS data is imported. The data is synchronised by comparing the time-stamps with the original data file. The sensor coordinate systems are then aligned to ensure all sensors are registering the same frame of reference. Heading is derived by performing several processing stages of; calibration of the magnetometer sensors, computing pitch and roll, coordinate alignment, and finally heading derivation. After this, speed is derived. Finally, the heading and speed are combined to derive a dead reckoned trajectory. The dead-reckoning path is corrected using external positioning information to eliminate accumulated errors in the final trajectory obtained.	94
5.4	The earths magnetic field is constructed of field lines. A declination angle determines the offset between magnetic and geographic north. Image Credit: Addison Wesley Longman, inc.	94
5.6	The dead-reckoning wizard features a number of modular classification steps. These are; GPS import, alignment, heading, speed derivation via VeDBA or applying a constant, dead-reckoning, and dead-reckoning with GPS correction	95
5.7	The user selects the relevant time fields (i.e. day, hours, minutes, seconds) in both the geo-referenced and smart sensor data sources to synchronise them into one data set. A graphical preview of the geo-referenced data is shown in the bottom of the panel.	96
5.8	Since the accelerometer and magnetometer coordinate systems may not be aligned. We allow the user to adjust the data channels by a pre-defined transformation function for the Daily Diary system. The user selects the accelerometer and magnetometer data channels, along with the orientation of the tag during deployment. When adjusting the orientation, the image of the sensor (bottom) is updated to reflect the settings specified.	97
5.9	Rotating a magnetometer around 360 degrees should produce a perfect sphere centred on the origin (left). When hard iron effects are present a constant bias is present (centre), resulting in a displaced sphere from the origin. Soft iron deposits distort or stretch the field (right).	98

5.10	The user selects the magnetometer data channels to perform error correction on. A graphical preview of the corrected channels is shown in a stacked time-series graph. The user can view the correction matrices applied, along with exporting / importing them from other data sets where the initial calibration took place.	99
5.11	Tilt correction computes the pitch and roll of the device during deployment and reverses these offsets on the magnetometer data channel. The user specifies the accelerometer components, along with the smoothing window to use to compute orientation. A graphical preview of the pitch and roll are displayed in a stacked time-series graph for verification purposes.	100
5.12	The user selects the orientation and error corrected magnetometer components to compute the heading from. A deviation angle corresponding to the heading declination angle can be applied, to transform the heading from magnetic north to geographical north. A graphical preview of the derived heading is shown in time-series graph format for verification.	101
5.13	VeDBA is computed by selecting the accelerometer components and an appropriate smoothing rate to utilise. The user can select to perform behaviour based VeDBA computation, where VeDBA is only applied to estimate speed for specific behaviours derived in Framework4 (e.g. running). A graphical preview of the derived VeDBA channel is shown below the parameter settings in a time-series graph.	103
5.14	The dead reckoning step combines the derived speed and heading channels to generate a trajectory. The user specifies the speed and heading channels to utilise for this procedure. A start index in the data (default zero) is required since usually an initial portion of the data is used for calibration which is not desirable to include in the resulting path. A heading offset allows the user to apply an angle of declination such that the heading corresponds to true north. A speed coefficient and threshold amount adjusts the speed accordingly. The start latitude and longitude specify the initial position of the trajectory. Once compute is pressed the resulting path is shown in map below. The user can interactively refine parameters until the expected result is achieved.	104
5.15	Dead-reckoned track of a European badger (<i>Meles meles</i>) in Northern Ireland leaving its sleeping quarters (red dashed circle) and moving through the underground sett to emerge at the entrance (yellow circle). The vertical axis representing depth is shown as the pressure difference between the surface and any time underground. The reconstruction assumes that animal speed is directly proportional to VeDBA [BQW12] underground in the same way it is on the surface. If this is not the case, the derived distances will be affected accordingly.	105
5.16	The dead reckoning correction stage adjusts the dead reckoned trajectory with positional fixes to reset accumulated error along the path. The user specifies the path to perform correction on, along with the positional data to use as a ground truth to force the dead reckoned path through. The user can down-sample the positional data to apply the correction at set resolutions. The resulting paths are overlaid in a map view. The coloured paths are associated as follows: Blue for the GPS trajectory, red for the dead reckoned trajectory, and green for the dead reckoned corrected trajectory.	106
5.17	Illustration of the dead reckoning correction algorithm. A heading correction factor is first calculated and applied which adjusts the heading between two trajectory fixes obtained using the dead reckoning algorithm to those of a specified ground truth source (left). The distance correction factor is calculated and applied which adjusts the distance between two fixes to those of the specified truth source (right).	107
5.18	The dead-reckoned trajectory of a cow (<i>Bos taurus</i>) in a field in Northern Ireland over 2 hours, colour-coded according to different activities green = grazing, black = walking, red = lying down	108
5.19	The dead-reckoned trajectory of a sheep (<i>Ovis aries</i>) in Argentina over 14 hours showing how VeDBA, a metric that correlates linearly with metabolic rate, (ranging from pale green to yellow [low values] through to red [high values] varies with location and track tortuosity. Note how the twists and turns in the trajectory are generally linked to lower VeDBA (circled in black).	108

5.20	Two-dimensional paths of the human participant as determined by (a) video Recording, (b) GPS, (c) dead-reckoning without correction, (d) dead-reckoning with correction every 2 seconds, (e) dead-reckoning with correction every 5 seconds, (f) dead-reckoning with correction every 10 seconds.	110
5.21	Mean distance error (meters) at each of the ground-truthing regimes. Error bars represent standard deviation of the mean.	111
5.22	The HIPOP head mounted unit consists of a daily diary and GPS tag embedded into a cap to compute heading, head pitch, and movement trajectories. This allows an estimate of how the head is oriented to be quantified.	112
5.23	Incidence of stylised peaks (blue symbols), troughs (green symbols) and fixations (red symbols) in head pitch over time in three participants operating in three different environments: a green space (botanical garden), a blue space (snorkelling in an underwater environment) and finally a feature-less corridor. Sections of the graphs where there no data are shown correspond to periods where the changes in head pitch did not correspond to the identified stylised behaviour. The green space featured participants mostly orienting their head in a downward position as they investigated the surround environment and plants offered by the garden. The blue space features a large amplitude difference in head orientation (80 degrees), presumably as there were interesting features of the environment surrounding the individual. The corridor had the smallest amplitude difference. The corridor environment was extremely depauperate in visual (and other) stimuli while the blue and green environments were both rich in stimuli.	113
5.24	The home page of the Framework4 website. Users are presented with an textual summary of the four core attributes Framework4 allows users to derive. We also feature a video illustrating a one minute summary of the core features.	113
5.25	A visualisation of the usage of Framework4. We report over 60 users across 12 countries.	114
6.1	A spherical scatter plot visualisation of static acceleration of data acquired from a penguin spanning approximately 15 minutes of activity. The trajectory (mapped to the surface of the sphere) illustrates the orientation of the penguin over this duration. The trajectory is colour encoded to the y axis of acceleration. A model of a penguin is placed in the centre of the sphere, with the orientation vector pointing outward of the spine for reference. The visualisation shows the penguin rotating through a variety of orientations while swimming.	119
6.2	A 3-d scatter plot of static (orthogonal) tri-axial acceleration data recorded from an Imperial Cormorant. The time-span visualised contains diving, sitting on the water surface and flying behaviours. Visualising the raw values in three-dimensional makes it difficult to interpret and identify the underlying trends and patterns present.	120
6.3	A spherical coordinate visualisation of (a) postural state plotted onto the surface of a sphere in three-dimensional space. (b) Each point is joined together in chronological order to show the temporal structure of the data. (c) The data is projected outwards from the sphere associated with depth, this shows a compelling visualisation of orientation associated with depth. This visualisation is useful for small sub-sets of data. However, when large time-spans are visualised the visualisation can become occluded.	121
6.4	This image illustrates the creation of a close to uniform and regular sphere tessellation. (a) The initial icosahedron. (b - e) the icosahedron is repeatedly subdivided with each of the points projected onto a sphere on each iteration.	122
6.5	A spherical histogram visualisation. This visualisation depicts frequent orientation and postural states. Each column projecting from the sphere represents the sample size of data items contained in each corresponding facet. The longer each column, the more data items residing in the respective facet on the sphere, and therefore more time the animal spent in the specified posture (orientation). This visualisation is of a deployment from a cormorant. The animal spends the volume of the duration horizontal on the surface of the water (depicted by the tallest column originating directly through the z axis of the model). Variations of the orientation depict less frequent diving and flying patterns.	124

6.6	Left to right: Histogram, Frequency shape (stacked), fixed shape (skittle). . .	125
6.7	G-Urchin of skittle shape (top), and stacked frequency (bottom) urchins emitted from the centre of each facet of the sphere. Each urchin represents the underlying data within the corresponding facet. The size of each histogram bin is colour coded accordingly.	126
6.8	Overview of the user interface for a data g-sphere. Top: toolbar for constructing a new g-sphere. Left: visualisation panel of the g-sphere. Top right: histogram view of a selected g-urchin in the visualisation panel. Bottom right: interface to select specific behaviours and data subsets to utilise in the visualisation.	127
6.9	This image illustrates the work-flow when creating a new g-sphere. The user selects whether to create a data or operational g-sphere. The data g-sphere is computed directly from the underlying x , y , and z data channels selected by the user. The operational g-sphere requires the user to specify two existing g-spheres (operational or data) and an operation to perform between them. The g-spheres are aligned and then the operation is performed before creating the new resulting g-sphere.	128
6.10	The dialog for creating a new g-sphere. Left: the user has selected to create a data g-sphere. The x , y , and z axes must be selected from the data source. Right: the user has selected to create an operational g-sphere. Two g-spheres must be selected, along with the operation to perform between them.	128
6.11	A tabular widget of behaviour classifications in the data is shown in the bottom right of a data g-sphere window (Figure 6.8). The user can enable / disable whole behaviour groups, and individual behaviour instances to visualise. . . .	129
6.12	The g-sphere alignment dialog which allow the user to align two g-spheres together. The user selects the amount to rotate the left, and right g-spheres by using (a) the spin boxes on the right, or (b) the cursor to rotate the g-spheres. The user can select the visualise the histogram, or raw underlying data to assist in the alignment.	130
6.13	Example behavioural data from a cormorant. Six dives and a short period of flight are visualised by (a) a point- based g-sphere (with point colour equating with dynamic body acceleration). (b) shows the same data as (a) but as a histogram plot. Both (c) images depict urchin plots of (b); (C1) shows percentages of dynamic body acceleration allocation taken across the whole g-sphere while (C2) shows percentages amounting to 100% per facet. Note the higher values of dynamic body acceleration attributed to flight and descent of the water column, particularly emphasised by the 100% facet percentage. Note also how certain spines show multi-modes (e.g. white arrow) which can be indicative of different behaviours at one body attitude.	131
6.14	Examples of posture and energy-linked posture visualised for two contrasting species (a human and a fish) over 24 hours. The human data are taken from a person on a walking/camping tour while the fish data are from a hole-dwelling reef species that often rests by wedging itself at unusual angles. The left-hand figures (a) show spherical histogram plots, indicating how time is allocated to different body postures (the 'North pole' position shows the species in the 'normal' upright position). The first right-hand figure for each species (b) shows how each posture is linked to varying putative power levels. Note how the human has higher power-proxy levels associated with the vertical posture due to walking. Both the human and the fish have low power-proxy levels at low 'latitude' angles acquired during resting/sleep, exemplified by the large diameter blue discs. Data normalised to give a global percentage for all angles may hide infrequent, but higher-energy, activities. Normalising the data to 100% per facet (c) highlights these though. In this case, the low-energy life style of the fish is still apparent (B2), with higher energies occurring fleetingly and only when the fish is vertical (white arrow). The colour coding has blue as low, and red as high, values.	132

6.15	Example urchin plots for four consecutive 24 hour periods after the release of a European badger (wearing a collar-mounted accelerometer) following anaesthesia. The 'North pole' facets show when the animal was properly horizontal (i.e. in standing or walking posture). Note how the first two days show no high energy activity because the animal was either resting or asleep. The second day shows only four changes in position. By day three, higher energy, normal posture activities such as walking are apparent at the North pole. This process is further enhanced in day 4, with North pole spine dynamic body acceleration distributions having modes that have moved up the length of the spines to indicate higher power use. Dynamic body acceleration values are colour-coded with maximum values (in red) of 1 g	134
6.16	Example posture and dynamic body acceleration values associated with 'state' in humans. (a) shows two spherical histogram plots for a person walking after seeing 'happy' and 'sad' film clips. A third differential spherical histogram plot highlights the difference between the two situations (blue = a higher relative frequency of 'happy' points per facet while red = a higher relative frequency of 'sad' points per facet). Note how the two conditions are reflected in the postural changes (b) shows urchin plots for someone trekking across snow pulling a sledge one minute before a fall and one minute after recovering from the fall. The differential urchin shows both differences in postures adopted between the two situations as well as the dynamism of the walking (red shows a higher relative dynamic body acceleration frequency 'before the fall' while blue shows the reverse).	135
6.17	Acceleration signal patterns according to tag placement on the body. The figure shows (two upper visualisations) raw data from (a) a back-mounted tag synchronized with (b) a hand-mounted tag during the last 1.5 seconds of 10 basketball penalty shots. The two lower visualisations (c) and (d), show the corresponding urchin plots. The colour coding has blue as low, and red as high, values.	136

LIST OF TABLES

- 2.1 An overview and classification of visual analytics methods for abstract temporal data. The survey is grouped horizontally according to task-at-hand (classification, clustering, indexing and retrieval, pattern discovery, prediction, and segmentation). Each group is then vertically sub-divided by the number of variables (univariate vs. multivariate), and the type of data (event vs. state). We refer the reader to Section 1.1.1 for an explanation of these. The entries of each sub-group are placed in chronological order. 33

- 3.1 A comparison of the existing approaches with the interaction tasks they support in Shneiderman’s [Shn96] task taxonomy and presentation tasks from Aigner et al. [AMST11]. 54

- 4.1 Notation of the symbols used to detail our fast normalised cross-correlation method. 75
- 4.2 Evaluation of the normalised cross-correlation algorithm in respect to data size versus time taken to evaluate. 78
- 4.3 Table illustrating the time taken (in milliseconds), precision, and recall results for TimeClassifier to label Ascent, Descent, Swimming, and Burst Swimming behaviours on a data set obtained from a Penguin deployment. 86
- 4.4 Table comparing time (in seconds) along with average precision and recall scores for our TimeClassifier system results with state-of-the-art machine learning methods. We utilise the x , y , and z axes of acceleration as feature vectors for input into the algorithms. We specify the parameters for each machine learning method in the table. 1NN with DTW and hierarchical clustering with DTW are placed in italics since they required segmented data as input. Hierarchical clustering failed to return any meaningful results, we therefore place a star for the results of this algorithm. 87

LIST OF TERMINOLOGY AND NOTATION

- Acceleration** The rate of change of velocity of an object with respect to time.
- Accelerometer** A sensor which measures acceleration due to gravity (g-force) in respect to the earth's gravitational field.
- Analysis** A task which is executed to confirm or refute an hypothesis from data exploration by identifying evidence which supports the users proposition.
- Autoregressive Integrated Moving Average (ARIMA)** A prediction model which is fitted to a time-series to predict future data points.
- American Standard Code for Information Interchange (ASCII)** A character-encoding protocol which is used to represent text in a computer.
- Comma Separated Values (CSV)** A data format for storing data in a tabular form in plain text. Each line of the file is a data record, with each attribute separated by a comma.
- Central Processing Unit (CPU)** The hardware device which executes instructions from software.
- Daily Diary (DD)** A animal tracking device which consists of a particular combination of sensors within a tag. Specifically, the combination of tri-axial accelerometers, tri-axial magnetometers, and pressure and speed transducers together with environmental sensors such as temperature, light, and humidity.
- Data Mining (DM)** Automated algorithm(s) which automatically identify patterns and solve problems through data analysis.
- Decision Tree (DT)** A classification algorithm which generalise a set of rules from labelled input data, such that, rules are applied to unseen samples to obtain a classification.
- Dead Reckoning (DR)** A process for calculating position based on a previously known location and a corresponding heading and distance over the elapsed time.
- Distance Measure** Derives a quantitative value corresponding to the similarity or dissimilarity between two objects.
- Dimensionality Reduction** The process of reducing the number of variables under consideration into a lower dimensional form.
- Dynamic Acceleration** The change in velocity as a result of body motion providing information on animal movement.
- Dynamic Time Warping (DTW)** A distance measure for deriving similarity between two temporal series varying in speed.
- Energy Expenditure** The amount of energy used to carry out an activity or physical function.
- Exploration** An undirected search where no prior knowledge of the data is known. The goal is for the user to explore the data to gain insight and understanding.
- G-Force (g)** A measure of acceleration relative to free-fall.
- Gyroscope** A sensor for accurately measuring orientation.
- Heave** In biological science heave is analogous to the y axis in Cartesian coordinates.
- Hertz (Hz)** A unit of frequency. In signal processing this corresponds to the number of samples per second.

Knowledge Discovery (KD) The task of unlocking hidden information and gaining insight into data to realise its value.

Magnetometer Measures the strength and direction of the earth's magnetic field to obtain a relative compass heading in respect to magnetic north.

Multi-Dimensional A data set consisting of more than one data attribute present.

Multi-Variant See Multi-Dimensional.

Overall Dynamic Body Acceleration (ODBA) A method for estimating energy expenditure by summing together dynamic tri-axial acceleration.

Presentation The process of reporting results found during analysis to support the user in communicating evidence of their findings with others.

Static Acceleration A measure of the incline of the accelerometer with respect to the earth's gravitational field which can provide information of an animal's body angle.

Surge In biological science surge is analogous to the z axis in Cartesian coordinates.

Smart Sensor A device which can be used to record or by proxy derive movement and activity. See Daily Diary.

Scalable Vector Graphic (SVG) A two-dimensional image format which produces high quality graphics.

Support Vector Machine (SVM) A classification algorithm, SVM, which finds a hyperplane to separate the data into the classes defined in a training set. Unseen data is assigned to a class based on the hyperplane region it falls under.

Swansea Laboratory for Animal Movement (SLAM) The research group at Swansea University we collaborate with whom study animal movement.

Sway In biological science sway is analogous to the x axis in Cartesian coordinates.

Time-Oriented An umbrella term for any data which has a time component associated with it.

Time-Series An ordered collection of real value data attributes at successive points in time at uniform time intervals.

Tortuosity The property of a curve / line / trajectory having many turns or twists (tortuous).

Tri-Axial A configuration of sensors aligned in orthogonal directions.

Univariate A data set consisting of one data attribute present for each time index.

Vector of the Dynamic Body Acceleration (VeDBA) A method for estimating energy expenditure by computing the vectorial distance of the dynamic tri-axial acceleration.

Visual Analytics (VA) A visualisation technique which combines the processing capabilities of computers with the intellectual strengths of humans through interactive visual interfaces.

Visualisation A human centric approach which leverages human visual perception to create geometric mappings which reveal hidden structure in data and allow us to communicate information through graphical representations.

Part I

Background and Motivation

1. INTRODUCTION AND MOTIVATION

“There is a magic in graphs. The profile of a curve reveals in a flash a whole situation the life history of an epidemic, a panic, or an era of prosperity. The curve informs the mind, awakens the imagination, convinces.”

– Henry D. Hubbard

Contents

1.1	Information Visualisation	6
1.1.1	Time-Oriented Data	7
1.1.2	Brief History of Temporal Visualisation	9
1.2	Smart Sensor Technology	10
1.2.1	Ecology Work-flows	12
1.2.2	Visualisation of Smart Sensor Data	16
1.3	Research Challenges	17
1.4	Contributions	18
1.5	Thesis Structure	20

TECHNOLOGICAL advancements over the past decade have increased our ability to collect data to previously unimaginable volumes. Large corpora of heterogeneous data from diverse sources, such as, remote sensing, advanced simulations, network activity, social media, medical studies, and business transactions, have all accumulated to form the information-age we live in today. Not only is data rapidly growing in size, but the opportunities for sharing it (e.g. cloud storage) allow us to make data accessible to a wider audience faster than ever before. However, our ability to store and transfer data now far exceeds the rate at which we can understand it [KKEM10]. There is now a growing need for advanced knowledge discovery solutions to make sense and use of an ever-increasing data space.

Extracting knowledge from large data plays a crucial role in today’s world for decision making, understanding, and gaining insight. Conventional data analysis has predominantly relied on either manual analysis, and more recently, automated algorithms (i.e. data mining). Manual analysis is a slow, error-prone, and cognitively demanding process when users are faced with large amounts of data. Computers are inherently good at processing large quantities of data and performing repetitive tasks. Data mining exploits the immense processing power of computers through automated algorithms to extract structure from massive data sets [RCM⁺12]. However, computers are limited in their flexibility to apply knowledge to support or refute hypotheses [KKEM10].

The human visual system has an innate capacity for recognising patterns [War04]. Visualisation is a human centric approach which leverage’s human visual perception to create geometric mappings which reveal hidden structure in data and allow us to communicate information through graphical representations. Images are a powerful resource, strongly ingrained into the human perception system [WGK10], which can allow vast amounts of data to be consumed quickly when presented in an intuitive visual format. Domain experts undertake the role of knowledge discovery by studying the resulting visual metaphors for important characteristics and underlying features of the data conveyed in the image, such as, trends, anomalies, clusters, and relationships [WGK10]. Visualisation embraces this to assist an individual in forming a mental model or mental image [Spe07]. The use of visualisation has become ubiquitous across many disciplines, such as, finance, engineering, medicine, and physics.

Effective visualisation assists the user in accomplishing the tasks they wish to undertake with their data. Ward et al. [WGK10] identify *exploration*, *analysis*, and *presentation* as three abstract tasks the user seeks to accomplish with visualisation. *Exploration* is an undirected search where no prior knowledge of the data is known. The goal is for the user to explore the data to gain insight and understanding. From this, the user may find interesting patterns or events which can be used to form a hypothesis. *Analysis* is used to confirm or refute

a hypothesis from the exploration stage by identifying evidence which supports the users proposition. This is a directed search from which facts about the data have already been obtained. Finally, *Presentation* is used to report results found during analysis to support the user in communicating evidence of their findings with others.

Generally, the field of visualisation can be separated into two distinct categories; scientific visualisation and information visualisation. The type of visualisation to be applied depends on the data to be presented. Information visualisation is concerned with abstract data, that is data which contains no inherent mapping to space, such as, database tables and sensed data containing quantitative or qualitative attributes. Shneiderman's task by data type taxonomy [Shn96] for information classifies visualisation by the data type to be presented; 1D linear, 2D map, multi-dimensional, temporal, tree, and network. The goal of information visualisation is to create an effective mapping of data variables to visual dimensions which empowers end-users to gain insight into the underlying data. Conversely, scientific visualisation is concentrated upon making sense of scalar or vector values each of which is associated with a spatial reference point, such as, geographic, medical, or simulation data. Each spatial coordinate is mapped to the screen coordinates via an appropriate projection. The goal of scientific visualisation is to show the relationships and patterns between the data dimensions while maintaining the underlying spatial distribution of the data. The two fields share many similarities, predominantly stemming from the creation of graphical models and visual representations which support the user in knowledge discovery [DOL03]. In this thesis, we are concerned with the field of information visualisation, specifically the application of information visualisation for time-oriented data acquired from biological studies.

The remainder of this chapter is organised as follows. In section 1.1, we introduce the field of information visualisation. In section 1.1.1 we define time-oriented data and characteristics of time. In section 1.1.2 we provide a brief history of temporal information visualisation. Section 1.2 defines the project background and our collaboration with biologists, before discussing the challenges faced during data analysis in section 1.3. Finally, in section 1.4 we identify the solutions we present in this thesis to address the challenges faced and associated contributions these bring to the visualisation research community.

1.1 Information Visualisation

John Stasko said at EuroVis 2014 [Sta14], "Use data mining when you know the question and visualisation when you do not". While this is compelling, when dealing with large data, visual approaches often fall short for exploring and communicating through a large information space (which we further explore in chapter 3). Information visualisation utilises human perceptual capabilities to gain insights into large and abstract data sets to make large data coherent. Similarly, automatic approaches are not tightly integrated with domain experts to give them the flexibility to answer core hypotheses during analysis when in the hands of domain experts. Visual analytics is a recent outgrowth from the information visualisation field for performing analysis given the complexity of an ever-increasing information space [KMS⁺08].

Visual analytics combines the processing capabilities of computers with the intellectual strengths of humans through interactive visual interfaces. Figure 1.1 illustrates the knowledge discovery work-flow realised through visual analytics. The human is engaged with analysis models through interactive visualisation with the underlying data. Users apply domain knowledge through visual interfaces to iteratively refine models. A user-in-the-loop approach allows the user to delve deeper into the data to continuously build and apply knowledge. The integration of the human and machine is often much faster than traditional analytical and data mining techniques [KKEM10].

Although the application of visual analytics had been used for many years prior, Thomas and Cook [TC06] first formally defined the visual analytics process in 2005 as "*the science of analytical reasoning facilitated by visual interactive interfaces*". Keim et al. [KKEM10] later provide a more specific definition, "*Visual analytics combines automated analysis (computational algorithms for discovering patterns) with interactive visualisations (graphical representations of data which the user can interact with) for effective understanding, reasoning and decision making on the basis of a very large and complex dataset*". In essence, users interactively modify parameters of associated models to drive and inform statistical analysis through interactive visualisation facilitated via a feedback loop. Visual analytics is a powerful tool for problems where the size and complexity of the data requires human input

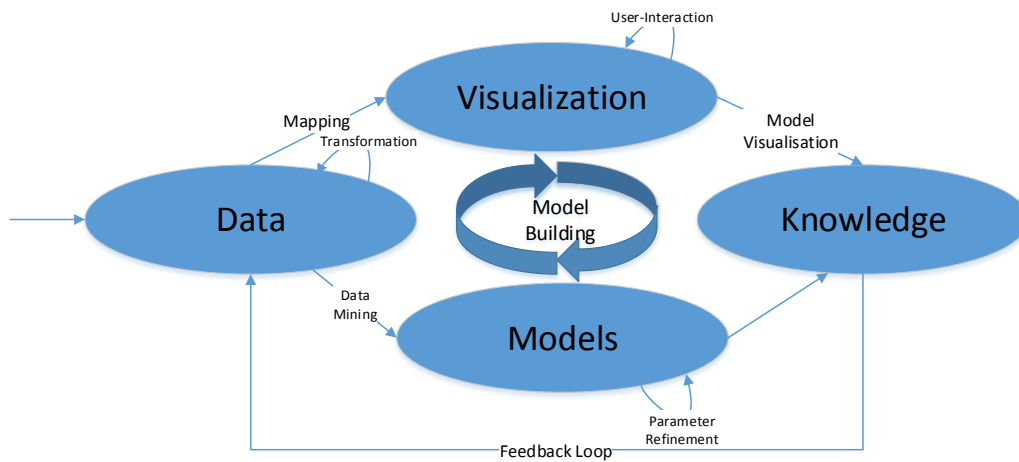


Figure 1.1: The visual analytics process pipeline [KKEM10]. The human engages with analysis models through interaction visualisation. The user refines the parameters of models through combining visualisation and user knowledge in an iterative user-in-the-loop process.

along with machine analysis [KKEM10].

1.1.1 Time-Oriented Data

Time is an inherent dimension which is a crucial aspect of human life, underpinning every action we take. Time is also a fundamental element of information visualisation, allowing us to learn from the past, plan, and predict the future. However, rarely do we need to consider the diverse types of time and the characteristics of associated attributes. Time-oriented data is an umbrella term for any data which has a time component associated with it.

The complicated structure of time consists of many various aspects which require particular tasks and methods to support effective analysis. Scale (ordinal vs. discrete vs. continuous), scope (point-based vs. interval-based), arrangement (linear vs. cyclic) and viewpoint (ordered vs. branching vs. multiple perspective) are all key attributes of the types of temporal data (defined by the temporal classification Aigner et al.[AMST11] introduce).

- *Scale* - The scale at which elements of time are given. Ordinal scales of time refer to only relations between temporal components, using terms, such as, 'before' or 'after'. Temporal sequences are inferred from the relations present, but are not explicit and may be textual summaries. Discrete scales of time map temporal intervals to sequential sets of integers, where each temporal region between two points of time is equidistant throughout. Conversely, continuous time scales map time to a set of real numbers, such that, between any temporal positions, another time point can be referenced. The temporal distance between any two points in time is not uniform.
- *Scope* - The scope of the elements which combine to produce the structure of the time domain. Point-based time is only associated to the time point specified, where there is no notion of what happened between two time-intervals, just the time points defined. For example, the time value 19:15 on the 13/11/2015 relates to one point in time. Interval based time relates to sections or regions of time. For example, the interval 19:15 on the 13/11/2015, to 19:15 on the 15/11/2015 features an interval of two consecutive days in the time domain.
- *Arrangement* - The arrangement of the temporal domain is defined by the temporal structure present. Linear time considers an arrangement ordered from past to future where each time value is enumerated sequentially throughout, such that, the previous item occurred before the current, and the next item occurred afterwards. Cyclic time refers to a temporal domain composed of a set of elements which repeat themselves. For example, seasons, months of the year, or days of the week are all example of cyclic time.
- *Viewpoint* - The viewpoint of time is concerned with time is perceived. We commonly perceive time as ordered, that is, events occurring directly after each other in a linear arrangement. Branching time, diverges into alternative scenarios. This allows the

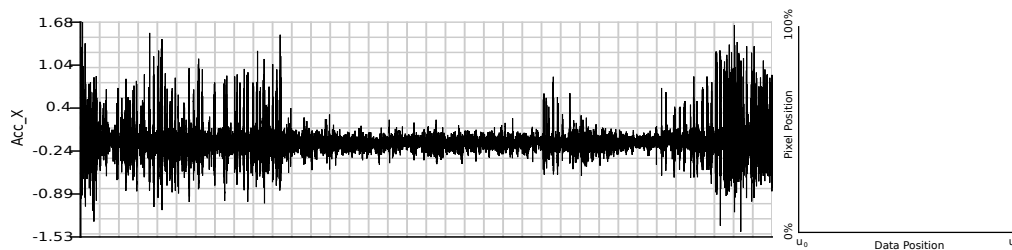


Figure 1.2: This visualisation illustrates the traditional time-series graph (left) as conceived by Playfair in 1786 [Pla86]. A linear mapping from the temporal domain to the time (x) axis is applied (right). This image utilises the mapping notation presented by Isenberg et al. [IBDF11].

comparison of different scenarios, which is useful for planning and risk management. Perspective time models present simultaneous views of the same phenomena from different viewpoints. For example, eye witness accounts are from different individuals giving their viewpoints on what they perceived occurred at a specific point in time.

In addition to the types of time, there are a multitude of attributes which can be associated. Aigner et al. [AMST11] specify that time dependent data channels can be characterised by *scale* (quantitative vs. qualitative), *frame of reference* (abstract vs. spatial), *kind of data* (events vs. states), and *number of variables* (univariate vs. multivariate).

- *Scale* - The domain type of each data attribute may be a qualitative or quantitative measure. Qualitative data attributes consist of categorical data. That is, non-numerical data which has a natural language description, containing either a natural ordering (ordinal) or unordered (nominal). Quantitative attributes contain numerical measurements. That is, they are based on a discrete or continuous range of numerical values.
- *Frame of reference* - The spatial domain the data is bound to. Abstract data does not exist in a spatial domain, or have any inherent spatial location associated to it. The role of the visualisation is to map data to a suitable spatial domain and layout, from which insight can be gained into the data. Conversely, spatial data has a spatial location associated to it (typically in two or three-dimensional space). That is, the data is already bound to space through the data acquisition process (e.g medical data).
- *Kind of data* - The way in which the temporal data represents associated values may be through events or states. Events refer to changes in state. States are phases between events. States may encode regularly sampled data, while events encode changes between states.
- *Number of variables* - The number of time-dependent variables in the data set, that is, the number of variables recorded over time. Univariate data refers to one single data value associated to each point in time. Multi-variate data consists of multiple data values (more than one) associated to each temporal component.

Time-series data is the most common form of time-oriented data, defined as an ordered collection of real value data attributes at successive points in time at uniform time intervals. Equation 1.1 represents a univariate time-series (Ta) of length (n). Multi-dimensional time-series data consists of multiple variables bound to each successive point in time. Equation 1.2 represents a multi-variate time-series (Tb) of length (n) with (k) variables.

$$Ta = \{t_0, t_1, \dots, t_n\} \quad (1.1)$$

$$Tb = \{\{t_{0,1}, t_{0,2}, \dots, t_{0,k}\}, \{t_{1,1}, t_{1,2}, \dots, t_{1,k}\}, \dots, \{t_{n,1}, t_{n,2}, \dots, t_{n,k}\}\} \quad (1.2)$$

Virtually every field of science is linked to some form of measurements over time, which all have strong connections to temporal phenomena. The values of each stock in the London Stock Exchange at the close of the day for a month, ECG measurements of brain activity at high frequency, accelerometer signals of animal activity, and website visitor data for each day of the previous year, are all examples of time-series data.

A time-series is commonly represented as a time-series graph (Figure 1.2). Each data item is represented by a point on a two-dimensional viewing plane, with the horizontal (x) position associated with the temporal location, and corresponding vertical (y) position defined

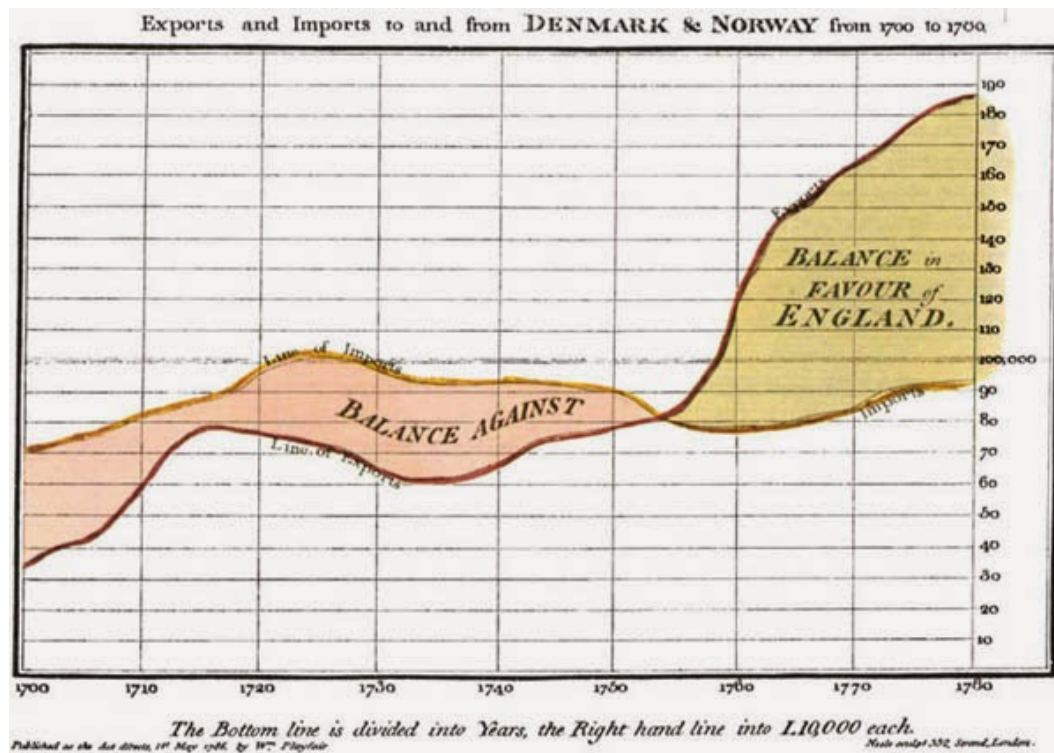


Figure 1.3: This visualisation illustrates Playfair’s chart of exports and imports to and from Denmark and Norway, overlaid with that of England. The result shows a clear correlation between them. Image Credit: [Pla86]

by the associated data value. A singular line joins together all of the points in chronological order to allow interpolation of values across time. A linear mapping is utilised from the display space to the time-axis, such that a uniform sampling is applied to each point in time.

1.1.2 Brief History of Temporal Visualisation

Data visualisation has been applied to understand temporal phenomena long before the invention of computers, and even before it was first formally defined. The earliest ancient visualisations were found in Dordogne, France dating backward to approximately 25,000–32,000 BCE, depicting phases of the moon throughout its cycle etched into the bones of animals [Mar72]. Over the centuries there have been many historical visualisations discovered, mainly for visually recording information for future reference, and story-telling purposes (e.g. the success of a hunt painted onto the stones in caves). However, it wasn’t until the 17th century that the benefits of representing statistics using graphical representations was first reported. In this section, we provide a brief introduction to some of the earliest notable historical visualisations for statistical analysis of temporal data which lay the foundations for the information visualisation field we report and build on in this document.

William Playfair (1759 - 1823) was one of the earliest visualisation pioneers, and is often referenced as the father of graphical methods for statistics. His publication *The Commercial and Political Atlas and Statistical Breviary* [Pla86] published in 1786 contained one of the first recorded occurrences of a time-series, depicting the national debt of England against a horizontal time-line with key historical events annotated. Playfair used this graph to analyse the impact of historical events on the levels of national debt. A total of 43 time-series were presented in this publication, including overlaying two time-series to identify correlations between the imports and exports from Denmark and Norway to England (Figure 1.3). His graphs formulated the basis of the line graph which is ubiquitous today. Several further representations in his publication outlined the pie chart, silhouette, bar graph, and line plot [AMST11].

Charles Joseph Minard’s (1781 - 1870) representation of Napoleon’s French Invasion of Russia, has been widely quoted as the best infographic ever created [Tuf01]. His visualisation (Figure 1.4) represents six data channels across two dimensions, including, geographic position, time, temperature, direction and distance of movement, and the number of surviving

forward a concept for a particular sensor configuration within a tag that they called the Daily Diary (DD), where analysis yielded value well beyond the simple mathematical sum of its individual parts [WSL08]. Specifically, the suggestion advocated the combination of tri-axial accelerometers, tri-axial magnetometers, and pressure and speed transducers together with environmental sensors such as temperature, light, and humidity.

The theory was that this constellation of sensors, sampled at infra-second rates (upward of 32 times a second), would allow tag users to be able to derive four key elements of animal ecology seamlessly. These are: (1) animal trajectory, and therefore position [WLD⁺07], (2) animal behaviour [SWQ⁺08], (3) energy expenditure [WWQ⁺06], and (4) the environmental conditions to which the tag carrier is exposed [WGS⁺02]. The development of animal-attached 'smart' tags have revolutionised biologists understanding of the ecology of wild animals [PTWS14]. This approach has allowed researchers to look at everything from the behaviour of whales chasing prey at depths of over 1 km underwater [ASJM⁺08] to the physiology of geese migrating over the Himalayas [HBB⁺11].

The tri-axial accelerometer data forms the main basis for the biologists in providing quantitative data on body posture, motion, and energetics. The three accelerometer axes are aligned in orthogonal directions. In biological science these are termed heave, surge, and sway, analogous to the y , z and x axes in Cartesian coordinates [GJL⁺09b]. Acceleration is the change in velocity over time. Newtons second law specifies that the acceleration of an object is directly proportional to the net force acting upon. An accelerometer utilises this law to measure acceleration due to gravity (g -force) in respect to the earth's gravitational field. Conceptually, an accelerometer can be considered as a weight suspended in-between two sides adjoined via springs (Figure 1.5). When subject to an acceleration the mass will experience a gravitational force applied, which will result in a shift of the mass to one side, compressing one of the springs while stretching the other. The distortion of the spring is proportional to the force applied on the mass by the acceleration, this records a linear acceleration in the given axis, equivalent to the g -force applied. It is important to consider that an accelerometer pointing directly through to the centre of the earth will measure an acceleration upwards of $9.8m/s^2$ or 1 g -force (g). This is due to the constant force of the earth's gravitational field. A triad of accelerometer sensors mounted orthogonally (otherwise known as a tri-axial accelerometer) allows the measurement of three-dimensional acceleration around the mounted sensor and therefore the reconstruction of the actual forces applied in all dimensions.

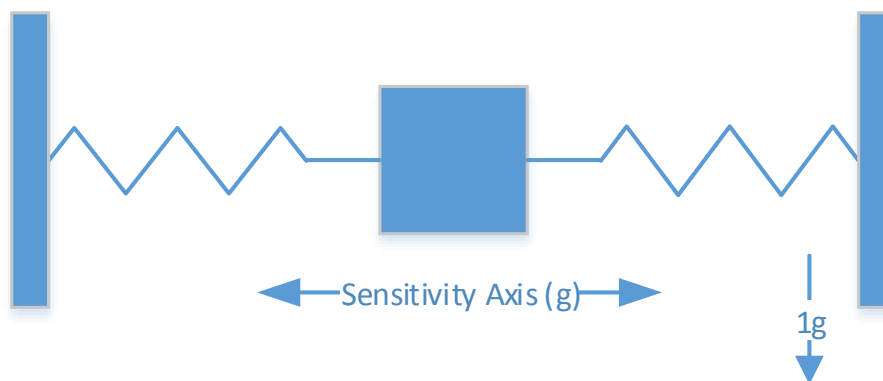


Figure 1.5: An accelerometer can be conceptually considered as an object suspended in-between two springs. When the object is subject to acceleration it is shifted. The displacement measured is proportional to the acceleration due to gravity (g -force).

The acceleration channel can be decomposed into static and dynamic components. Static acceleration is a measure of the incline of the accelerometer with respect to the earth's gravitational field and can provide information of an animal's body angle. Static acceleration measured in three axes enables the calculation of body pitch and roll, corresponding to rotations in the sway and surge axes respectively. The number of possible activities can be reduced by utilising body posture. For example, consider a Penguin upright for minutes or

hours must be on land (standing). Alternatively, if the Penguin is horizontal it can either be on land (resting) or in water (swimming).

The dynamic component of acceleration represents the change in velocity as a result of body motion providing information on animal movement. It can be derived by subtracting the attained static acceleration from the raw acceleration signal [WWQ⁺06]. The dominant signal on an axis can describe the type of motion undertaken. For example, a swimming fish will produce an oscillating signal predominately in the sway axis resulting from flapping in the tail. The frequency and amplitude of the signal are dependent on the speed and strength of the movement. However, some behaviours are the product of movements in multiple axes, for example figure 1.6 shows a barrel roll on a device deployed on a leatherback turtle where we see rotational patterns recorded across all three axes. The signal goes upwards and downwards in all axes representing flapping behaviour, as the turtle rolls the movement is identifiable across the remaining axes. Behaviour spanning across multiple dimensions are difficult to derive and interpret since it requires the signal to be reconstructed in three-dimensional space.

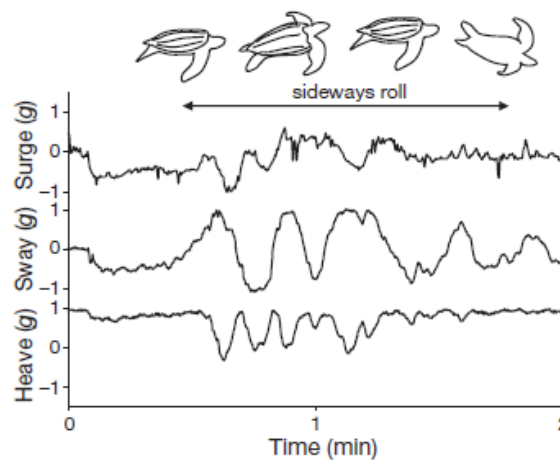


Figure 1.6: A leatherback turtle performing a rolling manoeuvre through 90 degree rotations in the heave axis, and 180 degree rotations in the sway axis. Image Credit: Shepard et al. [SH08]

Additional environmental attributes, such as pressure, temperature and light level support the identification of animal activity. For example, a change from high to low pressure generally indicates ascending behaviour in water in aquatic species. Various authors have presented the use of derived channels to highlight specific features of sensor data which indicate behaviour including posture [NSFR⁺12], signal dynamism [WTD⁺05], repetition in patterns [SSI⁺09a] and specificity in rates of change of particular signals [SAS⁺11].

Complementing the accelerometer and environmental sensors is the tri-axial magnetometer which measures the strength and direction of the earth's magnetic field to obtain a relative compass heading in respect to magnetic north. The earth's magnetic field is constructed of field lines approximating a magnetic dipole. Each field line originates at a point near the magnetic South Pole and terminates at a point near the magnetic North Pole. Specifically, this has uses for deriving geographical position, and behaviour through the visual inspection of animal orientation. We further discuss the tri-axial magnetometer and the associated methods for deriving geographical position in chapter 5.

1.2.1 Ecology Work-flows

A vital part of our collaboration with ecologists has been studying their processes and operations which they undertake to gain insight and understanding into animal behaviour. On a bi-weekly basis we assembled to observe the ecologists processes. Often this involved discussing associated problems and challenges faced in their domain area. Over the past decade there has been a primary focus on the advancement of sensor technology to record animal behavioural states. Predominantly, our conversations would originate from the fact the technology is now so far ahead of the available methodologies there is no sufficient way to unlock value from the acquired data to drive research advancement. In this section, we

report the observations we have made and highlight the challenges faced which we have perceived.

From our observations and understanding we report a simplified work-flow of the biologists processes (Figure 1.7). This generalisation consists of five steps: (i) deployment and retrieval of the daily diary device, (ii) exploration to identify a mapping of the recorded signal to animal behaviour, (iii) analysis to annotate the data with labels of animal behaviour, (iv) higher order exploration and analysis on the animal behavioural data, before finally (v) results of the biological research study are shared with the wider research community. In reality, this process may consist of more phases with transitions back to earlier tasks as the biologist gains knowledge and understanding in addition to looping back to further rounds of exploration and analysis applied to the animal behavioural states before being presenting to the wider community. To keep this section simple, we discuss the base processes and conceptualise the higher order exploration and analysis as a separate task. We now detail each of these components in more detail.

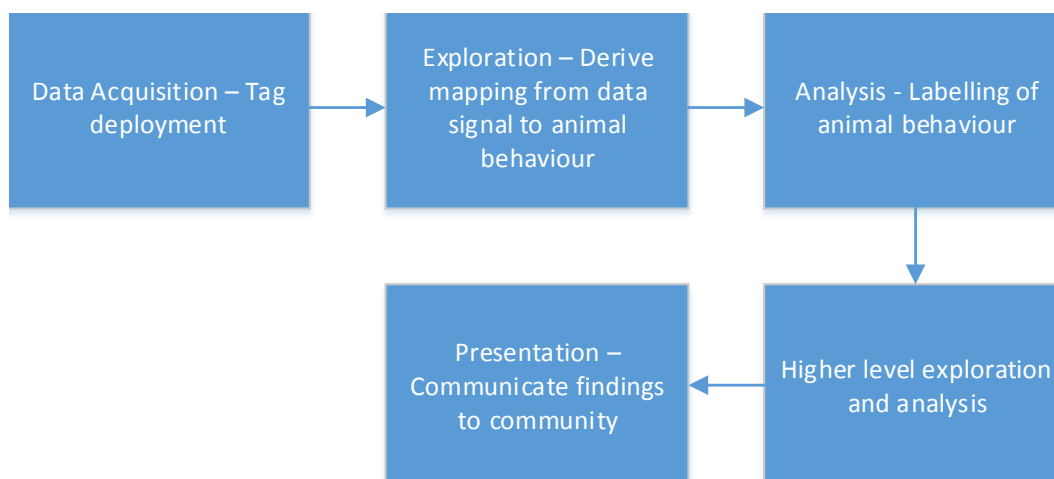


Figure 1.7: A conceptualisation of the biologists work-flows which we have observed over the duration of our collaboration.

Data Acquisition - The project initiates with the biologists formulating a research question they seek to investigate, for example, Thiem et al. [TDH⁺16] sought to investigate the activity and energy costs of an adult lake sturgeon during a fishway passage (a man-made structure around barriers otherwise blocking the fish from migrating, such as, a dam) to assess the conservational effects. Dependant on the research question to be studied the constellation of sensors is chosen. The choice of sensors is governed by the data required to support the investigation. In the previous study a minimum of magnetometer and accelerometer data channels are required to derive the associated animal heading, and behaviour coupled with energy expenditure. Each sensor bound to the device will result in a higher throughput voltage, and therefore will utilise more power which can affect the maximum duration of the deployment. In addition to determining which sensors to bind, a suitable sampling rate needs to be selected. Typically, a sampling rate of twice the minimum behaviour frequency is chosen, for example, a Cormorant wing beat occurs at a frequency of approximately 5Hz [Pen01], therefore a minimum recording frequency of 10Hz is desired to be able to extract this behaviour in the resulting data. The sampling rate, and constellation of sensors both impact the deployment duration. Often a trade-off is made between the two. As battery technology strengthens we will see more advanced sensors being utilised (e.g gyroscopes) at a higher recording frequency.

Often devices may malfunction, fall off, may not be retrievable, or may not contain data useful to support the study. To overcome these uncertainties, it is typical for biologists to undertake many simultaneous deployments to collect as much data as possible in the hope they return with the data they desire. Currently, the biologists tag as many animals as possible, although recent software has been introduced to assist in selecting a suitable number of deployments using fuzzy logic based on previous deployment success [BAA⁺14].

To collect the data the biologist must first deploy the tag on the animal. This process is largely reliant on the species and surrounding environmental constraints. If the animal is on land they are typically captured, a sedative may be used if required. For aquatic species where it is not possible to capture the animal (e.g. a whale), the device is placed in position

using a pole. As smart-sensor devices are still an emerging technology, there is currently no precise protocols for deployment, which is an active area of research in the biology domain. The result of this is the positioning of sensors differs between deployments as it is challenging to locate precisely the same location. Additionally, the variability in animal behaviour can lead to sensors moving (i.e. scratching).

Once the device is mounted, the daily diary device starts recording and the animal is released back into the wild. During the deployment, the biologist may record video data or keep observational logs if feasible to later help with the exploration and analysis tasks. However, often this is not possible due to environmental constraints. Retrieving the device is also a challenging aspect. For aquatic species, the mount used is time based, such that, after a specified period the device is released from the animal. It then floats to the surface which the biologists will then retrieve afterwards. Otherwise the animal will be tracked often using a complementary device (e.g. radio tracking) and captured.

Once the device is recovered the data is downloaded onto a computer. The daily diary records to a binary format (for data storage purposes) which is then loaded into a separate application which parses and serialises the binary data into a ASCII (American Standard Code for Information Interchange) formatted CSV (Comma Separated Value) file. We operate on the ASCII formatted data throughout this project since it is of the interest of the wider community to maintain an open format which can be easily converted between.

Exploration - The objective of data exploration is for the biologists to understand the mapping of the data signals obtained to animal behaviour. As previously mentioned this task may be assisted with the use of synchronised video or observational logs. However, this complementary data is rarely available due to environmental constraints. In which case, the mapping of signal to behaviour must be derived using just the recorded data coupled with the domain expertise of the biologist.

Currently, data exploration is performed using a off-the-shelf commercial software package [Sys]. Initially, the data series is plotted in full (Figure 1.8), where the user can select a region of data to zoom on and make the focus of the display using short-cut keys. The user zooms in on a data region to the required data resolution to identify behaviour and then begins panning through the data linearly across the temporal duration of the series. The software allows the user to add and remove plots of different data attributes throughout to assist in determining the associated behaviour. The biologist determines what signals and features of the data constitutes animal behaviours of interest to the research study.

The ecologists operate on the data by inspecting stacked time-series graphs (Figure 1.6), often with one for each component of the static and dynamic signal, in addition to supporting environmental attributes present (e.g. pressure) to assist in the identification of behaviour. Shepard et al. [SH08] discuss the process of deriving animal behaviour through a systematic approach. First using the static component to identify animal posture and reduce the number of possible behaviours. Behaviour is then further refined through the visual inspection of patterns in the dynamic acceleration components. Identifying the actual behaviour requires significant domain knowledge in this way. Such a systematic method is rarely applicable to the data but demonstrates how behaviour can be interpreted from the accelerometer signal in the hands of experts. Often it is a case of trial and error applying the biologists' knowledge of the mounting of the device, the technology (sensors) coupled with knowledge gathered from prior research on the species being studied.

Analysis - During analysis the biologists utilises the earlier found mapping of the data signals to behaviour. The goal is to label the entire data series with the animal behaviours of interest for the study. These are the behaviours to be utilised to answer the core research questions initially posed.

The biologist starts at the beginning of the data set and maintains a level of zoom required to extract the behaviours present. A corresponding spreadsheet is created which is used to store the time-stamped locations of each instance of the behaviours identified. For large data, this is an inefficient process, since often a percentage of the signal can be considered as non-interesting (predominantly through inactivity). Also, the biologist may further refine their understanding of the behaviour as they see more instances which either expands or reduces the search space. Behaviours may exhibit minute differences in the signal, in which case it is useful to compare existing behaviours. This is a cognitively demanding and time-consuming process when considering data recorded at a high-frequency over prolonged periods of time. Labelling the behavioural data can take up to approximately several hours for a data deployment lasting five days.

Higher-level exploration and analysis - Once the labelled data sequences have been

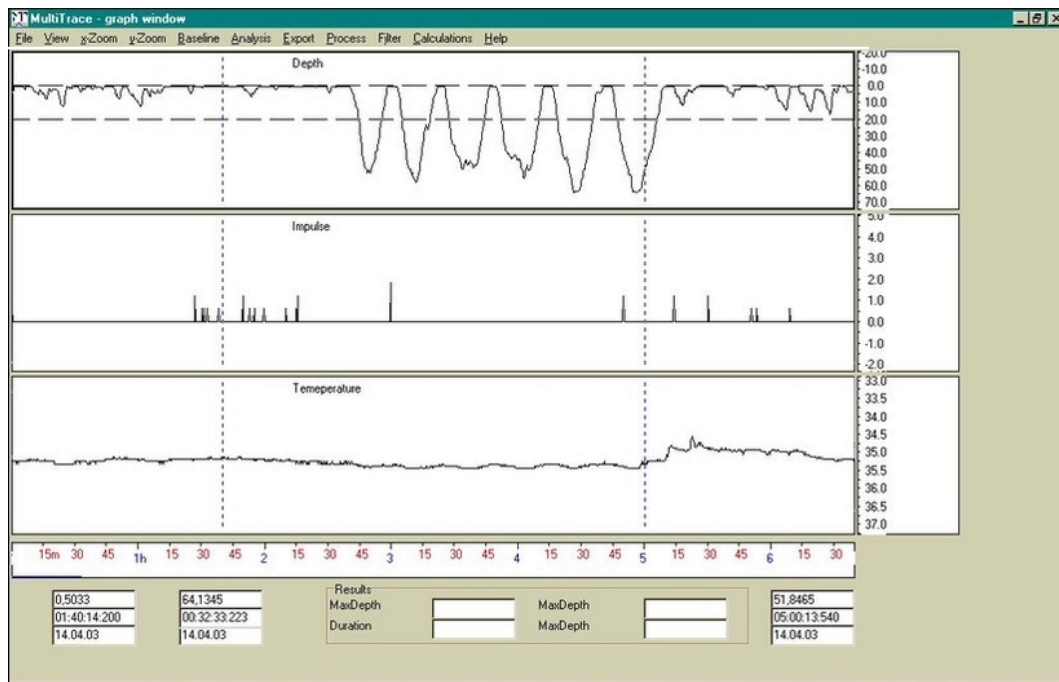


Figure 1.8: Commercial graphing software utilised by the movement ecologists at the SLAM research centre to explore, analyse, and present their smart sensor data. Image Credit: Jensen [Sys].

acquired. The biologists will seek to perform further iterations of exploration and analysis to derive complementary data, build hypotheses, and gain a further understanding of the data for each behaviour instance. This is often undertaken at a higher-order, that is, on the newly acquired labelled data sequences coupled with the underlying data. This may involve manually segmenting the original data series into the labelled data regions, or other transformations to obtain the data into a suitable format. We list some of the higher-order exploration and analysis tasks we have observed in the field. This list is by no means exhaustive but gives a general overview of some of the supporting evidence required for their studies.

- The occurrences and order of animal behaviour can help identify behavioural patterns. The sequences of behavioural state and transitions are explored and analysed to help understand what and why animals perform specific tasks.
- Linking locational or environmental conditions to behaviour can help understand the way an animal uses a specific location under certain conditions. The ecology of an animal is important for understanding conservational effects.
- Derivation and analysis of the amount of energy an animal uses can help indicate how animals survive in the wild. This may be on a time-scale basis (e.g. per day), across specified behaviours (e.g. flying), or per location (e.g. in water).
- Combining behaviour with movement trajectories assists in understanding what an animal is doing, where it was doing it, and how it was moving. This is particularly important in resolving animal activity (e.g. thermalling behaviour identified by gliding while navigating in a circular path to gain altitude).
- Comparing behaviour stylisation across species to understand how specific behaviours or activity are performed. In the sports science domain this has particular uses where effective movement must be stylised for maximum performance. The goal is to compare between different species performing the same behaviour to optimise or understand the way movement occurs.

It is worth noting that the exploration and analysis tasks are constrained by the software and methods available. Typically, these are undertaken using spreadsheet software as the research group lacks expertise in software development and statistical methods. As such,

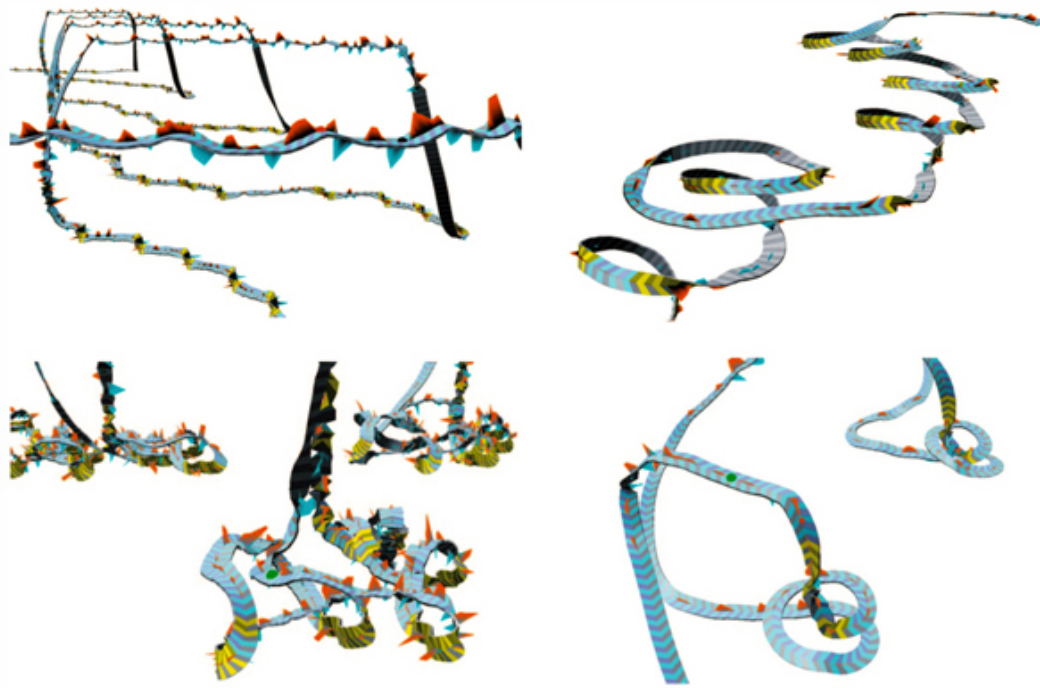


Figure 1.9: A visualisation of dead reckoned animal movement trajectories representing the kinematic patterns from Whales in the wild. Image credit: Ware et al. [WAPW06]

the lack of available methodologies and tools has limited the biologists ability to realise the value of their data.

Presentation - After the user has drawn their conclusions from the data and proved / disproved their hypotheses, they will seek to present their findings to their colleagues in their research group or to the larger research community through academic journals and conferences. To facilitate this, the user creates a corresponding document with screen captures and associated time-stamps for later reference. Visual indications of the relevant underlying data, such as, that shown in Figure 1.6 may be used to complement the findings as visual summaries of the data. It is typical that co-authors or co-investigators will seek to validate the results and pan through the series to identify these in the data. At present, there is no way to directly access the data regions marked, as such, this is often an overlooked task.

1.2.2 Visualisation of Smart Sensor Data

The current state-of-the-art for visualising smart sensor data focuses on enhancing the exploration of tri-axial accelerometer data to assist in finding interesting features and patterns.

Spherical coordinates are applied to show common behavioural cycles [GJL⁺09b]. Spherical scatter plots reveal the geometric distribution of data, along with spherical histograms which unveil common animal movements. This method is effective at enabling the exploration of tri-axial accelerometer data for the identification of behaviours. However, the temporal aspect of the data is removed which makes it infeasible to extract temporal patterns for larger data sequences. Overlaid plots of acceleration within the spherical scatter plot show common behavioural cycles within the data. We further detail this method in chapter 6, where we inherit and build on this approach to link behaviour and energetics.

The spherical plots are extended to show them in relation to other sensor channels by combining with parallel coordinates [WGJL12]. The advantages of spherical coordinates for the visualisation of tri-axial accelerometer data with those of parallel coordinates for interrogating large time series multi-variate sensor data are joined together. A link is generated between the two coordinate systems to enable the tri-axial data channels to be perceived in relation to the other data attributes present.

Further attributes have been derived to abstract from the raw sensor values. Animal movement has been reconstructed using a dead reckoning approach [WAPW06] whereby the initial deployment position is used to determine future positions by integrating heading

and speed across a known duration. The resulting movement trajectories are represented as ribbons, enabling orientation and roll information to be encoded (Figure 1.9). Arrows are textured onto each path to encode directional information to enable biologists to derive behavioural information from the visual inspection of the animal movement paths.

Labelled behaviour is used to show higher-order state transitions between behaviours [BBG⁺09]. Summaries of the behaviour transitions visualised within the context of the state they occur in provide easy-to-follow transition paths to expose common behaviour transitions.

Unlike previous work which assist in understanding and exploring sensor data, we introduce methodologies which span across the biologists pipeline, for exploration, analysis, and presentation.

1.3 Research Challenges

The excitement at the potential inherent in sophisticated animal tags has, however, been tempered by a lack of methodologies to overcome the diverse and challenging nature of the underlying data. The impact of this is currently no process exists to unlock value and advance research in the field.

Generally, information visualisation approaches suffer from specific challenges when operating on certain types of data. Our data is time dependent, with sensor measurements taken at regular intervals via a sampling rate between 32 - 128hz during the deployment. Each sensor value is strongly associated to the time stamp recorded when each sensor value is logged. Our data is large, typically a deployment lasts up to a week when sampled at a minimum of 32hz accounts for 19,353,600 measurements. The number of data items increases when sampling rates are higher, or deployment length is extended. Finally, our data is multi-dimensional, typically during deployment up to 11 data channels are recorded, including, tri-axial acceleration, tri-axial magnetometer data, pressure, light intensity, latitude, and longitude to name a few. We outline the challenges faced when attempting to visualise this data, categorised by data-type. These challenges are important to consider to understand the challenges faced by our collaborators operating on their acquired smart sensor data.

1. The abundance of large data commonplace today has coined the term 'big data'. Such data, is primarily concerned with data collections which require analytical methods to extract value. That is, where traditional analysis techniques are no longer adequate. One challenge is that screen resolution is small in comparison to data storage capacity. When more data items are rendered than the available pixels an over-plotting and occlusion problem occurs where more than one data item is assigned to each pixel, which leads to a loss of information [GPL⁺11]. Furthermore, when analysing and rendering a large number of data items, slow processing speeds can occur which can obstruct user-interaction. Communicating and navigating through a large information space is essential for effective information visualisation.
2. Multiple dimensional data is concerned with data that contains two or more data dimensions. Users typically wish to be able to perceive and identify correlations between dimensions. Effectively mapping data containing a high number of dimensions in a spatially low dimension environment (i.e. the two-dimensions of a computer monitor) to a visual metaphor from which correlations can be identified is challenging. Various visualisation techniques, interaction methods and visual layouts have all been proposed in the literature.
3. Time-dependent data is concerned with any data that contains variables strongly bound to a time component. The user often wishes to maintain the chronological order of events while identifying interesting patterns across the series, perceiving the distribution and correlations between different points in time. The different characteristics of time and diverse array of associated data attributes (Section 1.1.1) highlight the challenges of analysing time-oriented data and the importance of an effective visual representation. Appropriate interaction is required to navigate through time, and undertake the task-at-hand e.g. searching.

This thesis specifically presents methodologies to deal with the problem of the analysis of the high resolution, multiple channel (and therefore multi-dimensional) data acquired by smart sensor tags [JMF03]; by studying the biologists work-flows we can decompose

the research challenges they face into specific domain level problem statements which we investigate in this thesis. It is the hypothesis of this work that visualisation coupled with machine learning (where necessary) is able to effectively and efficiently enable biologists to undertake these tasks.

1. There is no protocol used for mounting devices during deployment. As such, the mounting location and fixing of devices to each animal differs between each study. The result of this is that the behaviour signals acquired differ between deployment for the same species of the same behaviour. Additionally, the device may move position during the deployment from animal behaviour (e.g. scratching).
2. Ecologists manually explore time-series graphs of several attributes to gain an understanding of the mapping from signal to behaviour. This is a cognitively demanding process requiring significant domain knowledge. The biologist will build knowledge as they go through the data series, tying together event sequences, along with inspecting associated environmental conditions and deriving additional attributes (e.g. posture and energy use). Often signals are compared to explore the underlying motion. There is no obvious process to facilitate exploring the data to assist the biologists in accelerating the understanding of the mapping between the data space and animal behaviour.
3. Once the biologist has determined the signal to behaviour mapping, the biologist will manually decode the data set and label the entire data series with the behaviours of interest. This is a time consuming task and entirely manual process currently supported by writing the time-stamps of each instance into an associated spreadsheet. Often signals need to be compared with the rest of the classified instances to understand how likely it is to be an instance of that behaviour. There is no support for these user intents through the existing software and methods available.
4. Once the biologist has explored and analysed the data it is typical they will seek to present their findings to their research group or to the larger research community through academic journals and conferences. There is currently no effective protocol for allowing direct access to findings which co-authors or co-investigators can use to validate or draw their own conclusions. Because of this it is often an overlooked task.
5. In 2008 Wilson et al. [WLD⁺07] proposed four key elements of animal ecology realised through the daily diary device. These are: (1) animal trajectory, and therefore position [WLD⁺07] (2) animal behaviour [SWQ⁺08] (3) energy expenditure [WWQ⁺06] and (4) the environmental conditions to which the tag carrier is exposed [WGS⁺02]. While some software is available to help determine some aspects of that advocated by the daily diary concept [BCGL⁺14], there is nothing that binds the concepts and data attributes together. Therefore, the four elements are usually derived separately through their own distinct tools. As a result of this the biologists typically use purpose-built spreadsheets and manually segment or transpose their data into the formats they require for analysis or as input to the off-the-shelf tools. A lack of expertise in software development and statistical methods currently limits the advancement of research in the field.
6. Since the data attributes required for analysis are typically computed as separate entities, there is poor support for higher-level exploration and analysis tasks (previously identified in section 1.2.1) which require data attributes to be bound together. Purpose built methods and visualisation techniques are required to effectively support these tasks.

In this thesis, we target methods for these specific research challenges. We realise this through the application of visualisation to enable movement ecologists to improve their understanding of smart sensor data and allow them to undertake their work flows.

1.4 Contributions

The main contributions of this thesis are as follows:

- An up to date state of the art review of temporal visual analytic techniques applied to typical data mining tasks. We introduce terminology for data mining, and time-oriented data visualisation. This review provides a useful introduction to the field of time based visual analytics for abstract temporal data and demonstrates its application for solving difficult problems arising from time-oriented data.
- We evaluate state of the art visualisation techniques for time-series and extensions to these existing approaches through a graphical survey, task based evaluation, and classification of the related approaches. Based on our evaluation we propose TimeNotes [WBJ16], a visualisation technique utilising built-for-purpose interaction techniques that supports analysis, interaction, and presentation tasks. TimeNotes is a more effective approach for chart visualisation and interaction with time-series data. We evaluate the effectiveness of TimeNotes through an empirical study and a field study which highlights the application of our approach applied to time-series data in comparison to the state-of-the-art methods. We further report feedback from a deployment of the software with biologists.
- The reality that researchers face with perhaps 10 channels of data recorded at sub-second rates is the subsequent search and classification of specific phenomena throughout the data, it is a bottleneck in the analysis pipeline. We introduce a generalisation of the tasks associated with the classification of time-series data. Using this generalisation as design guidance, we realise, TimeClassifier [WJL⁺15a], a visual analytics system for the semi-automatic classification of animal behaviour, combining the search of large time-series data with the semi-automatic classification of events using user-defined templates. Our system requires one instance of behaviour for the matching process to take place. We utilise visualisation to search for matching behaviour classifications and interaction to facilitate a user in the loop approach for the checking, rejecting and accepting of results. Our system provides biologists with a working solution which they have confidence in, and can analyse large complex data sets in a shorter time-frame with good precision and recall.
- We encapsulate TimeClassifier into an all-encompassing software suite, Framework4 [WJL⁺15b], which operates on smart sensor data to determine the four key elements considered pivotal for movement analysis from such tags. These are; animal trajectory, behaviour, energy expenditure and quantification of the environment in which the animal moves. The program transforms smart sensor data into movement trajectories, template-matched behaviours, dynamic body acceleration-derived energetics and position-linked environmental data before exporting it all into a single file. Biologists are thus left with a single data set where animal actions and environmental conditions can be linked across time and space. Framework4 is a user-friendly software that assists biologists in elucidating 4 key aspects of wild animal ecology using data derived from tags with multiple sensors recording at high rates. To achieve this, we introduce a new way of combining high resolution motion sensor data with GPS data to create a view of the animal at both the fine scale motor movement and large scale movement. We demonstrate the application of Framework4 through two real-world case studies; firstly, the introduction of a new head mounted logging device for the derivation of head orientation and its role in assessing the environment [WHW⁺15] supported by Framework4. We then further illustrate the insights Framework4 have made possible into the data obtained to determine a series of rules behind how humans assess environments [WNW⁺15].
- We present a new work-in-progress visualisation technique named the g-sphere for animal/human-attached acceleration data that allows power use for movement to be married with behavioural data derived in Framework4. G-sphere visualisations have the potential to highlight changes in movement patterns associated with, for example, human emotional state, injury and best practice in single sports manoeuvres but extend through to highlighting energy-based behavioural ecology in wild animals over time periods ranging from seconds to years.

1.5 Thesis Structure

The remainder of the thesis is structured as follows. In Chapter 2 we discuss state-of-the-art techniques in the visualisation and visual analytic fields for the analysis of time-oriented data. In Chapter 3, we comparatively evaluate existing methods for the visual inspection of time-series and introduce new visualisations for navigating and communication through large data space. We undertake and report an empirical and field study of our approach on smart sensor data. In Chapter 4, we identify that researchers faced with multiple data channels find it particularly challenging to manually decode behaviour. We introduce TimeClassifier a visual analytic system for the classification of time-series to assist in labelling smart sensor data. Chapter 5, extends this work to determine dead-reckoning movements, template-matched behaviours, energetics and position-linked environmental data from smart sensor data into one unified file format. In Chapter 6, we introduce a visualisation framework, g-sphere to link together the behaviour and energetics we derive in Chapter 5, and allow multiple dimensions to be visualised simultaneously. Finally, in Chapter 7 we conclude our findings and outline fruitful areas for future work in this exciting research area.

2. A SURVEY OF VISUALISATION TECHNIQUES FOR TEMPORAL KNOWLEDGE DISCOVERY

“Information is the oil of the 21st century, and analytics is the combustion engine.”
– Peter Sondergaard, senior vice president at Gartner

Contents

2.1 Introduction and Motivation	21
2.1.1 Contributions	22
2.2 Knowledge Discovery	22
2.3 Data Mining	24
2.4 Information Visualisation	27
2.4.1 Layout Based	28
2.4.2 Histogram Based	29
2.4.3 Lens Based	30
2.5 Visual Analytics	32
2.6 Literature Classification	33
2.6.1 Classification	33
2.6.2 Clustering	35
2.6.3 Indexing and Retrieval	36
2.6.4 Pattern Discovery	39
2.6.5 Prediction	44
2.6.6 Segmentation	45
2.7 Discussion and Summary	47

THE field of information visualisation is a relatively young discipline which is acquiring more traction across academia and industry as data becomes more complex. With this increase, there has been a strong focus on a recent outgrowth from the field, visual analytics, concerned with making sense of an increasing information space. The study of time-dependent data is a classic topic within the visualisation field, as temporal data provides many challenges originating from its diverse structure and types of associated attributes (section 1.1.1 provides an overview). The resounding complexity has resulted in a great amount of work to create methodologies which assist users in discovering knowledge from large, temporal data.

In this chapter, we introduce the knowledge discovery process and typical tasks users wish to undertake with temporal data. We then introduce the two traditional methods of realising knowledge discovery, through automated algorithms (data mining) and visualisation, specifically, information visualisation, associated with abstract temporal data (data which has no reference to a spatial domain). Finally, we review state of the art visual analytic approaches for abstract temporal data from the past decade, this gives insight into how the community are binding together information visualisation with automated algorithms through a user-in-the-loop approach.

The aim of this survey is to identify visual analytic methods for undertaking temporal mining tasks, concerned with abstract temporal data. We do not survey any literature for data with spatial attributes, which is beyond the scope of this work where we are interested in mapping data to a visual metaphor which has no previous spatial information associated. This survey serves as a useful introduction to the field of temporal visual analytics and demonstrates its application for solving difficult problems arising from time-oriented data. While, the aim of this chapter is to provide a broad overview we also scrutinise whether these methods could aide biologists in the field for analysing smart sensor data.

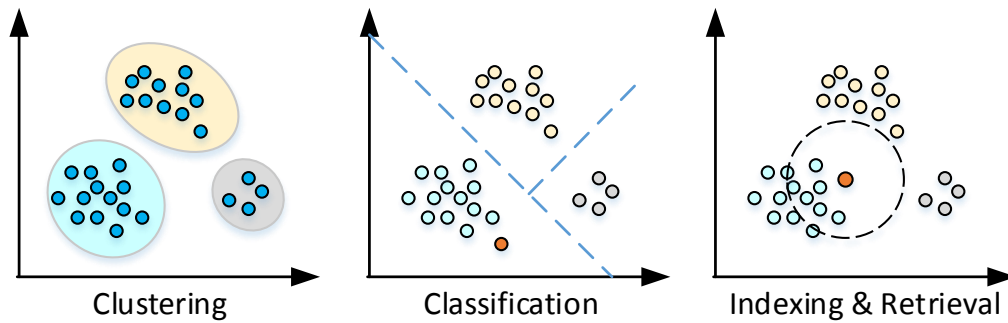


Figure 2.1: Clustering (left) is used to identify natural groupings of similar data items. The data in the image is partitioned into three groups (as represented by the coloured regions). Classification (centre) uses a set of pre-defined classes (partitioned in the image) to determine the resulting classification of an unlabelled input (red circle). Indexing and retrieval (right) is the task of querying the data with an input sequence (red circle) to retrieve similar matches from a corpus of indexed sequences (non-red circles).

2.1.1 Contributions

The contributions of this chapter are:

- A graphical overview of state-of-the-art information visualisation techniques for time-oriented data.
- An introduction to temporal visual analytics for temporal mining researchers.
- The classification of temporal visual analytic techniques from the previous decade into the mining task at hand: classification, clustering, indexing and retrieval, pattern discovery, and segmentation. We further subdivide each grouping by the type of time-oriented data utilised applying the characterisation by Aigner et al. [AMST11]: event data representing changes in state (e.g. a log of animal behaviour) vs. state representing phases between events (e.g. acceleration channels regularly sampled) and multivariate vs. univariate. Our classification table highlights fruitful areas for future research.

The organisation of this chapter is as follows. In section 2, we introduce temporal knowledge discovery terminology. In section 3, we discuss data mining approaches for knowledge discovery. In section 4, we introduce information visualisation literature targeted at visual data mining. In section 5, we discuss visual analytics terminology. In section 6, we present our classification scheme and literature survey. Finally, in section 7, we summarise the current state-of-the-art.

2.2 Knowledge Discovery

Data is costly to obtain and maintain but can hold a vast amount of hidden information for decision making, understanding, and insight. Knowledge discovery is an essential task in unlocking hidden information and gaining insight into data quickly and accurately to realise its value [CPSK10]. The terms knowledge discovery and data mining are often used interchangeably in the literature, although there is a clear distinction between the two. Knowledge discovery is primarily concerned with the tasks at hand, not the underlying process (implementation) for obtaining knowledge. While, data mining is knowledge discovery realised through algorithms.

Traditionally, transforming data into knowledge relied on manual analysis which is a slow, expensive, and a potentially subjective operation. An ever-increasing information space means a manual approach may be no longer feasible, and more advanced methods are essential. Data mining has received long-standing for performing knowledge discovery. Visualisation has more recently been widely adopted since the advent of visual analytics provides a bridge between the visualisation and data mining fields.

In this section, we discuss the relevant temporal knowledge discovery tasks applied. The data mining community often categorises these based on the primary tasks of classification,

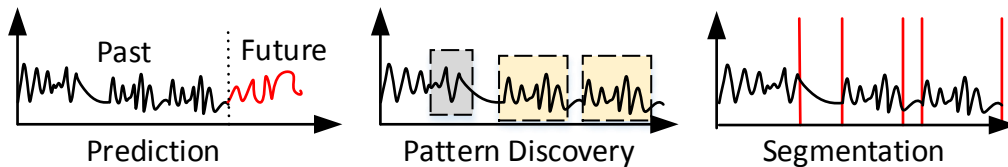


Figure 2.2: Prediction (left) is used to forecast future data values based on previous historical data. Pattern discovery (centre) is the task of finding interesting patterns, such as, anomalies (grey), and repetitive patterns (yellow). Segmentation (right) is used to divide the data into discrete segments (represented by the black vertical lines throughout the series).

clustering, indexing and retrieval, pattern discovery, prediction, and segmentation [AO01, LS06, BD13, Hsu07, Mit10]. Note, the tasks we are interested in operate on a singular time-series (Q) for intra time-series tasks (operating within the input series), we do not explore inter time-series tasks (operating between multiple time-series) although this is also an interesting area of research. We now introduce each of these tasks.

- Classification is the task of determining the corresponding classes of an unlabelled input sequence (Figure 2.1 centre). Given a predefined set of classes E and an input data series Q . The task is to label sub-sequences in the input (Q) to one of the classes (c) contained in E , such that $c \in E$.
- Clustering is the task of finding a natural grouping of data based on similarity to find previously unknown structure (Figure 2.1 left). Data objects are grouped such that objects in the same group are more similar than those in other groups, with each group of objects corresponding to a cluster. An input data series (Q) is segmented into a corpus of time-series signals (G) which are then mapped into a set of C natural groupings of the data, such that the distance between clusters is maximised while the intra cluster distance is minimised.
- Indexing and retrieval (query-by-content). Indexing is associated with organising an input series Q to enable fast retrieval when querying. Retrieval is the task of querying the data with an input sequence to retrieve similar instances in the data (Figure 2.1 right). Given an input time-series signal S and an input data series Q the task is to retrieve the set of the most similar matches of the input signal S within the data series Q .
- Pattern discovery is the task of discovering interesting patterns in the data without any prior assumptions (Figure 2.2 centre). This has traditionally been separated into two sub-tasks in the mining community. Firstly, anomaly detection to extract outliers or data items which do not confirm to the expected pattern in the data. Secondly, motif discovery to find frequently occurring repeated patterns in the data series. Given a data series Q , the task is to retrieve a set of sub-sequences C defined by the task at hand, typically frequently occurring sequences or anomalies.
- Prediction is the task of forecasting future data values in a time series given the previous data samples (Figure 2.2 left). Given an input data series Q , the task is to find future values in time $(n+k)$, where n is the length of the data series, and k is the number of samples into the future to predict the value of.
- Segmentation is the task of dividing the data series into discrete segments in order to reveal the underlying properties of its origin (Figure 2.2 right). The results are often used as prior input to other knowledge discovery tasks. Given a data series Q , the task is to divide the data into meaningful segments.

The realisation of these tasks is challenging and typically relies on complicated algorithms. Often solutions become black-box methods in the hands of the end users or the algorithms provide results that do not lead to a solution to the problem, because they omit relevant expert knowledge [KKEM10].

2.3 Data Mining

The data mining process typically consists of three stages; (i) preprocessing to acquire an appropriate data representation, (ii) selection of a metric which can extract desired properties from the underlying data, and (iii) execution of the mining operation / algorithm to obtain relevant knowledge from the data.

Preprocessing - Data mining techniques rely on the input data being of a suitable representation and quality. The term 'garbage in, garbage out' is often used to justify the importance of preprocessing, meaning, if inadequate quality data is given as input to a data mining process, the results will be of an equally poor quality. Real world data is often incomplete, noisy, or inconsistent. The objective of preprocessing is to manipulate the input data to gain an appropriate approximation of the underlying data representation by: cleaning (populate missing values, identify outliers / noise, and resolve inconsistencies), integrating (integration of other data sources), transforming (normalisation and aggregation into an appropriate format), reducing (minimising the representation of the data), and discretisation (partitioning continuous attributes). It may not be the case that none of these techniques are required, but often one or more combinations of these tasks are necessary.

Handling large temporal data efficiently is difficult without using large amounts of memory or processing power. A special form of feature extraction is often applied, called dimensionality reduction to reduce the footprint of the data series. Dimensionality reduction transforms an input series to a reduced representation by generating a higher-level abstraction of the underlying series. The abstraction must represent the data in such a way that the fundamental shape characteristics of the data can be retained, while reducing the dimensionality, such that, the same if not close to the same solution would have been obtained with the original data. These methods usually include, Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), Singular Value Decomposition (SVD), Piecewise Linear Approximation (PLA), Piecewise Aggregate Approximation (PAA), Adaptive Piecewise Constant Approximation (APCA), and Symbolic Aggregate Approximation (SAX). Sorzano et al. [SVM14] present a concise survey and introduction to dimensionality reduction techniques for temporal data.

Effective preprocessing of temporal data requires sufficient communication of knowledge from domain experts of the data characteristics (e.g. what constitutes normal values, anomalies, noise, etc). This step is often implemented as an automated method without the preprocessing results communicated for verification. However, this is important since, without an appropriate representation, specific attributes of the data may be removed or replaced, therefore producing incorrect results. Bernard et al. [BRG⁺12] introduce a visual analytic approach for time-series preprocessing to allow the expert to construct interactive preprocessing pipelines from which parameters can be adjusted through visual interfaces. Results are inspected throughout to determine the suitability and effect of each operation. Gschwandtner et al. [GAM⁺14] present TimeCleanser for cleansing time-oriented data to detect and correct corrupt or inaccurate instances in the underlying data. Cleansing requires intervention by domain experts to parametrise expected results of each attribute and how to resolve any inconsistencies appropriately. The TimeCleanser system detects problems with temporal data and guides the user through correcting them.

Distance Measure - Most mining operations rely on the notion of the similarity between data objects. A distance measure is used to derive a quantitative value corresponding to similarity or dissimilarity between two objects (Figure 2.3). The choice of distance measure is important when considering the ability to deal with outliers, amplitude differences, and time axis distortion [KP99]. The simplest form of distance measure is Euclidean distance, which computes the absolute sum of distances between adjacent points in the series. However, it is heavily subjective to noise, and cannot depict objects stretched or compressed along the x or y -axis [KR05a]. Other distance measures have been introduced to overcome these downfalls, such as, Dynamic Time Warping (DTW), and Longest Common Subsequence (LCS) although these entail a larger computational complexity $O(N^2)$. Serrà et al. [SA14] present an overview of relevant temporal distance measures.

Algorithm - Many algorithms have been introduced to support time-oriented knowledge discovery tasks. Since data mining is inherently task-oriented it is trivial to classify techniques based on the task-at-hand. We refer the reader to one of the many data



Figure 2.3: Distance measures are used to quantify the similarity between two data series (in this case, the top and bottom time-series graphs). Euclidean distance (left) calculates the sum of the absolute difference between adjacent data items. Dynamic time warping (right), stretches and compresses time to identify related items which vary in temporal duration. Image Credit: [KR05a]

mining surveys reviewing the state-of-the-art data mining techniques for temporal data [LS06, EA12, AO01, RLG⁺05, cF11, BD13, Hsu07, Mit10] for an in-depth review of the literature. We now present a brief overview of traditional data mining algorithms for each of the knowledge discovery tasks of classification, clustering, indexing and retrieval, pattern discovery, prediction, and segmentation.

2.3.1 Classification

Classification is a sub-set of machine learning called supervised learning. The learning process involves inferring a function which generalises relationships between input and output samples, such that, given a new observation, the function can be used to predict a membership to one of the predefined classes given in the training set. The membership prediction can then be applied to assign the input data to one of the predefined classes.

K-nearest neighbour (K-NN), support vector machines (SVM), decision trees, and random forests have all been applied to time-oriented data. K-NN is a lazy learning method which predicts class membership based on the k most similar training examples (defined by the distance measure) in the feature space. This method is often favourable due to its intuitive nature and speed. The SVM algorithm finds a hyperplane which separates the feature space into the classes defined in the training set. Unseen data is assigned to a class based on the hyperplane region it falls under. Decision trees generalise a set of rules from the training data, such that, rules are applied to unseen samples to obtain the classification. The resulting tree is often represented as a node-link diagram, where each node represents a test on an attribute and links between nodes cover the results of each test. The data set is partitioned by tracing through the tree to a leaf node which assigns the resulting classification label. Random forests construct many decision trees each modelling the training set. Each tree votes for the resulting classification. A data item is assigned to the class with the most votes. We refer the reader to Phyu [Phy09] for a general overview of classification methods.

2.3.2 Clustering

Cluster analysis is an unsupervised machine learning technique used to extract hidden structure in unlabelled data. Clustering categorises data into natural groupings based on similarity to find previously unknown structure. A quantitative value for the similarity of clusters is obtained using a distance measure (Section 2.3). The goal of clustering is to achieve a natural grouping which minimises intra-cluster similarity, while maximising inter-cluster similarity.

Partitional clustering is most commonly applied by the k-means algorithm. Data is partitioned into k clusters based on the distance to the nearest mean. The disadvantage of k-means is the need to specify how many partitions are in the data prior to clustering. Hierarchical clustering does not require any input parameters, instead computing the pairwise distances between all combinations of objects to group together similar data over a variety of different scales to form a cluster tree, often visualised as a dendrogram. The user chooses an appropriate level to cut at which defines the partitions. The mean shift algorithm is also non-parametric. Mean shift is an iterative algorithm, which considers the feature space as a probability density function. A kernel is iteratively shifted to a higher density region (and therefore the centroid of the data attributes within it) until convergence. We refer the reader to Berkhin [Ber06] for a general overview of clustering methods.

2.3.3 Indexing and Retrieval

Indexing is used to organise data, such that, a structure is obtained which enables the underlying data to be quickly located and retrieved when querying, without having to access and compare across every data instance. Indexing is commonplace within relational databases in the information retrieval field. Recently, time-series databases have been introduced which are optimised through indexing structures for handling and querying time series data [Keo06].

Retrieval is used to query raw data or a pre-built index to find instances which meet the query constraints. Often queries take the form of a data sequence where the user wishes to identify similar data objects, often this is applied by template matching. Matching can be divided into two categories, whole sequence matching, and subsequence matching. Whole sequence matching is performed on whole collections of data series to find other similar data series. Conversely, subsequence matching is performed to find matching regions within a data series. Subsequence matching can be generalised to whole sequence matching by extracting non-overlapping sections of the data-series [RCM⁺12].

A trivial method of applying sequence matching is to perform a linear scan over the data series computing the similarity of the input sequence at each location in the data, although this can be computationally expensive. Keogh et al. [KR05a] perform similarity search at different levels of granularities using a lower bound approach. A more efficient method is to transform the data series into an indexable structure which can be searched. The data is clustered together and organised in such a way that all the similar sequences are accessible by the same group. Vector based methods compress the data to indexable vectors. Metric based approaches organise the data based on relative object distance. Senin [Sen09] present a literature review of indexing techniques for time-series.

2.3.4 Pattern discovery

Pattern discovery is used to identify the presence of interesting patterns. Generally, this concerns, anomaly detection to discover surprising events, and motif discovery to extract frequently occurring patterns. The overall objective is to determine the presence and extract interesting patterns which either deviate from the normal behaviour in the series (i.e. anomalies) or contribute towards it (i.e. motifs).

Defining what constitutes normal behaviour is heavily domain specific and challenging. Typically, pattern discovery is a human centric task, as such this area often is less fruitful. A naive and computationally expensive approach for motif discovery involves computing the similarity between every combination of subsequence in the data series [LP02]. Many efforts have been applied to enhance the efficiency of extracting motifs, predominantly using dimensionality reduction [LP02]. Recently, motifs have been applied to discovering anomalies by looking at the inverse regions in the data which are not considered motifs and therefore anomalies [LKLC03].

2.3.5 Prediction

Prediction is used to forecast future data values based on historical trends. Prediction requires sufficient historical data, along with considering the different types of temporal patterns, such as, trend, cyclical, seasonal, and irregular patterns.

Classically, prediction has been based on statistical approaches which are often grouped as data-driven or model-based. Data driven approaches perform statistical analysis on previous data values to find the next value in the data. Exponential smoothing (e.g. Gaussian smoothing) is often favourable. This applies a weighted moving average across the data series with weights declining exponentially, such that, the most recent value is weighted highest. The result is the next data value in the series formed from combinations of historical values. Model based techniques build a predictive model from the data by inferring relationships. The autoregressive class of models are most commonly used for deriving relationships among time-varying processes. The model can later be queried to obtain future values in time. Commonly used regressive models include the autoregressive moving average model (ARIMA), and box-Jenkins method. Bontempi [BBTLB13] provide an introduction to forecasting methods.

2.3.6 Segmentation

Segmentation is used to divide data into discrete chronological regions. Generally, segmentation is domain specific according to underlying properties of the data to find the start and end boundaries of events. Segmentation is often used as a dimensionality reduction technique to represent the data series using only the relevant data transitions.

Segmentation approaches for time-series can generally be thought of as a line fitting problem to best represent data using k straight line segments. A predetermined criterion specifies the importance of a data value. Sliding windows grow a segment until it exceeds the set criteria. Top-down algorithms initialise as a straight line covering the whole series, and recursively split the data series until the stopping criteria is met. Bottom-up algorithms initialise as the set of segments covering the smallest possible size, each segment is iteratively merged together until the stopping criteria is reached. Keogh et al. [KCHP04] present a survey of segmentation techniques.

2.4 Information Visualisation

Visual data mining is a new term to indicate the application of visualisation to knowledge discovery [Kei02]. Visual data mining facilitates a human centric approach by transferring the role of data mining to the user through visual data exploration. The knowledge discovery process is integrated into the visualisation, where users directly interact and inspect data to draw conclusions and insight from visuals. This fusion often allows faster exploration and can provide better results in comparison to data mining methods [Kei02]. This is largely due to the flexibility offered by allowing the user to explore the data selectively while directly applying domain knowledge. Furthermore, an entirely transparent methodology gives users a higher degree of confidence in the results they obtain (Chapter 3).

The visualisation community is more concerned with the application of solving domain level problems by allowing users to undertake many roles implicitly, and not explicitly task oriented like in data mining [KKEM10]. Primarily, visual data mining is concerned with allowing users to explore, analyse, and present interesting patterns. The use of visualisation has been widely applied to the result of data mining algorithms [JBMS09, HMJ⁺12], however, this is beyond the scope of this chapter where we are interested in methods which empower users to undertake exploration, analysis, and presentation. The visual discovery of patterns in time-oriented data is an established research area in information visualisation. There have been a number of surveys of visual data mining techniques [Kei02, DOL03] and time-oriented visualisation methods [AMST11] which we refer the reader to for further detail.

Playfair was one of the earliest pioneers for visualising temporal data introducing the line plot in 1786 [Pla86]. The line graph is ubiquitous [JME10, Tuf01, MMKN08], and is the foundation of many time-oriented visualisation techniques. Data items are plotted as a series of points on a two-dimensional plane joined together in chronological order using straight lines to illustrate temporal patterns. However, when applying Playfairs method to large data, an occlusion and over-plotting problem occurs. That is, there are more data items than pixels available which means data items are effectively plotted on top of each other in the resulting visualisation causing occlusion. In this section, we provide a summary of time-oriented visualisation approaches and associated methods which target the large data landscape commonplace today. Throughout this overview, we introduce a graphical overview of extensions to the line graph. Generally, this is through alternative layouts, aggregation, and lens based approaches.

To aide our illustration of the occlusion problem, and understanding how visual approaches represent time, we utilise terminology from Isenberg et al. [IBDF11] who present a generic framework to provide design guidelines for charts containing dual scales to enhance data perception. A generalised transformation function is introduced which describes any chart showing two domain scales, which involves mapping data space to a display space. We utilise their terminology to describe the current literature and apply their work to describe any time-series chart featuring multiple scales and layouts. We have implemented all the methods in our software in a consistent way and generate automatically the associated time domain to x-axis transformation functions as corresponding graphs. These are exported to the Scalable Vector Graphic (SVG) format and incorporated into this document as an illustration of the method and of our classification.

We use for our data a 15 minute (approximately) sub-section of remote animal monitoring data obtained from a deployment of a behavioural data collection tag on a Cormorant (an aquatic bird) consisting of 34,746 data items collected at 40Hz. Each of the visualisation methods utilise 400 pixels width, which leads to an over-plotting ratio of 86 data items per pixel in the traditional time-series graph (Figure 1.2). This is problematic when considering a single Cormorant wing beat occurs at a frequency of approximately 5Hz [Pen01]. The problem worsens when presented with larger data (e.g. the full deployment of 857,407 items).

2.4.1 Layout Based

Layout based techniques modify the spatial arrangement of time while providing a linear mapping which transforms time-series graphs to reveal hidden temporal structure and patterns between dimensions. TimeWheel [TAS04] couples each data attribute with time by visual arrangement of radially arranged parameter axes around a central time axis. Lines are drawn for each time step from the time axis to associated points on each attribute axes to depict temporal trends between dimensions. Space time cube [Häg70] utilises three-dimensional space to map spatial attributes to a two-dimensional plane and time along an additional axis. Events or transitions are mapped across space and time by position in the three-dimensional space to show temporal trends across space and time [KW05, Kra03, GAA04]. A spiral arrangement of time-series encodes the cyclic nature of time to illustrate periodic behaviours [WAM01], and extended to spatio-temporal data using spiral glyphs placed on a map display to show periodic patterns across space [TSWS05]. CloudLines [KBK⁺11] adopt a logarithmic temporal axis to accentuate more recent temporal events and divert attention from older elements. Time is not constrained to a spatial position and can be encoded through geometry (e.g. shape), visual channels (e.g. colour), and dynamic representations (e.g. animation). While these techniques are powerful for the aspects of time they seek to encode, when applied with large data sets they fall short, since there is no capability in the methods to adapt to the size and scale of the data represented, thus when presented with large data an over-plotting and occlusion problem occurs.

Zoom plot embeds a time-series graph within a zoom-able widget of which the user has control over the zoom level to define the level of granularity they require for inspection (Figure 2.4). As the zoom is increased, the width of the time-series graph is expanded. The display view-port remains constant. Scroll bars allow the user to scroll smoothly through the expanded time-series. This allows the user to navigate through a large data landscape at the required resolution, however, context is lost with the remainder of the series either side of the zoomed region.

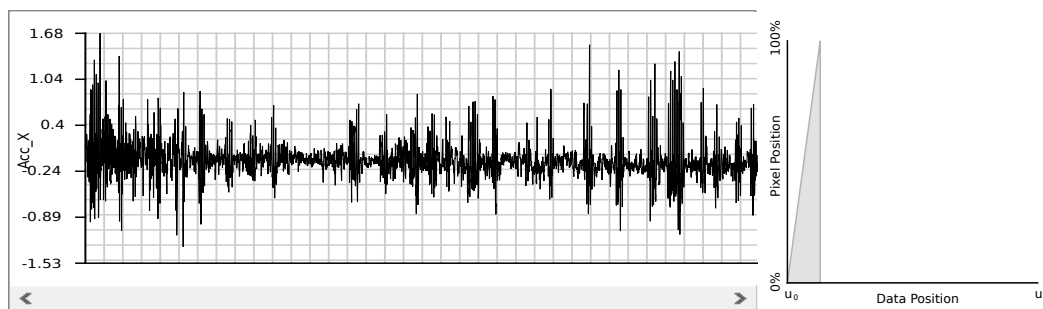


Figure 2.4: Zoom plot and temporal to position map. The zoomed in view-port is displayed as a linear function on the map of the visible data range.

Overview and detail displays [PCS95] provide a multi-coordinated view of a focus region, along with an overview of the entire data series which provides context to the focus. A context line plot (Figure 2.5 top) visualises the whole data set, from this, users can brush areas of interest to inspect in further detail. Selected data subsets are visualised in a separate focus line plot display (Figure 2.5 bottom), showing the data in more detail while still maintaining context with the whole data set. While the user can perceive the temporal context of the focus region with the rest of the series, it is not possible to view and compare multiple regions across the series, an important task in time-series analysis [Shn96].

Javed et al. [JE10] present stack zoom, a multi-focus zooming technique (Figure 2.6). Multi-focus zooming maintains context and temporal distance whilst zooming. User selection

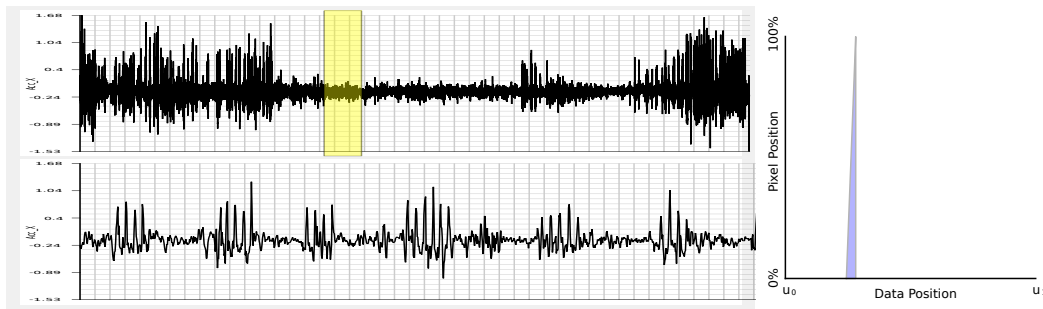


Figure 2.5: Overview and detail plot [PCS95] and temporal to position map. The overview is represented as a linear mapping of the entire data range. The detail is represented as a disjoint region overlaid on the map for the data range selected.

creates a hierarchy of zoomed line graphs, represented in a nested tree layout. Graphs are stacked on top of each other with the whole data set shown in a line graph at the root node. Each higher-level zoom is represented as a new child node, stacked below the parent node. A layout manager maintains that the nodes on each level are temporally ordered. Colouring and arrows are used as visual cues to illustrate the positioning of child nodes in relation to their parent. The tree structure serves as a graphical history for communication purposes.

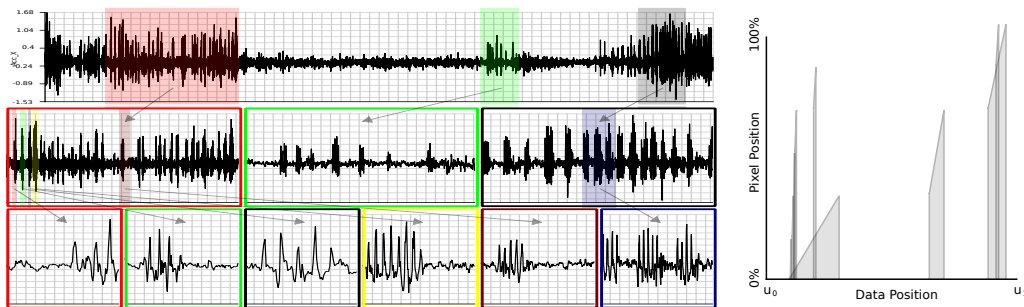


Figure 2.6: Stack Zoom [JE10] and temporal to position map. Each node in the hierarchy is represented as a disjoint region overlaid on the map for the data range occupied.

2.4.2 Histogram Based

Histogram based approaches applied to temporal data typically aggregate data points into segments of time. Rendering large data in a line graph implicitly aggregates time together (via over-plotting) but in a non-meaningful way representing a fraction of the underlying data per pixel. Effective aggregation depicts statistical features of the items in each segment of time through a meaningful visual mapping. TimeHistogram [KBH04] extends two-dimensional histograms to three-dimensional space to encode temporal changes in attributes over time. BinX [BM04] utilises adaptive binning across a line chart to overlay statistics of each segment of time across different temporal scales using line plots, box-plots, and min-max bands. CircleView [KSS04] adopts a radial display to encode multiple data attributes of a user-defined number of time slots. Colour represents aggregated windows of time for each attribute to enable comparison of continuous data streams between temporal ranges. Keim et al. [KKA95] build a recursive pattern display to highlight the hierarchical structure of time-oriented data by depicting data at multiple levels of granularities at each hierarchical depth. This allows the user to gain an overview of large datasets which can be recursively drilled-down to obtain further details-on-demand.

Pixel based displays represent a time-series as an arrangement of pixels encoded with different hues which encode the underlying data. Kincaid et al. [KL06] apply a pixel based display to multiple time-series graphs. Each time-series is split into uniform segments of time, such that each pixel in the visualisation is assigned a time segment. Figure 2.7 shows the maximum value in each time segment mapped to a yellow to blue colour hue. Each bin is one pixel wide, resulting in 400 bins, with each segment containing approximately 86 data items.

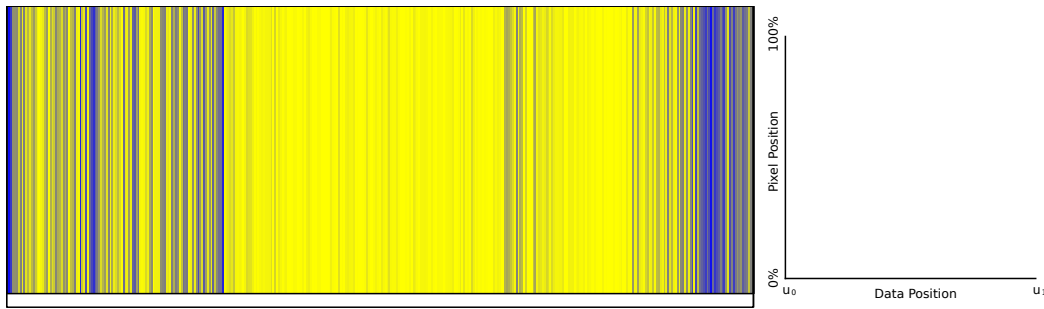


Figure 2.7: Pixel Plot [KL06] and linear temporal to position map.

Hao et al. [HDKS07] extend the one-dimensional pixel display to two-dimensional space utilising the x and y axes. Each segment of time is represented by a colour cell chronologically ordered navigating from bottom to top, and left to right. By utilising two-dimensions, the visualisation efficiently occupies space, which increases the number of time segments which can be displayed resulting in less loss of information when summarising. Figure 2.8 shows the maximum value in each time segment, each bin occupies 1 pixel width, and 3 pixels' height. The resulting visualisation contains 34,746 bins, which leads to 86 data items per column and 1 item per 3 pixels for this data size. In addition to the horizontal data mapping there would be a saw tooth like vertical mapping (not shown).

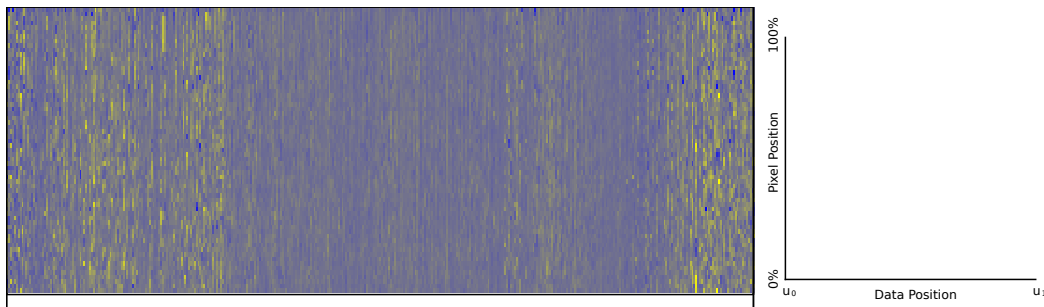


Figure 2.8: Two-dimensional Pixel Plot [HDKS07] and linear temporal to position map.

River plot (Figure 2.9) [BPS⁺07] depicts a number of statistical properties of time-series predictions. A bounded blue area connected through time represents the minimum and maximum bounds for each time segment. A black central line depicts the mean value for each segment of time.

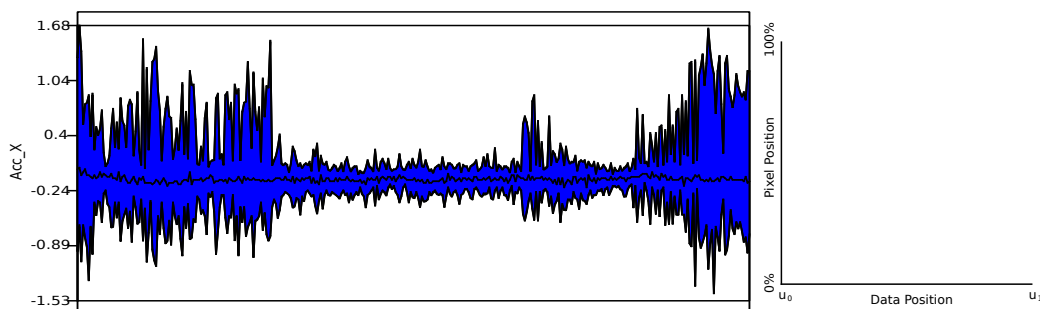


Figure 2.9: River Plot [BPS⁺07] and linear temporal to position map.

For smart sensor data the underlying data signal needs to be encoded to be able to extract animal behaviour, simplifying the underlying time-series to visualise statistical summaries of time-boxed regions does not give this information.

2.4.3 Lens Based

Lens based approaches provide on-demand alternative visual representations of the data underlying user-defined regions through interaction with the visualisation. Typically, this

entails time-axis distortion to enhance segments of interest, while maintaining context with the remaining visual [Kin10] and additional statistical summaries of the underlying data [TSAA12]. Brushing and filtering allow the user to select specific data ranges to accentuate in the data [BC87] and multi-linked views to combine different visualisation techniques together [Rob07].

Kincaid et al. [Kin10] present SignalLens for the visual analysis of electronic time series. An in-place magnification is added to the time-series plot which distorts the time-axis to magnify areas of interest. The data either side of the magnified area is compacted to maintain context, while allowing the inspection of low-level details of interest. A number of lens functions are introduced for comparison: linear, cubic, quadratic, hyperbolic, spherical and Gaussian. Figure 2.10 illustrates SignalLens applied with a linear magnification function.

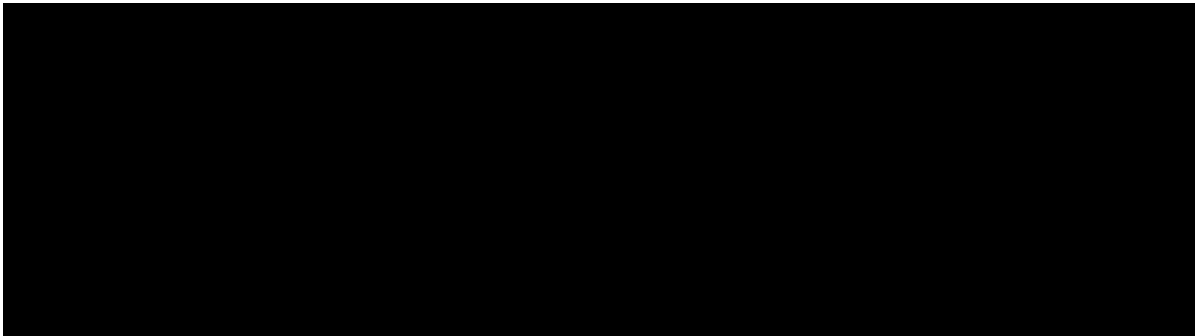


Figure 2.10: SignalLens [Kin10] and temporal to position map. The temporal to position map represents this region as a higher linear gradient.

High magnification generally requires a smooth drop-off to avoid occlusion in the context region. Kincaid et al. [Kin10] introduce a smooth visual transition option between the focus and context regions (Figure 2.11). This entails three zoom levels, the focus area which is of a fixed magnification in the centre, a lower magnification drop off area either side, and finally the context displayed in the remaining space.

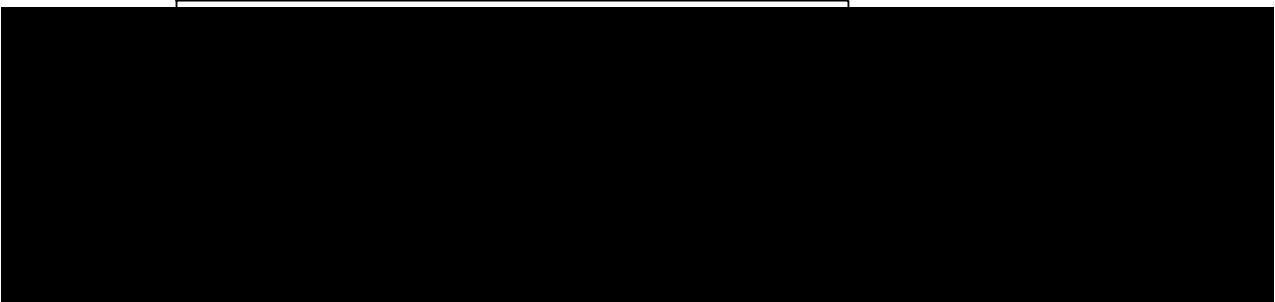


Figure 2.11: Smoothed SignalLens [Kin10] and temporal to position map. The temporal to position map represents this region as a higher linear gradient.

While, histogram approaches generally encode statistical summaries of the data, they could be used to encode areas of activity which may be interesting as a first level before delving deeper into the data series. We include a hybrid of the SignalLens and river plot displays to augment the statistical river plot approach with details-on-demand, we call this combination a 'RiverLens' visualisation. The user is presented with the river plot which provides an overview of the series. Brushing temporal regions expands a time-series plot, with an overlay of the river. The river plot is shown either side to provide context (Figure 2.12).

The previous lens techniques only allow one focal point to be explored simultaneously with a lens. Zhao et al. [ZCPB11] present Chronolenses, an interactive, visual analysis lensing technique to support more elaborate data analysis tasks, without the need to derive new time series visualisations (Figure 2.13). Multiple lenses are over-laid on top of traditional time-series graphs by user-selection and can display derived data, such as derivatives, and moving averages. Zoom, resize, and movement operations applied to lenses are used to overcome occlusion by magnification of the time-series by on the fly transformations of data points.



Figure 2.12: RiverLens and temporal to position map. The zoomed in region features a lower-density of data items per pixel. The temporal to position map represents this region as a higher linear gradient.

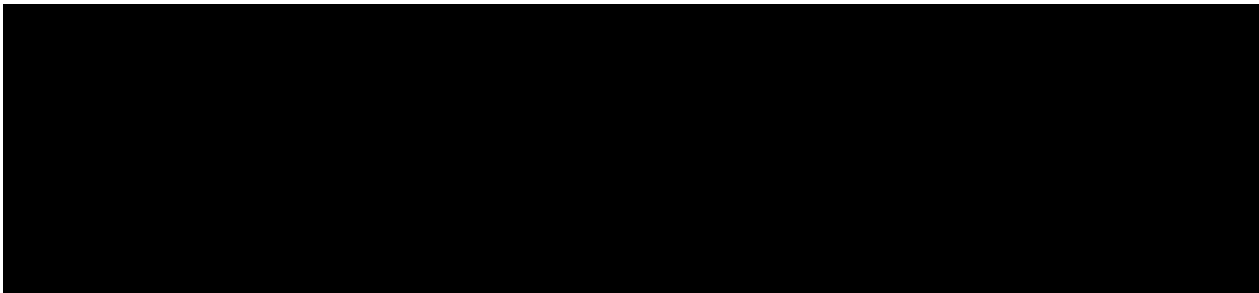


Figure 2.13: ChronoLenses [ZCPB11] and temporal to position map. Each lens is represented as a disjoint region overlaid on the map for the data range occupied.

Lenses are an effective technique for obtaining details-on-demand interactively in a visualisation. However, the screen real estate itself is constrained by the number of lenses which can be applied simultaneously before an occlusion and over-plotting problem occurs between the lenses [ZCPB11].

2.5 Visual Analytics

Visual analytics is at the core of dealing with vast amounts of information by combining together the enormous processing power and storage capacity of computers with the flexibility, creativity, and domain expertise of humans through interactive visual interfaces [TC06].

Including human intelligence early in data analysis is indispensable. However, humans have difficulty navigating through large amounts of information. Machines have been effectively utilised in the data mining community for processing massive data. Visual analytics applies interactive visualisation to unite humans and machines together to provide a transparent analysis process which the user can examine throughout to steer analysis of large data.

Figure 1.1 illustrates the knowledge discovery work-flow realised through visual analytics. The human is engaged with analysis models through interactive visualisation with the underlying data. Users apply domain knowledge through visual interfaces to iteratively refine models. A user-in-the-loop approach allows the user to delve deeper into the data to continuously build and apply knowledge. The integration of the human and machine is often much faster than traditional analytical and data mining techniques [KKEM10].

Visual analytics is inherently highly multidisciplinary cutting across many fields including, data mining, data management, visualisation, interaction, statistical analysis, scientific analysis, and human perception and cognition [KMS⁺08]. Literature reviews of visual analytic systems have been presented for commercial systems [ZSB⁺12], temporal-spatial data [AAD⁺10], and in general to provide an overview [KKEM10, SWLL13]. This review fills a gap in the literature reporting the combination of data mining and visual analytics to fulfil the task of extracting knowledge from complex abstract temporal data sets. The combination of complementary techniques from these two fields is an interesting research direction to

		Classification	Clustering	Indexing / Retrieval	Pattern Discovery	Prediction	Segmentation
Event	Univariate		[WSSM12]		[WaCPR11] [LAB ⁺ 14] [BAG ⁺ 15]		
	Multivariate	[CISSW06] [vdEvW11] [XCH ⁺ 14]		[FKSS06] [CGK ⁺ 07] [JS10]	[MTJ ⁺ 07] [LSS09] [DNKS10] [KMJE12] [WLY ⁺ 14] [ZCW ⁺ 14]		[HMJ ⁺ 12]
State	Univariate		[VWVS99] [ZJG ⁺ 10] [HSCW13]	[KP99] [HS04] [BAP ⁺ 05] [RLL ⁺ 05a] [HDK ⁺ 07] [HF09]	[LKL05] [LLO12] [SKU ⁺ 12] [SLW ⁺ 14] [RLRS15]	[BPS ⁺ 07] [HJM ⁺ 11] [BAF ⁺ 13] [BAF ⁺ 14] [BAF ⁺ 15]	[ABG ⁺ 14] [RLS ⁺ 14] [RLK ⁺ 15] [GSB ⁺ 15]
	Multivariate		[TPRH11]	[WB98] [SZWK12] [LCP ⁺ 13] [KWS ⁺ 15]	[DAW13]		

Table 2.1: An overview and classification of visual analytics methods for abstract temporal data. The survey is grouped horizontally according to task-at-hand (classification, clustering, indexing and retrieval, pattern discovery, prediction, and segmentation). Each group is then vertically sub-divided by the number of variables (univariate vs. multivariate), and the type of data (event vs. state). We refer the reader to Section 1.1.1 for an explanation of these. The entries of each sub-group are placed in chronological order.

solve challenging problems in the temporal data domain.

2.6 Literature Classification

We organise the literature of this review based on the primary knowledge discovery tasks of, classification, clustering, indexing and retrieval, pattern discovery, prediction, and segmentation (Section 2.2). We conduct the literature review based on the previous decade of literature at major visualisation venues, namely Visweek, EuroVis, PacificVis, and the journals IEEE Transactions on Visualisation and Computer Graphics (TVCG), and Computers & Graphics. We further search through the related work in the initial paper selection to obtain a wider coverage of literature. Table 2.1 presents a visual break-down of the literature, further subdivided into the types of time described by Aigner et al. [AMST11]. We firstly categorise by the type of temporal data utilised, consisting of event and state data. This is then further divided at the second level into number of variables, univariate and multi-variate. Additionally, each reference is colour encoded by the scale of the data, quantitative and qualitative. In this review, we exclusively survey abstract temporal data. Techniques containing an inherent spatial coupling are not included, rather the reader is directed to a prior survey [AAD⁺10]. We now classify the literature based on the task at hand.

2.6.1 Classification

Classification is the process of predicting which set of pre-defined categories a new observation is a member of. The classification process relies on the basis of training a classifier, which infers a function from a data set containing already labelled observations. Typically, the classification process requires extensively labelled data collections to learn relationships between the input data and resulting groupings. It may be the case that already labelled data does not exist in the domain area (i.e. in biology this data is often infeasible due to operational constraints), in the machine learning landscape it is not possible to proceed without this data available. When training data is available, the user must specify which features of the data to utilise for the classification. The results of the classification process rely on the training process (feature selection and set of training data), typically this is a ‘black-box’ process, without any transparency on how input signals are mapped to a classification group. If the results are incorrect or less than desired it is unclear without sufficient knowledge whether investing more time creating additional training input, modifying parameters, or

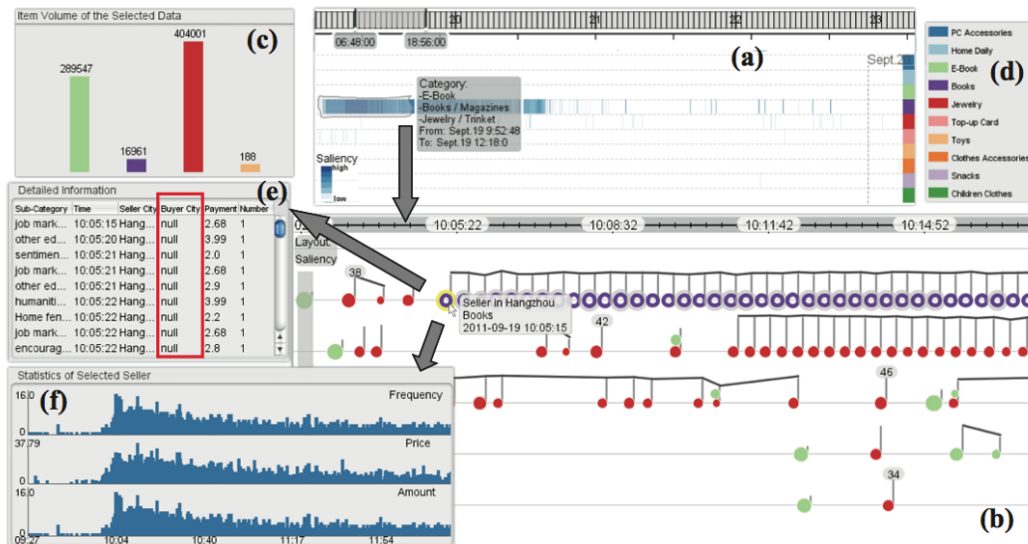


Figure 2.14: The VAET system highlights interesting events in e-transaction data. Initially all data items are displayed, as the user highlights interesting transactions in the visualisation the relevant information space utilised is adjusted accordingly. An overview of saliency values are encoded in a pixel-based display. A knot view encodes each transaction, with lines drawn between to represent clusterings. Image Credit: [XCH⁺14]

using a different learning algorithm is beneficial.

Decision trees are traditionally created by setting parameters, executing a tree construction algorithm, and evaluating the constructed tree, in a trial and error process, tweaking the parameters on each iteration. Often those constructing the tree hold a vast amount of domain knowledge, but lack expertise in the underlying algorithms. Elzen et al. [vdEvW11] present an interactive construction and analysis tool for decision trees which allows domain experts to apply knowledge to steer the construction process through a combination of visualisation and automated support, extending the work of Ankerst et al. [AEEK99] to support pruning and analysis tasks. Interaction incorporates the underlying tasks to, grow the tree manually by defining split points or with algorithmic assistance, optimised with the application of domain knowledge to change attributes and fine-tune the layout, and finally pruned to generalise the tree to prevent over-fitting through merging and deleting sub-trees. Visualisation is used throughout to provide insight into the data, tree (mapping function), and resulting classification.

Discovering interesting and relevant transactions in e-transaction event data requires analysis of subtle, inter-leaving, and varied data. Xie et al. [XCH⁺14] present a visual analytic system which leverages classification to assist users in identifying relevant transactions (Figure 2.14). The user marks relevant data features and transactions through interaction visualisation. A probabilistic decision tree is constructed which assigns a relevance value for each transaction the user has specified as relevant. The results are rendered in a pixel-based display of the encoded as coloured pixels ordered by time and partitioned by the users selected features to obtain an overview of their contribution towards the classification. Transactions selected in the pixel display are shown in detail in a knot view which renders each transaction on a time-line to encode information of associated transactions, lines are drawn between each transaction to identify clusters. An incorporated feedback loop allows the user to continue marking transactions as relevant, when the user does so the decision tree is adjusted, and therefore the visualisations are updated with the new classifications. This combines data exploration with iterative visual analysis to aide the user in identifying features of the data to perform classification on.

Chen et al. [CISSW06] study conflicting opinions gathered from a corpus of book reviews collected over time. In this domain area, it is challenging to identify what terms are useful for distinguishing between positive and negative reviews and how they affect the underlying

classification. Firstly, the underlying topics of each review are extracted and clustered using TermWatch [ISS04], which provides dimensionality reduction on the corpus. Based on the dimensionality reduction of topics, the user selects relevant terms to grow a decision tree through coordinated arc and node-link diagrams which allow the user to explore the dynamics of the conflicting opinions and select different terms to assess their suitability for performing classification to distinguish between positive and negative reviews. The resulting decision tree is visualised and presented to the user to explore the ways to reach a positive or negative review given the terms selected. This enables the user to discover and verify the most predictive terms for classifying the reviews, aided by statistical analysis using a log likelihood test.

Reflection: This collection of papers largely focus on the decision tree algorithm, unsurprising given this is considered the state-of-the-art process for classification in the machine learning community. Wijk et al. [vdEvW11] investigate exposing the construction of the decision tree to the user, allowing them to apply domain knowledge in constructing the classifier. This further exposes the decision tree process to domain areas where there may not be any training data available. Although their tool also encapsulates constructing a decision-tree semi-automatically from data and allowing the user to verify and modify the tree. The remaining papers study selecting features of the data to use for the decision tree construction process and communicating how the selected features impact the tree and therefore final classification results for book reviews [CISSW06] and e-transaction data [XCH⁺14]. Current solutions are disjoint and do not combine the concepts of classifier training, feature selection, and visualisation of results together into one unified framework, which is important since they have a combined effect on the results obtained. Furthermore, the feature extraction processes are domain-specific for event data and sets of opinions. The lessons learnt from these studies need to be applied to other domain areas or made generic for the wider research community. One area where we fail to identify any literature is enabling users to validate classification results and understand how to proceed if the results do not meet their analysis requirements. That is, should the user provide more training instances, use a different algorithm, or adjust the algorithm parameters. We believe more work is required in this area to fulfil the user intents when performing classification.

2.6.2 Clustering

Clustering is the task of grouping an unlabelled set of input data into similar distinct groups (clusters). The resulting clustering attempts to label objects in the same group which are similar, while minimising similarity between each grouping. This method of machine learning is referred to as unsupervised learning, since it requires no training data, instead, natural groupings are inferred directly on the input data. This area of research is particularly challenging since the underlying clusters provided are heavily dependent on the features selected from the input data to use for clustering. The input data must therefore also be of the correct format, and only contain relevant data to create clusters from. Traditional k-means and hierarchical algorithms rely on input parameters which must be identified in advance. More recent algorithms, like c-means are parameterless. Selecting the right algorithm and associated parameters is challenging in the machine learning field without knowledge of each underlying algorithm. After executing the algorithm results are obtained. Verification of the results, understanding how they were derived (i.e. why are the groupings obtained), along with how to adjust the results if they are incorrect is also challenging when utilised by non-experts.

Hierarchical clustering results are traditionally visualised in the machine learning field using a dendrogram [Eve06], a tree representing the arrangement of clusters obtained from executing the algorithm. Users then horizontally slice the dendrogram to obtain the clusters represented at that level. Dendrograms often become highly cluttered for larger data sets, and are constrained to a node-link layout which is not particularly insightful for identifying temporal patterns and trends. Wijk et al. [VWVS99] introduce a graph and calendar view visualisation which depicts the clustering results enabling the identification of patterns and trends occurring on multiple time scales simultaneously (i.e. days, weeks, seasons). Interaction with the visuals allows the user to select days to overlay, and find similar days using the resulting clusters. In effect, this allows the user to visualise the resulting hierarchical groupings without having to select the number of clusters to utilise, instead visualising the relationships. Should the user wish to adjust the time-spans or distance measures utilised

they can adjust them and receive the updated results. However, this is a trial-and-error process based on exploring the result set.

Wei et al. [WSSM12] study user interaction click stream behaviour patterns for gaining insight to inform future design decisions. Clustering the data allows analysts to perceive groups of how people use the system. Grouping and clustering the data requires domain input, since they will have specific persona's they seek to explore. Each click stream is mapped to a two-dimensional viewing plane using a self-organising map, coupled with a Markov chain probabilistic model to provide an overview of the structure of click-streams. Each click stream is visualised as a rectangle, composing of ordered segments coloured according to user actions. The analyst sketches distinct groups in the overview display which is then used as initialisation to constrain the k-means algorithm. The analyst then explores the resulting clusters, and utilises interaction to inspect, examine, and refine the clustering result. The refinement method of Turkay et al. [TPRH11] is applied to enable modifying the clusters which are then reused as input to the k-means algorithm.

Hadlock et al [HSCW13] introduce visualisation to explore the effect of different algorithmic parameters on the resulting clusterings produced for dynamic network data consisting of nodes and edges with associated temporal attributes. Typically, analysts wish to find temporal relationships to uncover stable, evolving, and unreliable facets. This is very difficult to extract since the relationships occur under different temporal structures which can only be extracted using multiple search parameters. A matrix visualisation provides a preview of cluster results in comparison to different cluster input settings (attributes, subsets of nodes, temporal ranges). Each cluster is represented as a time-series graph glyph with the background colour encoding similarity, lines between clusters to indicate connectivity. A time-series view shows brushed clusters disseminated to provide details-on-demand. This allows the user to execute multiple cluster parameters and compare results.

Reflection: Early work in the area has focused on visualising the results from executing clustering algorithms, to enable users to perceive the effort of different parameter settings through a trial-and-error process [VWVS99, ZJG⁺10]. More recently Wei et al. [WSSM12] allowed the user to initialise the k-means algorithm to indicate what they would like the process to achieve. The work of Turkay et al. [TPRH11] is introduced to provide a feedback loop, where the user iteratively refines the algorithm results based on the visualisation of results. Finally, Hadlock et al. [HSCW13] visualise the impact of feature and parameter selection on the underlying clusters obtained. An iterative approach allows the user to truly dictate what they wish to achieve from the analysis process. However, we still consider clustering to be an open problem, given the diverse nature of algorithms available. Knowing which algorithm and features of the data is best for a users' analysis requirements is still unclear for non-experts when applying clustering.

2.6.3 Indexing and Retrieval

Indexing is used to construct a structure which enables the underlying data to be queried efficiently. While, retrieval is the process of querying raw data or a pre-built index to find data objects which match an input query. We therefore sub-categorise index and retrieval into the separate operations to identify literature for each.

Indexing

Indexing temporal data is a complex task. Conventional techniques for indexing relational databases, such as, b+ trees are not useful for indexing data which spans temporal intervals. The most effective way to index temporal data is dependent on the task at hand, since to make retrieval efficient the data must be organised (indexed) in such a way that enables efficient querying. Furthermore, temporal data may be of different durations, along with a multitude of associated attributes. In the machine learning field there is relatively little literature on temporal indexing [CSP⁺14]. Although the utility of time-series databases has been recently introduced which have custom query languages to enable retrieval [Keo06].

It is of little surprise we fail to find any visual analytic approaches for indexing. It is not a user based task. We do not perceive any value in such a method, but do appreciate further work is required in the data mining community to advance algorithms to assist in the

retrieval process, using indexing is vital as visual analytics requires real-time and interactive algorithms for retrieving data from large corpuses.

Retrieval

Searching temporal data to retrieve relevant sub-sets has received long-standing in the literature. Searching temporal data is challenging when considering the size and scope of queries users may wish to perform. Standard querying of relational databases has relied on strongly formed query languages (e.g. SQL), which formulate queries to identify rows in or across tables which meet the conditions specified. Such structured languages are often not applicable for querying temporal data, where it is difficult to express patterns of interest formally. Retrieval over temporal data is more complicated since the structure and relations between rows (or time-index) often need to be searched which is dependent on what the user seeks to achieve. Standard methods for querying data are formed away from the data and provide little assistance in query formulation to determine if a region of interest in the data provides any insight. In this section, we decompose the retrieval methods into the most common types of retrieval for temporal data. These are, parameter based, search-by-example, and sketching.

Parameter Based Visual parameter based search was first applied on time-series data by Hochheiser et al. [HS04] through the introduction of the TimeSearcher application. Interactive time boxes are overlaid on time-series graphs to specify query constraints. Values included in the brushed time interval and data range specified are highlighted throughout the series. Multiple time boxes can be added to form conjunctive queries. Interaction allows the user to specify the sensitivity and flexibility of time-boxes by data enveloping, query inversion, time transitions, variable time boxes and angular queries. This interactive coupling of query specification overlaid on the time-series allows users to manipulate queries dynamically on the underlying data while perceiving the results.

Fails et al. [FKSS06] introduce a visual interface for searching temporal event data to identify the occurrence of events with specific time-spans. The user defines events using panels, specifying the design pattern of data attributes for each and time spans between. The event specifications are chained together to build a pattern of event sequences to query with. A results visualisation is used to explore discovered patterns using a table format, where each row represents a match and events are represented as circles with lines in-between illustrating the ordering.

Sanderson et al. [SWR⁺12] investigate query based analysis for dealing with large scale scientific data to query temporally evolving features, such as intra-time and inter-time patterns. Querying is achieved through range-based queries to narrow the scope of the search-space coupled with distribution based queries which search on distribution and frequency of the underlying data. Users sub-select entities based on distributions via a histogram display of the distribution and frequency of matches, the frequency of matches at each time step, and the distribution of variables that match the query.

Search-By-Example While filtering data using dynamic queries reduces the information space displayed, once filtered, analysts are often interested in finding repeated instances of specific patterns of interest they have discovered in the data. Buno et al. [BAP⁺05] introduce TimeSearcher 2 [BAP⁺05], an extension of the TimeSearcher 1 application [HS04] which introduces pattern discovery through query-by-example on the time-series data to find similar occurrences. Rubber band selection over the visualised data series allows the user to perform specific pattern search on the remaining data using Euclidean distance. Results are overlaid on the time-series and interactively refined through modifying tolerance levels. Stoffel et al. [SZWK12] further extend the TimeSearcher time-box concept [BAP⁺05] to define a target pattern by selecting a time frame and subset of dimensions to search. A multi-dimensional search algorithm is applied which allows the user to specify similarity across the whole series at user-defined attributes instead of just the start and end bounds of the time-box for one dimension. A recursive pixel based visualisation allows the user to explore patterns and results.

Commonly a fixed distance measure is used when computing the similarity between queries and data items. However, similarity is often subjective and dependent on the task-at-

hand. Keogh et al. [KP99] introduce relevance feedback which learns from users interaction to adjust the similarity measure to the users expectations. The user selects a query by selecting sub-sequences from the data. This is stored as lines along with associated weightings, initially weighted equally throughout. The user provides a rating for each of the results found which then adjusts the weightings in the similarity model and executes the search again. The process iteratively continues until the user is satisfied with the result set obtained.

Chang et al. [CGK⁺07] apply query-by-example to identify suspicious activity in on-line banking transactions. WireVis, achieves this by guiding the user to identify suspicious transactions by replacing otherwise blind queries with informed decisions based on patterns from visual exploration of keywords within each transaction. A keyword network view identifies relationships and distributions between keywords, while a heat-map view illustrates clustered relationships between keywords and frequencies to provide an overview. A strings and beads view shows the time and value of transactions for each account over time in detail. Once the user has found a suspicious transaction using these views they can query the data to identify similar transactions exhibiting potentially suspicious activity. Histograms show the distribution of hits per keyword. The user can interactively adjust the sensitivity of results included using a difference slider.

Filtering and zooming are often not sufficient for navigating large multi-dimensional information spaces. Hao et al. [HDK⁺07] introduce IVQuery, a combination of interaction with automatic analytical methods to assist in discovering interesting regions in the data and their causes. The user selects a focus area of a pattern of interest. Relationships between attributes are identified by searching for similar patterns in the data and highlighting correlations between attributes. IVQuery selects appropriate analytical methods to measure relationships between the data and attributes to display positive and negative correlations. The system allows the user to plug-in different methods and similarity measure across parallel coordinates, scatter plots, and time-series charts.

Sketching Queries are often formed using hard constraints to specify exact bounds to search with, either via parameters to define bounds or through query-by-example. Visual queries allow users to sketch what they would like to search for, although they are inherently approximate and users may therefore wish to incorporate flexibility in the underlying queries [Wat01]. QueryLines [RLL⁺05a] supports querying two-dimensional data sets by drawing lines on graphs. Bounded lines represent maximum and minimum search ranges, trend lines are used to find regions with a similar slope, and goal lines allow pattern specification. Lines can be assigned different constraints to define flexibility when performing matching. A preference line creates an idealised target to sort the results by, while soft constraints shows both matches which fully satisfy the constraints and those which come within a user-defined deviation.

The adjustment of similarity parameters and query specification are often separate entities in the pattern matching process. That is, search is performed first then similarity is increased or reduced to adjust the result set. Holz et al. [HF09] introduce relaxed selection to define similarity prior to search. The user provides a single gesture interaction using the cursor to determine the spatial and temporal deviations at local extrema points in the pattern. Deviations are visualised as circles defining the derived tolerances and overlaid on the matched regions in the data to indicate the similarity of a match.

Interactive search is traditionally undertaken directly in the visualisation they are presented in which are often not designed to explicitly support such systematic analysis tasks. Jin et al. [JS10] introduce VizPattern, a visual query system which adopts a comic strip metaphor to build temporal queries and perform search using icons and graphics extracted from visualisations. A comic strip metaphor using the marvel query language [JS09] allows the user to construct queries by dragging graphical elements directly from a visualisation implementing the VizPattern API into the search panel. Constraints are applied by defining the, series of events, and time-frame (absolute, time interval, specific times, and absolute time ranges). The results panel is used to illustrate results through the original icons, connected with lines to illustrate the temporal gap in-between events. Queries can be directly refined in the results view by repositioning and adjusting results.

Reflection: Retrieval has been long studied prior to the definition of visual analytics. TimeSearcher [HS04] formed the foundations for a number of studies into apriori search (search by example) for univariate time-series data so it is of little surprise we see an abundance of literature. In comparison, event data for retrieval has received less attention, prominently we believe this is because it is more challenging to define temporal searches for event sequences. Generally, apriori search is not flexible enough in this situation where the user wishes to search with extra degrees of freedom in relation to time. VizPattern introduces event search via a comic strip metaphor which aides in the search process for event sequences [JS09]. The performance of algorithms and tools is an issue for querying large data, especially for complex queries. Further work needs to be undertaken in this area to encapsulate indexing prior to the retrieval process to maintain a real-time visual analytic approach for the retrieval of large temporal data.

2.6.4 Pattern Discovery

Pattern discovery is the process of identifying the presence of interesting underlying patterns and trends in a large information space. This is a challenging area in the machine learning community, since defining what constitutes interesting is a domain specific task. Although, generalised techniques have been introduced to uncover anomalies, and frequently occurring patterns [LP02, LKLC03], typically these methods rely on selecting appropriate input parameters. The field of visual analytics is well suited to this user-centred task. We therefore split the literature into two further sub-sections. One concerned with general methods for motif and anomaly detection, and another concerned with domain specific methodologies which are centred around fulfilling the pattern discovery process in specific fields.

Motifs and Anomalies

Lin et al. [LKL05] introduce VizTree to provide an overview of the time-series data to assist in identifying frequent patterns, subsequence matching, and anomalies. The time-series is converted to a symbolic representation (SAX) which is then used to construct a suffix tree encapsulating the local and global structure of underlying time-series. The tree is visualised and augmented with branch width to encode frequency, which are chained together to depict temporal trends (Figure 2.15). Users query the tree to retrieve signals of a specific shape. Common patterns (motifs) can be discovered by selecting the thickest branches across the tree, conversely anomalies can be identified by selecting the thinnest branches. Such a visualisation is compelling since it provides a visual index of the underlying data. However, the process requires the length of the pattern be defined apriori.

Li et al. [LLO12] present a system for finding variable length motifs by grammar induction requiring no input parameters prior to search. The time-series is converted to symbolic representation (SAX) from which a grammar is inducted to identify hierarchical rules and patterns. Visualisation is used to display the grammatical rules representing the underlying time-series of motifs. The user examines the frequencies of rules and explores them to gain insight into the motifs and associated data. Senin et al. [SLW⁺14] further extend the GrammarViz [LLO12] system to incorporate parameterless discovery of anomalies in time-series data. They introduce sorting to the rules identified in GrammarViz, such that, motifs are identified by sorting the grammar rules according to the most frequent patterns. Anomalies are identified by sorting by least frequent patterns, along with visualising the grammar rule density to identify anomalies through visual examination. Colour is assigned to reflect the number of rules which are encapsulated at each point in time. This is powerful since the grammar process is parameterless, although the time-series is represented through textual rules which are difficult to relate back to the underlying signal they represent.

Finding interesting patterns in time-series usually relies on computing statistical summaries across user-defined time scales and starting points to reduce the result set returned. Sips et al. [SKU⁺12] introduce a visual analytic system which computes statistical values for all possible time scales and starting positions of intervals. A matrix view decomposes the time-series into a hierarchy, each pixel encodes statistical quantities for each time scale and starting position. The user visually explores the matrix view to detect interesting patterns, and query the matrix to extract detected patterns.

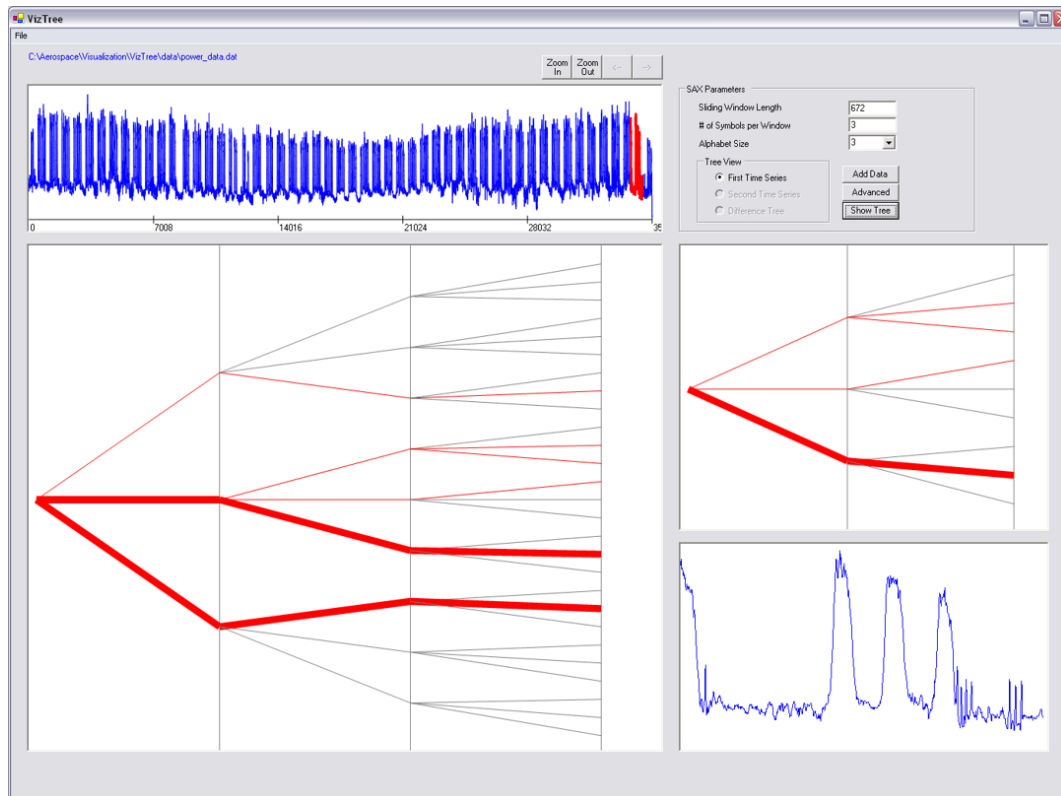


Figure 2.15: VizTree provides a graphical overview of time-series. A suffix tree encodes local and global structure with branch width representing amplitude frequency, chained together to show temporal trends. Image Credit: [LKL05]

Identifying relationships of repeated patterns across sets of time-series (ensembles) is typically performed using a windowed cross-correlation between two or more data series at different intervals and lags. This process can produce hundreds of thousands of matrices which is often infeasible to analyse. Köthur et al. [KWS⁺15] introduce a semi-automatic statistical analysis and visual exploration of large ensemble matrices. The results of the windowed cross-correlation are depicted using extended Hinton diagrams [BGD94] which statistically aggregate the matrices into a condensed view. Glyphs are used to show statistical values of the correlations and uncertainties for each aggregated cell, where the user can filter the matrices and obtain details-on-demand of areas of interest. This allows the user to visually explore the ensemble relationships, which can then be used to guide new parameter to search with guided by the insights made through the visual exploration of results.

Identifying temporal relationships occurring between variables in high-dimensional temporal data is challenging. One solution is to examine all pairs of variables as scatter plots, although this results in a large number of visuals to inspect. Dang et al. [DAW13] introduce TimeSeer (Figure 2.16) for the exploration of high-dimensional time-series data using scagnostics [WAG05], a method of characterising the statistical distribution of scatter plots by providing summaries of, density, skewness, shape, outliers, and texture. TimeSeer displays the derived scagnostics through matrix displays, small multiples, and super imposed scatter plots in line graphs. The user explores pairs of variables to discover anomalies by filtering, brushing and drilling down into the data.

Often dominant patterns which are already well known are over represented and clutter the results space when applying pattern discovery which by consequence hides less represented unknown patterns. Lammarsch et al. [LAB⁺14] introduce an event simplification process to reduce the number of unimportant patterns processed. This is achieved by adaptively combining events according to rules created by domain experts which merge repeated events together. Visual analytics is used to perform parametrisation of the event simplification process and inspection of the resulting patterns through river views, arc diagrams, and time-line displays.

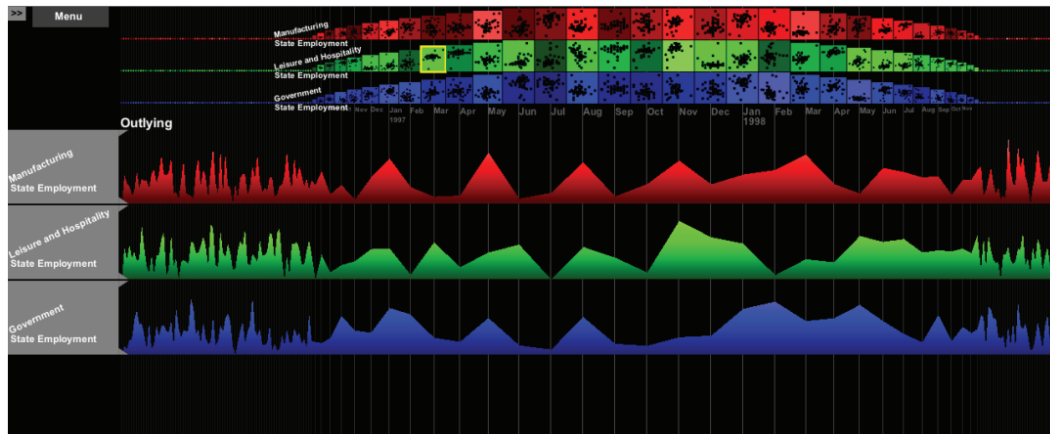


Figure 2.16: TimeSeer utilises scagnostics to characterise scatter plots of data attributes at each time index. The statistical summaries allow the user to explore pairs of variables to identify anomalies in the data. Image Credit: [DAW13]

Analysts need to understand the dependencies between events to select an analysis method which is appropriate to extract the desired patterns. Bodesinsky et al. [BAG⁺15] introduce a visual analytic approach for analysts to gain an understanding of the dependencies between events to select an appropriate analysis method to meet the users analysis requirements. Each event is represented as a rectangular bar coloured by event type, each event is then chained together to form event cases aligned horizontally, and vertically beneath each other. Arcs are drawn between each event to indicate repeated patterns inside a case to provide an overview of pattern frequency. Event cases can be ordered by time, frequency, and sequence length. Events of interest can be selected, grouped, and explored to understand the nature of the temporal event data. This allows the user to be suitably informed about the underlying data to then select an appropriate analysis technique. Although this process does not suggest how to make that selection, which requires domain knowledge of the underlying algorithms available.

Finding patterns across multiple categorical time-series relies on analysing and comparing between each series to find patterns and relate them. Row-wise alignment of each series can support users but only when an effective sorting is used. Richter et al. [RLRS15] provide a visual analytic approach to support the user in selecting effective sorting parameters. Firstly, an automated sequencing algorithm is executed which groups similar time series and forms transitions between groups, this creates an initial ordering of the time-series. The user steers the algorithm through interaction to choose between weightings to focus on different temporal aspects which adjusts the resulting ordered sequences.

Domain Specific

Pay-per-bid auctions operate by charging for each bid placed during the auction. Each bid raises the price of the item by a small increment and extends the duration of the auction. There is no optimal analytical strategy known to successfully bid. Walker et al. [WaCPR11] present a visual analytic system to explore different bidding scenarios and patterns which lead to a successful bid. In the resulting visualisation, each bidder is given a unique row, with each bid represented as a node aligned horizontally along the auction time-line. Scrolling and zooming depicts the visualisation at different levels of abstraction. The user can search the system for specific people and bid types to assist in refuting hypotheses. This allows the user to perform 'what if' analysis to uncover patterns which lead to winning an auction.

Social media is widely used by journalists to support the coverage of news and events as they emerge. The amount of content on social media can quickly become overwhelming as time progresses. Journalists seek to efficiently analyse this data to augment their reporting. Diakopoulos et al. [DNKS10] introduce a visual analytic system to assist journalists in extracting value and making sense of social media during broadcast events. A view of the

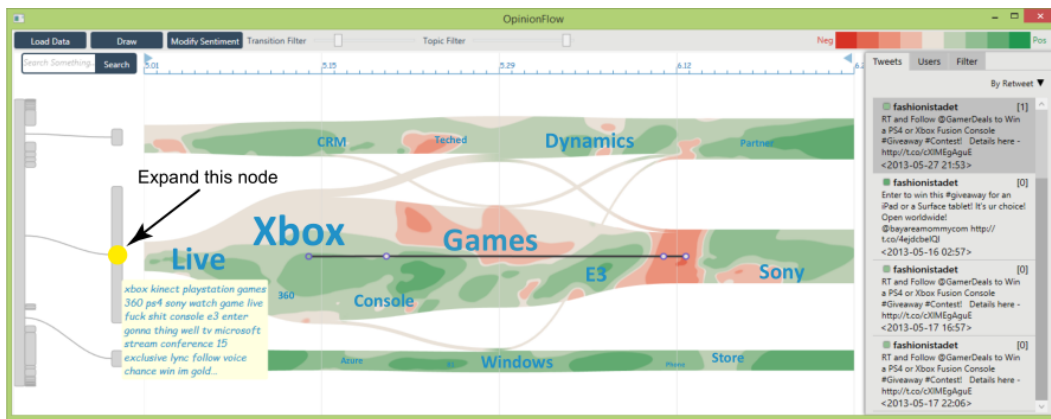


Figure 2.17: OpinionFlow depicts social media trends through a river type visual metaphor. Divergence and convergence in the flow illustrate how opinions propagate over time. This instance depicts opinion diffusion on the Xbox when the Xbox One was announced. Image Credit: [WLY⁺14]

media source (in this case a live news stream) is shown, along-side individual social-media messages captured. Statistical summaries of tweet volumes, ratios, and keywords are all visualised to assist in identifying patterns. A sentiment view aggregates the tweets of the segment response to positive, negative, controversial, and neutral. The user explores the visualisations and filters based on specified analysis measures. This allows the user to narrow down on interesting tweets and topics of interest to support their news coverage.

Public opinions rapidly propagate on social media. The diversity and rapid evolution of opinions pose challenges for effective and efficient analysis to assist in building response strategies. Wu et al. [WLY⁺14] introduce OpinionFlow, a visual analytic system to assist in detecting opinion propagation patterns and provide insight into their evolution (Figure 2.17). Existing analytical methods do not consider how opinions spread, operating on the corpus of opinions. An information diffusion model is used to capture the temporal propagation existing in social media. An enhanced Sankey diagram is overlaid with the derived propagation patterns using line integral convolution. Interaction with the visualisation allows the user to form and validate hypotheses about the diffusion. The user can feedback into the model to ask 'what if' questions by converting selected social media opinions.

Social media analysts often seek to pin-point anomalous information behaviours to identify interesting sources of information which deviate from global trends occurring. Machine learning is capable of detecting anomalies but it is difficult for analysts to trust or make use of results without understanding the underlying methods applied to obtain them. Zhao et al. [ZCW⁺14] present FluxFlow, a human supervised analysis tool to assist the user in understanding the analytical results of the one-class conditional random fields model used to derive them. The model computes an anomaly score for each conversation thread. To illustrate how a score was derived, feature space distributions are shown using a multi-dimensional scaling view. Thread glyphs show summary statistics for each thread. This presents the analytical results of the model, while allowing users to understand the data with interactive visual exploration.

Point-of-sale data is used to identify risks and forecast changes in market share for competitive intelligence. However, such data is difficult to comprehend efficiently due to its temporal, multi-variate and distributed nature. Ko et al. [KMJE12] introduce MarketAnalyzer, a visual analytics system to compare, explore, analyse, and predict trends from point of sale data. A pixel-based view efficiently utilises screen space to display the entire market, split into three categories to depict sales, trends, and growth predictions of each company. Companies can be compared to identify market risks, opportunities, and the impact of strategies. Multi-coordinated views of the data are augmented by trend analytics and forecasting which identifies regions of growth and risk.

Monitoring animal and human health can provide an early warning system for human

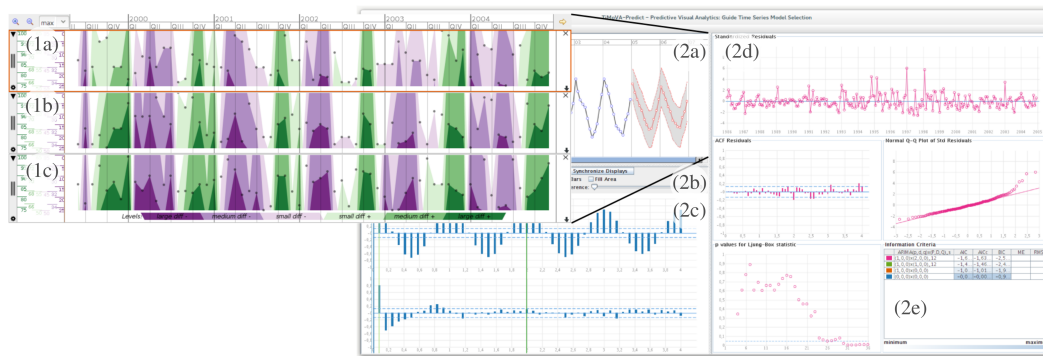


Figure 2.18: The TiMoVA system enables comparison of prediction models. The user is guided through selecting appropriate models and parameters using several visual aides. Image Credit: [BAF⁺15]

disease and provide fewer false alarms. Often this involves integrating several multi-field, multi-scale, and multi-source data sources. Maciejewski et al. [MTJ⁺07] introduce LAHVA, a visual analytic framework for analysing human-animal emergency data to provide event detection of disease outbreaks based on previous temporal trends. Data management normalises the data sources into a central relational database. An initial query of the data by the analyst is used to provide an overview. Statistical measures are integrated in a statistics view linked with a spatio-temporal viewing window. The analyst searches the data for regions of interest in both views with assistance from statistical summaries to build hypotheses. Once a problem has been visually identified the user can execute statistical analysis on data subsets to confirm or refute hypotheses.

Airline travel purchase requests consist of combinations of transaction events which terminate at a successful or failed purchase. Identifying abnormal purchase failures is of interest to analysts so they can resolve underlying issues which lead to a system failure. No precise metrics exist to distinguish between normal and abnormal behaviour in this domain. Current analysis involves performing unguided database queries with specific parameters, however, it is not possible for analysts to perform all possible combinations and test them one by one in queries. Liu et al. [LSS09] introduce SellTrend, a visual analytic system to analyse the distribution and structure of transaction failures. The overall interface acts a faceted browser for querying the data based on visual overviews and insights made into the data. Bar charts represent the failures across the time-line per day and hour. Selecting time periods update a table view depicting a spreadsheet of the raw data, along with a map view which encodes transactions along with categorical values of each attribute within. The size of each rectangle illustrates the number of failures for that value globally. Historical trend computation allows the user to compare the difference of the current value with historical values. Visually querying the data allows the user to identify patterns which lead to a failed transaction.

Reflection: Given the nature of pattern discovery being a human centric and domain specific task we unsurprisingly find the visual analytic field contains a wealth of techniques for undertaking pattern discovery. However, there are a number of generalised techniques for uncovering motifs and anomalies, where we also see a multitude of techniques in the machine learning field [LP02, LKLC03]. While the time-series approaches introduced are valuable for identifying motifs and anomalies, the appropriate methods and parameter specification for operation are still an open problem requiring more research to guide parameter selection. Li et al. [LLO12] introduce parameterless pattern discovery, however this requires data to be converted to a symbolic representation (SAX) prior to input which requires parameter length and segment size to be defined which have a direct effect on the results produced. Sips et al. [SKU⁺12] introduce true parameterless search executing across multiple time-scales and starting points, however, this comes with a much larger computational complexity, which does not give real-time results, essential for a visual analytics process [KKEM10]. Therefore, more work is needed in this area to provide true parameterless search as a real-time operation. We also see numerous methods for other data types; high-dimensional data [DAW13], and categorical time-series [RLRS15]. Event

data is also well studied where we see methods for collapsing events prior to execution to reduce irrelevant results being computed [LAB⁺14], and understanding event dependencies for algorithm selection [BAG⁺15]. Further work is necessary to unify these approaches, and investigate the application of other machine learning techniques supported by guided model and parameter selection.

A number of domain-specific visual analytic systems are introduced for pattern discovery in the fields of pay-per-bid auctions, social media, point-of-sale, human health, and transaction failures. Social media receives a large amount of interest, presumably since this is a relatively new discipline and the novelty for research and necessity for business to harness the value from the large amount of data present. We see research to support searching for relevant tweets [DNKS10], understanding how they propagate [WLY⁺14], and identifying anomalies [ZCW⁺14]. The literature holds value in the domains they solve pattern discovery within. Further work is required to see how the lessons learnt and techniques introduced can be applied to other domain areas where we fail to see any literature present (e.g. finance, and biology).

2.6.5 Prediction

Prediction is the process of forecasting future data values using historical data. The prediction process relies on utilising statistical measures or temporal patterns of past data to forecast into the future. Typically, the forecast value produced is heavily reliant on the model, and parameters utilised since these decide which attributes of the series contribute towards the forecast produced. Obtaining an accurate prediction without expertise and knowledge of the underlying prediction models and associated parameters is challenging. Model selection demands a combination of human judgement and automated computation.

Classical approaches for prediction are usually data driven or model based. An alternative approach are econometric models which assume a relationship between the current time-series and a set of historical time series measuring the same phenomena. Time-Searcher 3 [BPS⁺07] extends the TimeSearcher 2 [BAP⁺05] functionality to predict future values in time-series using pattern matching search to find all historical time-series similar to the current selected trend. Matching temporal regions are visualised through a river plot view analogous to a continuous box plot for illustrating the distribution of matches found, corresponding to the forecast obtained. Results are shown in a simultaneous preview interface which encapsulates multiple previews of different parameters using the river plot view. This allows the user to explore parameters and the effect on the resulting forecast.

Hao et al. [HJM⁺11] allow the user to steer the prediction process by applying domain knowledge. The Holt Winters prediction algorithm applies weighted moving averages to forecast seasonal trends using a function of past and current data through exponential smoothing. Smoothing can often be a compromise between removing noise and retaining valuable information. The system allows users to select interesting patterns from the time-series to apply peak preserving smoothing and utilise for the prediction process. A peak preserving simulation slider is used to reduce the points without missing peaks, and a weights slider adjust weights between peaks and time distances to use for the prediction. Accuracy indicators and certainty bands are applied, along with brushing and linking to directly compare predicted data to historic values. Visualisation of the algorithm results show an overview of past and predicted data values, coupled with connecting predictions to similar past trends. The user can interact with the model to adjust smoothing-parameters based on visual feedback of the resulting prediction.

Bögl et al. [BAF⁺13] introduce the TiMoVA system to assist in exploring and selecting an appropriate ARIMA (Autoregressive Integrated Moving Average) model to use for prediction. The ARIMA model generally consists of an autoregressive, integrated, and moving average part. ARIMA models can include account trends, seasonality, cycles, and errors to forecast future data, but this is dependent on difference and model order parameters selected. Firstly, the user decides on appropriate model parameters using complementary visualisations of an auto correlation function and partial auto correlation function to determine the behaviour of the model. The model is executed, with results immediately visualised. Model diagnostic visualisations consisting of summary statistics allow the domain expert to inspect how well the model fits the underlying data. Based on their findings, the analyst can refine the model

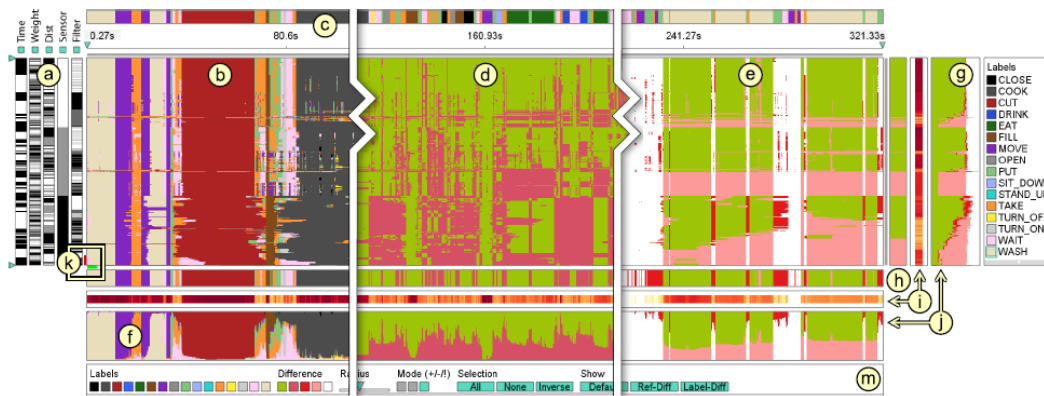


Figure 2.19: This visualisation supports the user in understanding the influence of parameter selection on resulting segmentations for activity recognition. Vertically aligned strips encode parameter settings with the respective segmentation results shown alongside. Compound views provide statistical summaries. Image Credit: [RLK⁺15]

ordering and associated difference parameters to obtain immediate feedback in the views.

While the TiMoVA system is powerful in the hands of experts who understand the statistical summary views. It does not enable the ability to perform prediction for domain experts to inspect the underlying results. Bögl et al. [BAF⁺14] extend the TiMoVA system [BAF⁺13] to include the ability to perform prediction and overlay the respective results to offer an alternative perspective on the accuracy of the model. A one-step ahead prediction, is visualised by fitting the selected model to the first half of the data. The second half is predicted, with a red line indicating the predicated data, confidence bands highlight uncertainty values with the actual data plotted using blue lines for direct comparison. The difference between actual and predicated values is indicated using vertical lines and fitted area graphs. This is a compelling visualisation which allows non-experts in the underlying ARIMA models to inspect the quality of their predictions. Multiple models are incorporated in a further refinement of the system by Bögl et al. [BAF⁺15] to allow prediction and comparison of multiple models simultaneously (Figure 2.18).

Reflection: This selection of papers focus largely on time-series prediction. Buno et al. [BPS⁺07] apply search to prediction which is a novel method of enabling non-experts to predict future values based on previous historical events. The process of predicting through interaction, exploration, and applying domain knowledge is highly suitable for visual analytics, although does require large a corpus of existing historical data to be effective. Where interaction is not appropriate, prediction models must be used which automatically perform the prediction process. Visual analytics is suitable in this area for assisting users in obtaining an appropriate model which can then be used automatically during analysis. Choosing the right model and associated parameters is challenging in the hands of non-experts which is where visual analytics has the potential to provide the most benefit. We see the remainder of the papers focused on this aspect, for the ARIMA, and holt winters algorithms [BAF⁺13, BAF⁺14]. A wealth of models in the machine learning field exist for prediction. We fail to find any literature for assisting users in model selection generally, although Bogl et al. [BAF⁺13] assist in selecting appropriate ARIMA models. We believe more work is required in this area to aide in model and parameter selection which still requires users to have considerable domain knowledge of the underlying algorithms to operate them effectively. Furthermore, we do not identify any literature for prediction of event data although there are obvious uses for such techniques (e.g. prediction of seizures in the healthcare domain).

2.6.6 Segmentation

Segmentation is the process of dividing a temporal data series into a sequence of discrete segments to reveal underlying properties. Segmentation is typically a domain specific task according to underlying properties of the data to be able to identify the start and end

boundaries of events. This is particularly challenging since data mining is more concerned with underlying algorithms and less so with its application to solve domain level tasks. The segmentation algorithms introduced in the machine learning field are highly dependent on the parameters selected which define the quality of the segmentation produced. Without a knowledgeable understanding of the underlying methods it can be challenging for the non-expert to insightfully select appropriate parameters. Furthermore, it is widely accepted in the data mining community that 100% accuracy is infeasible when applied to real-world data, therefore sufficient validation and adjustment of results by domain experts is required to achieve a high accuracy rate. As such, manual inspection of results and refinements are often approached through trial-and-error.

Alsallakh et al. [ABG⁺14] apply a visual analytic approach to the segmentation of multi-variable time-series data combining automated algorithms with interactive time-series visualisations to inspect, identify, and correct segmentation results by iteratively adjusting parameters and visualising their effect on the updated segmentation results. While this enables the user to see the direct cause and effect of modifying parameters it still has an element of trial-and-error where the user refines parameters until they achieve their desired result.

Rohlig et al. [RLS⁺14] support the user in understanding the influence of segmentation parameters and guide the user in fine-tuning the resulting segmentation produced. A set of parameters equally sampled within a specified range are created. Segmentation is performed for each instance, with the results visualised in a pixel based display, each segmentation is encoded horizontally as coloured bars in relation to the raw data, and parameters utilising the vertical dimension. The user orders the resulting segmentations to group together similar rows in the display. A compound view summarises the obtained labels to illustrate the most probable label, uncertainty values for each label, and a breakdown to show the distribution of assigned labels for each segment. Rohlig et al. [RLK⁺15] further extend this approach for determining the parameter influence on segmentation for activity recognition by allowing the user to explore hundreds of parameter settings, compare their influence, and given this make decisions on how to increase the accuracy of computed results (Figure 2.19).

There is an overwhelming amount of options for the user to select when undertaking segmentation; model, methods and multiple parameters. The previous solutions address these issues separately. Gschwandtner et al. [GSB⁺15] argue the need for a unified framework which ties these concepts together. They introduce the following conceptual model: Firstly, model building to select appropriate method. Secondly, model usage to identify appropriate parameters, and finally visualisation of results to illustrate uncertainty such that the user can evaluate and fine-tune the result set obtained.

Gad et al. [GJG⁺15] study the segmentation of temporal events, specifically identifying where and why change points occur in evolving topic streams. Data mining methods capture key breakpoints, however, communicating insight into why a change has occurred is an understudied area. A temporal segmentation algorithm is applied which extracts segments which characterise shifts in topics (clustering of topics, convergence and divergence). Visualisation is used to illustrate these change points in topics. A time-line at the bottom of the display is divided by each segmented change point. Transitions between each topic are illustrated by a discontinuous curved line which varies in shape, thickness and colour. The result is an interactive visualisation which allows the user to see how keywords scatter and gather to gain insight into how the segmentation algorithm derived the obtained result.

Reflection: Early solutions in the field introduced for time-series segmentation provide solutions for parameter selection [RLS⁺14] and facilitating the verification of the segmentation results [ABG⁺14]. Although this is compelling, these solutions operate independently of each other. Gschwandtner et al. [GSB⁺15] propose a conceptual solution which binds together the model selection, parameter selection, model execution, and validation, involving the domain expert to define the segmentation process at each stage. Realising this conceptual framework appears to be the next step in evolving research in this area. We only find the work of Gad et al. [GJG⁺15] for event based data studying the segmentation of complex evolving topic streams. Overall, the segmentation of event data is not a challenging

problem for trivial event data, since each event defines a natural transition which can be used for segmentation. It is therefore not surprising we do not find any further literature for applying event segmentation to visual analytics in this space where human intervention may be considered not as necessary except when the complexity of the data requires it (e.g. evolving topic streams [GJG⁺15]).

2.7 Discussion and Summary

Time is an inherently challenging attribute which requires support for specific user intents to extract knowledge efficiently and effectively. No single unified solution exists to support the user which is why we see an abundance of methodologies and tools to assist in undertaking specific tasks. We provide an overview of the existing methods classified by the underlying task to serve as a useful introduction to the field of visual analytics to demonstrate its application for solving difficult problems arising from time-oriented data. This chapter surveys the state-of-the-art of temporal visual analytics methods and to our knowledge provides the most up-to-date and comprehensive review of knowledge discovery methods for time-oriented data to date. Only relevant literature in the temporal domain is discussed although there exists a wealth of techniques beyond this scope. The task analysis of visual analytics for other domains is an area for future research.

Visual analytics is still a relatively young and emerging field, despite this, table 2.1 clearly illustrates that the tasks of pattern discovery and retrieval have received wide attention in the community. Pattern discovery is generally a human-centric and domain specific problem so it is of little surprise there is a larger number of literature in this area where visual analytics is well suited for this task. Indexing has been long studied prior to the definition of visual analytics. TimeSearcher [HS04] formed the foundations for a number of studies into apriori search (search by example) for univariate time-series data. In comparison, event data for indexing has received less attention, prominently we believe this is because it is more challenging to define temporal searches for event sequences. Generally, apriori search is not flexible enough in this situation where the user wishes to search with extra degrees of freedom in relation to time. An exploratory approach, such as, pattern discovery is often more suited where we see an abundance of studies.

Conversely, classification and clustering, two of the most commonly undertaken tasks in the knowledge discovery process to extract higher-level abstractions of data have encountered little attention. These methods have received long standing in the data mining community, but are generally challenging in the hands of end-users, perceived as black-box and difficult for non-experts to apply effectively. We fail to find any prediction approaches applied to event data which is somewhat surprising given the utility for predicting events. Finally, segmentation has been studied for univariate time-series data but in the context of classification to identify effective input parameters for labelling human activity [RLS⁺14, RLK⁺15]. Segmentation is a domain specific task, which is widely accepted as an unsolved problem in the data mining community [KCHP04]. We see many approaches in this area in the visual analytics field where the input of domain knowledge is essential to define what constitutes a change boundary in the data.

Temporal data is growing in size and complexity which provides an ever-increasing need for advanced knowledge discovery solutions. The current trend has been toward systems which either improve the speed and accuracy of existing methods, or make complicated procedures accessible and understandable in the hands of end users by applying and abstracting away from the underlying algorithms with interactive visual interfaces. However, further research is necessary in this area to fully support the knowledge discovery tasks. We summarise the areas where we believe future work is required.

- **Classification:** Current systems introduced are disjoint with classifier training, feature selection, and result verification considered as separate tools. In addition, there is no capability to validate and correct results when the accuracy is low or they do not meet the users analysis needs. Support for all the tasks required during classification need to be considered and unified into one framework.
- **Clustering:** Knowing which algorithm and features of the data to use for clustering is still an unsolved problem in this area. Further work is required to introduce a system which empowers non-expert users in selecting a suitable algorithm and parameters guided by visual interfaces.

- **Indexing and retrieval:** The retrieval process is well studied; however, we fail to find any techniques which utilise indexing which is essential for maintaining a real-time application as larger data sets are applied. Further work in this area is needed to introduce visual analytic processes which utilise indexing for retrieval to future-proof the techniques.
- **Pattern discovery:** Time-series pattern discovery systems rely on parameters being specified prior to execution. Parameterless search has been introduced but is computationally inefficient for larger data [SKU⁺12]. Further work is required to introduce parameterless search, guided by visual interfaces along with increasing the efficiency of the underlying algorithms to enable real-time visual feedback for larger data sets. A wealth of domain specific solutions are also identified, it would be of benefit to the wider research community for these studies to be applied to other domain areas.
- **Prediction:** A multitude of models in the machine learning field exist for predicting different facets of temporal data. We fail to find any literature for assisting users in model selection amongst these techniques generally, although, we did find literature for selecting an appropriate ARIMA representation [BAF⁺13]. The prediction of temporal event data is another area requiring further work.
- **Segmentation:** Further work is required to realise the conceptual framework presented by Gschwandtner et al. [GSB⁺15] which aims to unify the model, methods, and parameter selection process for segmentation into one framework.

While the current focus in the visual analytic community has been on applying visual interfaces to existing algorithms from the data mining community, we perceive an interesting direction of research for the two fields will be building user-driven algorithms specifically geared for a visual analytics approach to solve difficult problems previously inconceivable independently.

Generally, data mining is often perceived as 'black-box' in the hands of end users when they do not understand how the algorithms operate and therefore do not know how to use them properly which leads to incorrect or unsatisfactory results and therefore we see the black box phrase being used to communicate their frustration. Within the literature, we see a wealth of techniques for assisting users in selecting an appropriate algorithm, model and associated parameters. We also see literature to assist users in validating results. While this is compelling, the future direction for the visual analytics field needs to be to abstract away from the underlying data mining algorithms and associated parameter definitions from an explicit defined process, to an implicit one which occurs in the background. We see a handful of techniques which introduce feedback loops [BAF⁺13, WLY⁺14, KP99, XCH⁺14] to allow the user to provide an indication to the computer of the results they expect to derive and the algorithm does the remainder of the processing work to achieve their goal without having to understand how the algorithm operates or being exposed to any parameters, instead the visualisation of the users needs defines the operation undertaken. When the results are incorrect, the user iteratively provides more feedback to the algorithm. This type of approach clearly seems the way forward for unifying data mining and visual approaches to empower non-expert users.

Part II

Research Development

3. A STUDY ON EFFECTIVE CHART VISUALISATION AND INTERACTION

“What about confusing clutter? Information overload? Doesn’t data have to be boiled down and simplified? These common questions miss the point, for the quantity of detail is an issue completely separate from the difficulty of reading. Clutter and confusion are failures of design, not attributes of information”
– Edward R. Tufte, *Envisioning Information*

Contents

3.1	Introduction	51
3.2	Background	52
3.3	Tasks and Design	53
3.3.1	Task-Based Evaluation	53
3.3.2	System Development	55
3.4	TimeNotes	57
3.4.1	Layout	59
3.4.2	Node Interaction and Rendering	60
3.4.3	Overlay	60
3.4.4	Annotation	61
3.5	User Study	61
3.5.1	Tasks and Stimula Design	62
3.5.2	Study Hypothesis	63
3.5.3	Study Analysis	64
3.5.4	Experimental Setting	64
3.5.5	Analysis Results	65
3.5.6	User Study Discussion	67
3.6	Field Study Discussion	67
3.7	Summary	68

IN the previous chapter, we presented an overview of state of the art visual-oriented methods for temporal data. One of the key underlying challenges highlighted is that screen resolution is small in comparison to the data storage capacity faced in the current age. When more data items are rendered than the available pixels, an over-plotting problem occurs where more than one data item is assigned to each pixel, which leads to a loss of information. Multi-scale representations [JE10, PCS95], histogram [KL06, HDKS07, BPS⁺07], and lens based interaction techniques [Kin10, ZCPB11] have been introduced to enhance the exploration of large time-series data (Section 2.4).

Data mining is one solution but it is only advised when the question the user seeks an answer for is known prior, alternatively, visualisation is applicable when this does not hold true (i.e. it is necessary for the user to explore the data) [Sta14]. Exploration is the task of understanding the data to build hypotheses. Analysis often involves identifying segments of the time-series where phenomena occur and comparing between time segments for interesting patterns, which can be used to form, prove or refute the hypotheses. After analysis the findings are communicated to a wider audience.

During the exploration of smart sensor data, movement ecologists manually explore time-series graphs of several attributes to gain an understanding of the mapping from signal to behaviour [SH08]. Often this is a combination of applying domain knowledge, inspecting environmental attributes, and deriving additional attributes (e.g. posture and

This chapter is based on the publication ‘TimeNotes: A Study on Effective Chart Visualisation and Interaction Techniques for Time-Series Data.’ [WBJ16].

energy use). After a preliminary investigation, the data is analysed by labelling the behaviours of interest which have occurred throughout the duration of the deployment. This is a cognitively demanding and time-consuming process when considering data recorded at a high-frequency over long periods of time. The results of the analysis are presented to the wider research community. This typically involves extracting the signals of interest and annotating them, before inclusion in publications or presentations.

In this chapter, we evaluate current visualisations and extensions to these existing approaches. Based on our evaluation we propose TimeNotes, a visualisation technique utilising built-for-purpose interaction techniques that supports exploration, analysis, interaction, and presentation. We evaluate the effectiveness of TimeNotes through an empirical study and a field study which highlights the application of our approach applied to time-series data. While, effective methods have been introduced to address time-series mining tasks [BAP⁺05, LKL05, HF09, HMJ⁺12]. In this chapter we focus, on effective exploration, analysis, and communication for when data mining is not applicable [Sta14]. TimeNotes forms one part of a larger piece of software which is the subject of the following chapters, specifically, we expand TimeNotes to include analytical capabilities and advanced visualisation which assist biologists in their work-flow.

Our work consists of the following contributions:

- We develop TimeNotes, an approach for chart visualisation and interaction with time-series data utilising a node-link layout to visualise multi-dimensional time-series data. Our approach is purpose built for communicating and navigating through large time-series data.
- We undertake a user study comparing TimeNotes to the current state-of-the-art method, stack zoom [JE13, JE10], which has been previously evaluated against traditional techniques in the literature.

The rest of this chapter is organised as follows. In section 2, we discuss the project background and discuss the biologists existing work-flow before presenting two early prototypes. In section 3, we introduce tasks and requirements when operating on time-series data. In section 4, we introduce TimeNotes. In section 5, we outline the empirical study and results obtained. In section 6, we detail the field study and results. Finally, in section 7, we summarise our findings.

3.2 Background

The result of this chapter, TimeNotes, was produced through an iterative and collaborative approach applying agile methodologies to respond to the unpredictable and experimental nature of the project undertaken with our domain experts, movement ecologists. In this chapter, we focus on interactive data visualisation to assist in the exploration, analysis, and presentation of the large time-series data to help biologists understand their smart sensor data.

In chapter 1, we detail the main challenges faced by ecologists while undertaking their research. In this chapter, we target exploration to assist the user in understanding the mapping from signal to behaviour, interaction approaches for analysis to aide users in identifying animal behaviour and presentation intents for communicating findings to the wider research community. These correspond to the domain level challenges 2, 3, and 4, identified in chapter 1 section 3, which we now summarise.

Exploration is currently a cognitively demanding process, which heavily relies on the domain expertise of the biologist undertaking the study. The biologist acquires knowledge as they inspect the data series, tying together sequences and annotating their knowledge discovery journey. Currently, exploration is an inefficient process, as existing tools available do not support the biologists in undertaking these tasks, instead the ecologists note labels and sequences of interest in an associated spreadsheet for later reference. During analysis, the users categorise the data by labelling behaviour instances. This is a time-consuming process, often the user has to reverse engineer how they determined the earlier found mappings to proceed. The user will have an understanding of the data from the exploration, and may know where the events of interest to the study occur, therefore a percentage of the signal can be considered as non-interesting (predominantly through inactivity). The earlier identified behaviour instances are used as reference to determine the behaviour groupings

in the data set during analysis. Neither of these analysis tasks are supported in the current infrastructure without interpretive effort. Finally, after the user has explored and analysed the data, they will seek to present the findings to their colleagues, research group, or to a larger research community through academic journals and conferences. To facilitate this the user creates a corresponding document with screen captures and associated time-stamps for later reference. It is typical that co-authors or co-investigators will validate the results and pan through the series to identify these in the data. At present, there is no way to directly access these data regions without manual effort.

3.3 Tasks and Design

In this section, we investigate task based taxonomies for temporal data to understand the typical tasks users wish to undertake with their data during exploration and analysis to guide the production of the software and visualisations to ensure they meet user-demand.

We adopt a user-centred design methodology [AMKP04], a framework for multi-stage problem solving revolving around the user of the system. Initially an understanding of the users wants and needs from the system are obtained (our task-based evaluation coupled with the domain level background we have obtained through our observations). Given, the initial requirements, prototypes are produced which are used to test the validity of the requirements in practice. Observing the users interacting with the system and gathering their feedback is then used to drive forward the development. This is undertaken in an iterative manor until the requirements needs are meet. To achieve this the biologists are involved in the development cycle from starting point. Prototypes are developed for the biologists with feedback gathered for each iteration, along with observing the biologists performing their tasks to aide our understanding of the problems and work-flows. We apply this methodology as the diverse nature of the project and data mean it is difficult to interpret the users upfront needs, therefore an agile methodology is suitable.

We implement the current state-of-the-art and traditional techniques in the literature for time-series visualisation (see section 2.4 for a graphical overview). We incorporate these into a software tool the biologists are already familiar with. The biologists utilise their data to perform their processes on each visualisation technique. We observe the way they use the tools and gather associated feedback. It quickly became apparent that the support for their tasks using the existing methods was not enough to fulfil their needs. In short, the current approaches feature scalability issues when applied with large sensor data. As every visualisation is constrained to the display space which impacts their suitability for large data (something we also discuss in section 2.4). We do not formally present the feedback we received since from a scientific perspective it is of little value to do so for each iteration, we do however note the lessons we learnt throughout and present a formal validation of our final approach via a user study.

3.3.1 Task-Based Evaluation

While we have presented a graphical overview of the current state of the art techniques for time series visualisation (Section 2.4), Javed et al. [JE13, JE10] provide a comparison of stack zooming against standard techniques for the navigation of temporal data. They perform both design based and controlled user studies to assess advantages and disadvantages of multi-focus techniques versus overview+detail techniques. Findings include applicability of the stack zooming approach to several scenarios and report an increase in performance when compared to current standard techniques.

Effective visualisation assists the user in accomplishing the tasks they wish to undertake with their data. Ward et al. [WGK10] identify *exploration*, *analysis* and *presentation* as three abstract tasks the user seeks to accomplish with visualisation. In section 1.2.1 we map these user intents to the ecology work-flows.

Our feedback and observations of biologists using the traditional and state-of-the-art methods from the literature (pixel plot, 2D pixel plot, river plot, SignalLens, smooth SignalLens, RiverLens, ChronoLenses, Stack zoom, overview and detail, and zoom plot) establish difficulty navigating through the data series while keeping context with the underlying data. In this chapter, we seek to understand how the visualisation layout method utilised affects a user's ability to undertake exploration, analysis, and presentation on time-series data. We hypothesise that while stack zoom supports these three user intents, the space-filling tree

Technique	T1	T2	T3	T4	T5	T6	T7
Pixel plot	✓						
2D pixel plot	✓						
River plot	✓						
SignalLens	✓	✓					
Smooth SignalLens	✓	✓					
RiverLens	✓	✓					
ChronoLenses	✓	✓	✓	✓	✓	✓	
Stack zoom	✓	✓	✓	✓	✓	✓	✓
Overview and detail	✓	✓					
Zoom plot	✓	✓					

Table 3.1: A comparison of the existing approaches with the interaction tasks they support in Shneiderman's [Shn96] task taxonomy and presentation tasks from Aigner et al. [AMST11].

map utilised by stack zoom is not the highest performing layout for these tasks. As part of this investigation, in this chapter, we explore new visualisation methodologies and interaction techniques which build on the stack zoom approach. Applying our design methodology, we formulate TimeNotes. We undertake an empirical study and a field study with our collaborators which tests both the static and interactive features of our environment to validate the state of the art techniques and their integration into a novel approach.

In the following section, we propose a classification of the literature not only in terms of visual encoding but in function of tasks plus domain. We give further breakdown of these three abstract tasks. The aim is to understand specific user intents but also in support for these in the current techniques in the literature.

Exploration and Analysis

Shneiderman [Shn96] presents a task by data type taxonomy which lists several low-level tasks required to perform analysis and exploration in large data collections:

- T1. *Overview* - Gain an overview of the entire collection of data.
- T2. *Zoom* - Zoom in on data items of interest.
- T3. *Filter* - Filter out uninteresting items.
- T4. *Details-on-demand* - Select an item or group and get details when needed.
- T5. *Relate* - View relationships among items.
- T6. *History* - Keep a history of actions to support undo, replay such that the user can retrace their steps to show how they obtained their findings.
- T7. *Extract* - Allow extraction of sub-collections and of the query parameters, so that once users have obtained what they desired, the set of parameters that facilitated their findings can be sent to others which illustrates the steps they undertook.

When specifically dealing with time-series data, one can also consider the task models by Andrienko and Andrienko [AA05] and McEachren [Mac95] as presented and described in Miksch et al. [AMST11]. The *elementary* tasks (Andrienko and Andrienko [AA05]) consist of data element look-up, comparison and relation seeking. *Synoptic* tasks involve patterns and trends in the data and relationships within data or to external data. It can be seen that the time-series data tasks complement the Shneiderman task taxonomy. The former concentrate on the ability to analyse, whereas the latter suggests the provision of the mechanisms and environment for that analysis to take place.

We compare the existing approaches to the tasks supported in Shneiderman's [Shn96] taxonomy for exploration and analysis tasks. Table 3.1 shows the results. Frequency based approaches only provide an overview of the entire data series. Lens based, overview and detail, and zoom plot techniques expose interaction to support zooming in areas of interest from an overview of the time-series. ChronoLenses supports a wider array of interaction tasks applied to each lens which allows an overview, zooming, details on demand, and relationships to be perceived. Stack zoom fully supports all the tasks through using a hierarchical layout which maintains a graphical history of user interactions.

Presentation

Aigner et al. [AMST11] introduce three requirements for incorporating detected event instances into a visual representation which communicates to the user relevant information, namely communication, emphasis, and conveyance. Such that, the visual *communicates*

the fact that something interesting has been found, *emphasize* interesting data among the rest of the data, and *convey* what makes the data interesting.

None of the existing methods were built with presentation intents. In stack zoom, a disjoint nested tree view is automatically built during exploration which contains the hierarchical layout and acts as a management interface for constructing a presentation view. Each node is labelled with the stack coordinates, from which the user selects nodes and sub-trees to display. However, by using a separate view the tree loses the context with the underlying data and stack zoom layout to be fully regarded as a presentational tool.

Conclusion

We perform a task-based evaluation for the exploration, analysis, and presentation user intents on visualisation methods for time-series in the literature. We report that none of the existing methods fully support the exploration and analysis tasks by Shneiderman's task based taxonomy [Shn96], with the exception of stack zoom which utilises hierarchical zooming [JE10]. Hierarchical zooming provides an efficient method of navigating through time-series by allowing the user to divide the information space and build a view of only the relevant data at the required granularity which also acts as an implicit graphical history of user actions.

None of the methods were built with presentation intents, although a hierarchical layout provides an implicit history of interaction. As such presentation is not explicitly built into the visualisation. A disjoint presentation tree is utilised in the stack zoom method, away from the core visualisation.

It is clear that a hierarchical layout is powerful for supporting the user intents (fulfilling all of Shneiderman's tasks) and serving as a visual history of how users obtained their findings. As stack zoom is the current state-of-the-art method utilising hierarchical zooming, we comparatively evaluate our method against this and build on the methods presented.

3.3.2 System Development

Following our initial understanding of the problem we developed two prototype solutions which provide an integrated environment for the biologists work-flow. The goal of these prototypes is to help us further understand the problems faced and obtain early domain-level feedback to drive our implementation by exploring methods of representing history (T6) in the visualisation for exploration, analysis and presentation. Firstly, we present an embedded zooming approach, and secondly, a layout based methodology. We allowed the biologists to experiment with the tools and perform some typical data exploration, analysis, and presentation on data they were already familiar with. Based on the feedback we received we refined the visuals and interactions in an iterative manor. Eventually, from these two prototypes TimeNotes was conceived. We detail our early prototypes before identifying the creation of TimeNotes which we discuss in the remainder of this chapter.

Embedded zoom

Zoom and pan methods are the de-facto approach when undertaking time-series analysis for biologists when operating on smart sensor data. While zoom and pan has been effective at enabling the identification of animal behaviour, time is often wasted panning through non-interesting data regions, typically, animal inactivity in the case of smart sensor data. Furthermore, by zooming in to the data, the context is lost with the rest of the neighbouring activity. The context a behaviour occurs in is crucial in understanding and identifying patterns and sequences across time. We have witnessed these challenges by observing biologists using the zoom and pan method for their smart sensor analysis. One proposed solution is a lens based approach, however, current methods do not allow direct control over time-axis distortion, instead opting for pre-defined transformation functions and scales (e.g. Gaussian). We introduce embedded zooming, an interaction technique which gives the user direct control over the mapping of data to display space. Expanding regions of interest, and collapsing regions of little importance allows the user to build a time-series graph utilising optimal screen space of only relevant data. The result is a visualisation which emphasises interesting data to convey only important segments (Figure 3.1).

The embedded zoom method initialises as a traditional time-series graph, consisting of two static zoom handles at position zero and position N in the time-series, where N is the length of the series, inserted respectfully at pixel position zero, and the width of the display such that when initialised a standard time-series graph is formed. The initial zoom handles

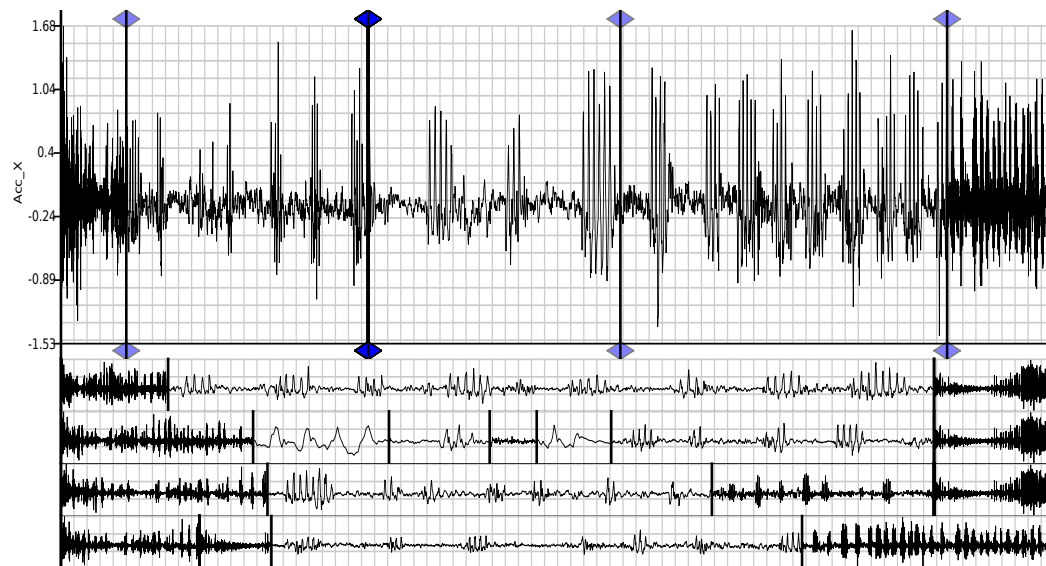


Figure 3.1: The embedded zoom visualisation is constructed using multiple zoom handles (vertical lines) which allow the user to interactively define time-axis distortions to the time-series graph. Snapshots of the visualisation during exploration and analysis are shown below the main display for later access during presentation and reporting.

are static, meaning that they cannot be moved by the user, doing so would reduce the size of the graph which is of little value. Using the right click button on the mouse, the user can select to insert dynamic zoom handles (represented as black lines) at specific locations along the graph which are fixed at the time index they are inserted at. The user can drag and drop the handle to change the distortion applied in real time along the x-axis which condenses / expands the portions either side of the handle to modify the amount of display space allocated between two time adjacent positions. Each handle always represents the time position it is inserted at. The pixel location of the represented time position changes during movement of the handle which creates the resulting distortion in the time-series graph.

The user builds up their representation of the data to show specific patterns and trends which support their exploration, and analysis requirements. Once the user has constructed a relevant visualisation they can snapshot the current visualisation for later access while presenting and reporting. Each snapshot is displayed directly below the main graph ordered by their creation time. Snapshots can be reopened by doubling clicking on an instance which sets the current main graph to the corresponding previously stored state.

We trialled the embedded zoom feature with our collaborators. The general feedback was positive for being able to maintain a graphical history of the display. Interacting with the plot embedded zoom handles was challenging, and typically the user had to add several embedded zoom layers in tandem to spread the temporal duration large enough to view the regions they were interested in, at which point all context to the remaining series was lost. The limitation to the number of simultaneous distortions before an occlusion problem occurs between each zoom handle is challenging when trying to navigate through a single 1-dimensional axis (time).

Stack lens plot

Layout based approaches utilise multiple spatial arrangements of time to allow users to extract insight. Multiple spatial arrangements are often disjoint and therefore do not directly show the context a data region occurs in (e.g. focus and context displays). We introduce stack lens, a layout based interaction technique which allows the user to build nested hierarchies through multiple levels of distorted time-series graphs. Levels of time-series act as a graphical history, while the linkage between them highlights the temporal ordering of the series when distortion is applied. Each level of the graph distorts time by allowing the user to expand regions of interest and collapse them in relation to an earlier generated graph, avoiding embedded zoom levels (e.g. the embedded zoom method). The user expands and collapses regions adding new nested levels on each iteration. Figure 3.2 illustrates the stack

lens method.

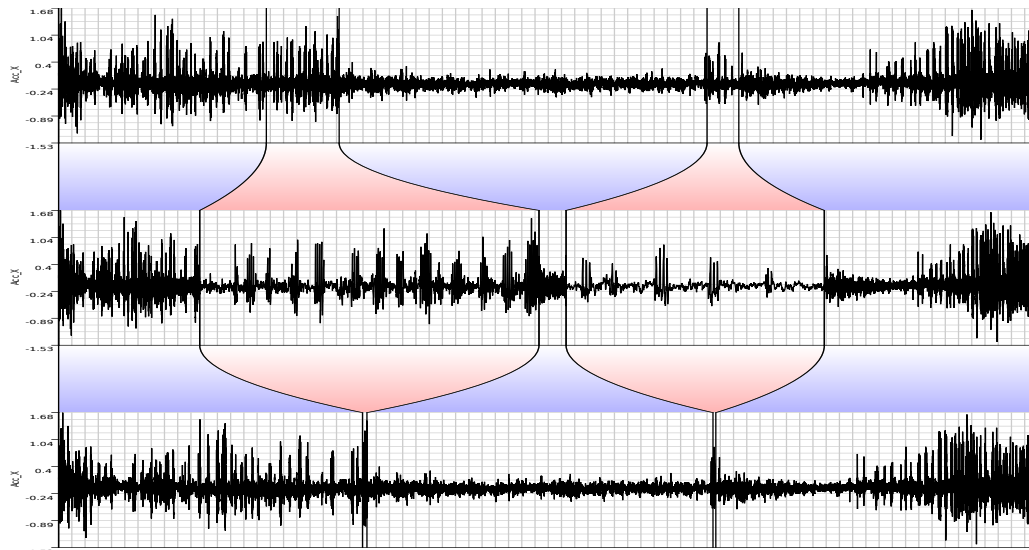


Figure 3.2: The stack lens visualisation is a layout based method comprising of multiple layers of time-series graphs. Zoom handles span between adjacent layers to define the time-axis distortion. The start of each handle defines time index it represents, and the end of the corresponding handle defines the position to map the corresponding temporal position to on the display. Layers create a natural graphical history of interaction with the visualisation.

The stack lens plot initialises as a traditional time-series graph at the root level. Child plots can be added at any time which are displayed directly below the original time-series. The user adds zoom handles between the layers, which allow the user to distort the child graph in terms of the parent graph (i.e. the graph directly above it). Each zoom handle spans two layers, between the current and child graph. The upper portion of a handle defines the time index location on the parent graph to distort, while the bottom portion on the child graph defines the pixel location of the corresponding position to map to in the display. Handles can be copied downwards when a new level is added to the graph. This in effect, allows the user to open and close portions of the time-series graph, such that, aspects of the data are expanded and collapsed until the entire series has been inspected.

Our collaborators preferred specific aspects of the layered zooming approach with respect to the embedded zooming method previously shown. The zoom layers enabled users to trace through the graph to perceive the context an event occurred in with the remainder of the series, which also acted as a graphical history of the explored regions for analysis and later use during the presentation process. However, the display quickly became occluded and difficult to interact with when applied to larger data sets when a large number of layers were required to inspect all the important features present in the data.

3.4 TimeNotes

Our feedback of the initial prototypes indicated that maintaining a graphical history of exploration and analysis had a positive impact on the current work-flow. However, lens based approaches (i.e. embedded zoom) are challenging for users to navigate through the data series without an occlusion problem occurring between lenses. Layout based approaches (i.e. stack lens) provide more promise, offloading the time-axis distortion to layers in the visualisation, however when faced with larger data the number of layers grows, which is difficult to efficiently present to the user without an occlusion problem between the layers. Creating an effective and scalable visualisation which enhances the biologists work-flows is challenging without appropriate support for user tasks.

In this section, we present TimeNotes, inspired by the stack zoom approach and feedback from our initial prototypes. TimeNotes contains additional features and interaction (such as, a flexible node-link layout, overlays, bookmarks, smooth curves to increase usability for reading hierarchies, fused interaction for presentation, and an integrated workspace with import/export of visualisation state) which off-set the problems we discovered, in particular scalability with respect to the size of the data used while supporting the exploration, analysis,

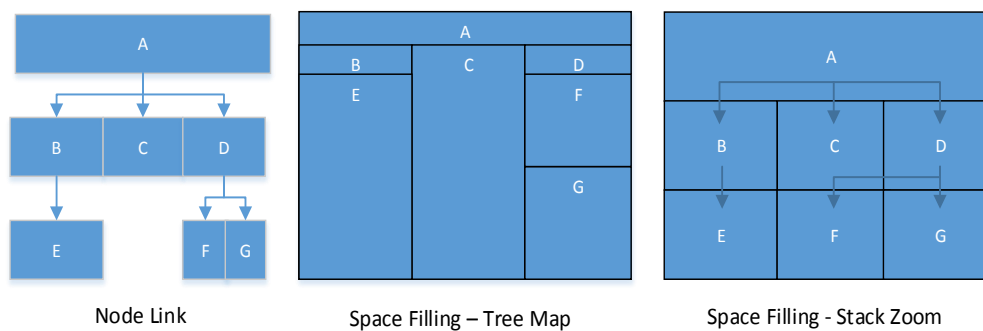


Figure 3.3: Layout techniques commonly applied when visualising hierarchical data. Node link diagram (left), space filling tree map (center), and a space filling layout as used in the Stack Zoom technique (right)

and presentation intents. Chronolenses does not include any hierarchical features which are the focus of this work, but does include overlays and excellent analysis tools, although we offer similar functions and improved overlays. All our new features facilitate exploration, analysis, and presentation of time-series data using hierarchical zooming. We demonstrate the features of TimeNotes that satisfy Shneiderman's task taxonomy and we test these features in the task based user study demonstrating their increased effectiveness.

On initialisation of TimeNotes an overview of the whole data set is drawn on a time-series graph at the root node (T1) (these T numbers refer to Shneiderman's task taxonomy in Section 3.3). Applying rubber band selection across the series creates a new zoom level (referred to as a child node) of the selected data range in further detail (T2). Each node can be repeatedly drilled down such that a hierarchy is generated of the relevant data at the desired level of detail. A flexible node link layout is adopted to represent the generated hierarchy which allows the user to move nodes to a suitable location and size across the two-dimensional viewing plane. Nodes and whole sub-trees can be filtered (T3) by collapsing them to bookmarks. These can be later reopened for further inspection or to communicate the related data. Details on demand can be accessed for each node by viewing statistical summaries of the data contained in each node (e.g. min, max, skew) (T4). Snapping nodes together combines them into an overlaid time-series graph which allows relationships (i.e. frequency and amplitude) between temporal regions to be perceived (T5). The hierarchy created serves as a history of user actions [JE10] (T6) from which the user can construct a visualisation of the relevant data sub-sections for exploring and analysing the data. Data can be exported as raw (sensor) and/or derived (posture, energy use) for report inclusion (SVG export) or in interactive presentation (via our integrated work-bench) (T7). Data can be exported as CSV files. We built TimeNotes with presentation in mind.

Figure 3.4 illustrates TimeNotes applied to data from a Condor. This data set was retrieved from a deployment in Argentina spanning 30 hours at 40Hz (4,320,000) recording accelerometer and magnetometer data channels. The Condor is one of the largest birds of prey in the wild. As a result of this it expends a large amount of energy to fly. The Condor takes advantage of thermals, a natural phenomenon consisting of a column of rising air in the lower altitude of earth's atmosphere. In the TimeNotes display the user has selected several repetitive flapping patterns across the data-series. The data is too dense to identify behaviours at the root level, but it is possible to gain an indication as to the presence of a behaviour by inspecting the high-energy portions of the signal. Zooming shows a detailed view of these allowing the user to differentiate between the signals. TimeNotes depicts the story of the Condor. The Condor flies (flapping it's wings - left tree) to find thermals. We allow any node to depict any data channel. Once the Condor is in the thermal it glides in a circular motion to gain altitude (heading is plotted to show the circular motion - central tree). The Condor then glides through the high altitude to conserve energy before flying to seek another thermal (overlaid flapping behaviour - right tree) to gain altitude. Interesting activity is bookmarked (on the centre right of the data-series) which is minimised for the user to further explore later.

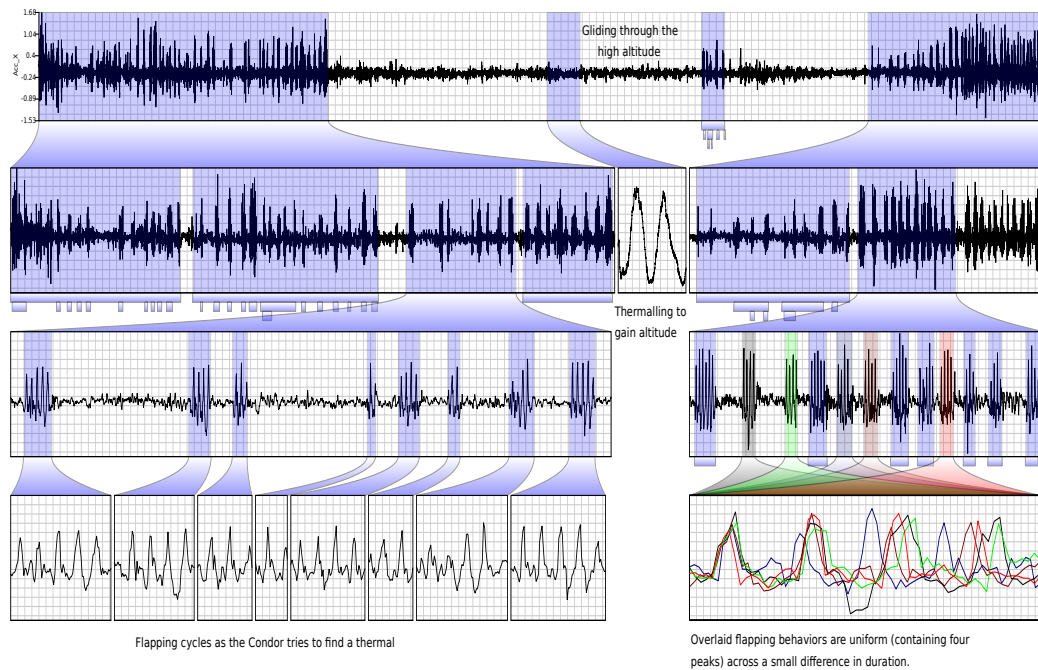


Figure 3.4: TimeNotes displaying data from a Condor, one of the largest birds of prey. The Condor flies (flapping its wings - left tree) to find thermals. The Condor glides in in a circular motion in the thermal to gain altitude (central tree heading is plotted, showing repeated cycles through 360 degrees). The Condor rises in altitude then glides through it to conserve energy. Once the bird loses altitude it flies to the next thermal and repeats (right tree).

3.4.1 Layout

Many methods exist for representing hierarchical data structures effectively [AMA07], we refer the reader to chapter 9 of Munzner [Mun14] for an overview. The most common representational form for a tree is the node-link diagram [WGK10]. This explicitly illustrates the relationship between parent-child nodes at the expense of the display space occupied by the visualisation. Space filling techniques (e.g. tree-maps and stack zoom) attempt to optimally utilise the space they occupy, however they fall short at representing the hierarchical structure of the tree [VWVdW99]. A comparison of these is shown in Figure 3.3. Perceiving the connection between nodes identifies the context in which a signal occurs, which is often the case when undertaking higher-order tasks. Presentation aspects require the visualisation to be of an intuitive nature, where the learning aspect needs to be minimal.

TimeNotes utilises a space filling node-link diagram to represent the hierarchical zoom structure. Each child node defaults to being placed directly below its parent and within its horizontal and temporal extents (although the user may later move and resize). The allocation of display space for each child is proportional to the amount of data represented within that layer. If a new child is added or removed the space occupied by each child is recomputed so that the display space is used optimally.

Space allocation for each child node is computed using the following formulas to calculate the width $w_i = (s \times S_i)$, and 2D coordinates $(x_i = \sum_{j=0}^{i-1} w_j, y_i = d \times (h + l_s))$ of a child node i on the viewing plane, where $s = (w / \sum_{i=0}^N S_i)$ is the horizontal space per data item, w is the width of the parent node, N is the number of children, S is the set containing the number of data items each child represents, d is the current depth in the hierarchy, h is the height of each line plot, and l_s is the spacing between each layer. This can be extended for nodes of varying height by assigning each node a variable height and calculating the cumulative sum during traversal. A layout manager maintains the temporal ordering of the nodes from left to right for optimal chart readability. Ordering nodes also ensures that connections between layers do not overlap that would otherwise occlude the display and make the hierarchy difficult to trace through. A special case occurs where the temporal bounds of children overlap [JE10], for this, we only readjust the position when the entire data bound is not overlapping to minimize node movement.

The connection between each parent-child node is represented as a filled spline to provide a smooth transition through the hierarchy. Opacity is used so if connections overlap it is still possible to trace through the hierarchical structure (Figure 3.5). The user can change

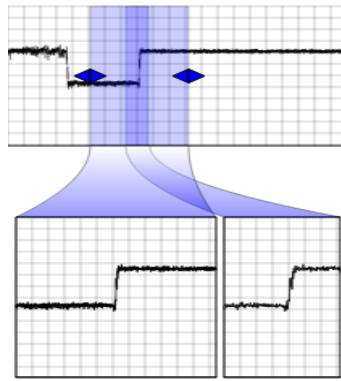


Figure 3.5: This figure shows two zoom regions overlapped. A semi-transparent blue smooth connection is drawn between parent and child nodes to enable temporal positioning to be perceived even when data bounds overlap. Dragging and repositioning the temporal position bounds on the parent node (top) allow the user to modify the data bounds visualised in each child.

the colour of connections and the distance between the parent and child nodes using the associated properties panel.

3.4.2 Node Interaction and Rendering

Each node in the hierarchy is represented as a one-dimensional line-chart. The root node represents the whole data series, while all other nodes in the hierarchy represent a subset (x_1, x_2) of the data, where x_1 and x_2 represent the start and end indices of the data subset where $x_1 < x_2$ holds true. A number of interaction techniques are applied to each node to assist in completing the user intents.

Nodes may be resized in height and width by dragging markers on a node bounding rectangle, allowing the user to accentuate a node of interest. Free movement of nodes on the two-dimensional viewing plane allow the user to reposition a node at any location on the display. The auto-filling layout is disabled for a parent node once one of its direct children is moved, this avoids repositioning nodes the user has purposefully moved and preserves their mental map of the information space. The user can override this by unlocking the child node at any time.

The data bounds each child represents is overlaid in a transparent blue on the parent node. Grabbing the associated region with the cursor and dragging across the parent node pans through the data set (Figure 3.5 top). The child bounds $((x_{c1}, x_{c2}))$ must be constrained to the parent bounds $((x_{p1}, x_{p2}))$, such that, $x_{c1} \geq x_{p1}$ and $x_{c2} \leq x_{p2}$. This prevents the user panning beyond the extents of the data subset represented via the parent. Panning updates the range of data visualised in the child node and all subsequently related children further down the hierarchy by adding the offset moved to each of the child ranges visualised.

Hierarchical layouts are limited to the number of elements that can be visualised. To free up canvas space, nodes and whole sub-trees can be minimised. In doing so they are represented as a rectangular section below the parent node, which we call a bookmark. Hovering over a bookmark shows a graphical preview of the underlying data. Double clicking reopens the sub-tree, using the layout algorithm. Nodes can be deleted from the hierarchy using the delete key or right click menu. If the node is an intermediate node (i.e. it contains children) the user is asked if they would like to move the sub-trees of the deleted node to those of the parent node (Figure 3.6). Similarly, the user can disconnect subordinate nodes at any time and reconnect to any superior node in the hierarchy.

3.4.3 Overlay

Often during analysis, signal characteristics (e.g. frequency and amplitude) need to be compared. This can be difficult when they are positioned far away in the hierarchy or even side-by-side when there is a just noticeable difference. In TimeNotes, snapping nodes together (by dragging and dropping them on top of each other) overlays the nodes together into the same 1D line plot (Figure 3.7) to allow the direct comparison of signals. By default,

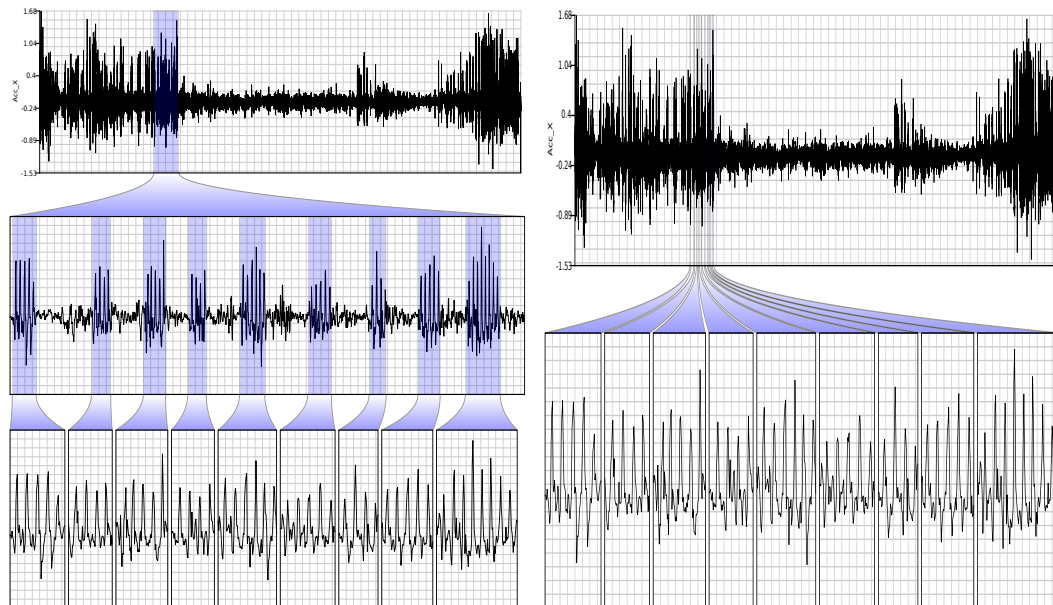


Figure 3.6: Intermediate node removed in the TimeNotes display (left). All child nodes are shifted upwards to the superior node (right).

the plots are stretched such that they occupy the same display width in the visualisation. This is useful when the phenomena varies in speed or time (see dynamic time warping [KR05b]). When this is not the case, we allow the user to align the signals, left, central, or right to maintain temporal duration in relation to each other. Phase can be adjusted by moving the parent panning slider. Connections are mapped to the overlay plot, with a unique colour applied to each which is also applied to each associated line in the overlay so the user can correlate between where each signal originates from and by association its temporal position in the data set. By using the right click menu the plots can be snapped away from each other and restored to their original location on the display.

3.4.4 Annotation

While interacting with the data, users may have comments or insights for themselves or to share with others. Our annotation function allows text to be placed anywhere on the display space. Annotations can further be attached to nodes, such that they move with the node and are hidden when a node is minimised. Text can be resized and coloured according to the users preference.

The data workspace may be saved at any time and reopened at a later date. This allows TimeNotes to be shared between individuals to communicate findings. This also allows the use of TimeNotes to create interactive presentations as an alternative to using current presentation software. This adds an extra dimension of engagement for viewers with the presentation. More importantly, it directly allows access to the original data which means the full context of the data can be shown during a presentation.

3.5 User Study

A user study was carried out to assess the effectiveness of our new visual design with respect to existing similar solutions. A comparison with ChronoLenses [ZCPB11] was considered for testing low-level tasks focusing on pure data analysis, our main focus however was to test the power of using a hierarchical layout with explicit node-link relationships. In this context, a comparison against Chronolenses would have been unfair as much of the hierarchical information is implicit in the visualisation. TimeNotes design was significantly inspired by the approach proposed by Javed et al. [JE10] (which will be referred to as stack zoom for the remainder of the document), moreover Javed et al. [JE10] successfully compared their approach to existing state of the art techniques; we therefore decided to start from their findings, reproduce the stack zooming software according to its description in [JE10], and use it as our worthy antagonist.

To design our study we consulted with researchers in Biological Sciences to identify

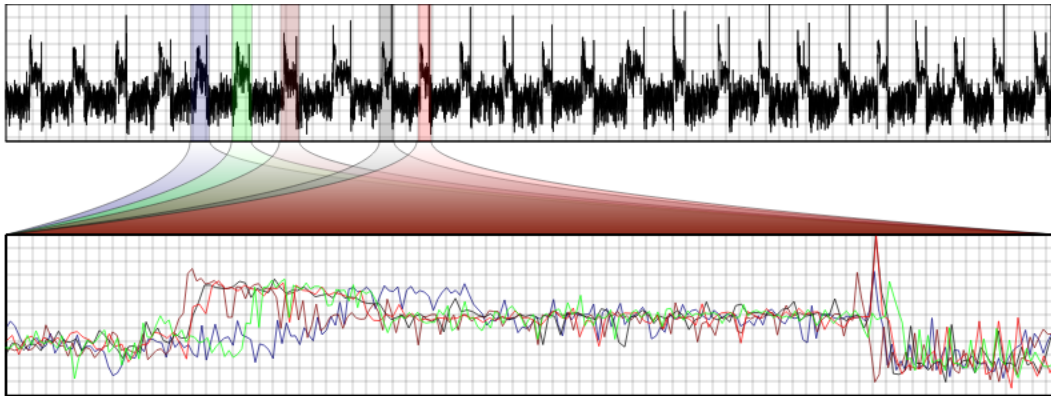


Figure 3.7: In this figure we illustrate the overlay functionality of TimeNotes. Five nodes are snapped together to form an overlaid time-series graph view. A unique colour is applied to each line which corresponds to the connection drawn to the parent node to perceive the temporal location of each line. As we can see, the selected temporal sequences contain a low variance in both frequency and amplitude across time.

suitable tasks that would address questions of potential interest during the analytical process of charted information. A set of four major actions was identified: data traversal and labelling, trend detection and comparison. Further scrutiny allowed to group actions into two main categories: (hierarchy) navigation and comparison. Each action was then broken into its core components, each component refined and translated into a tasks generating a total of five main tasks: Leaf Counting (Task A), Amplitude Comparison (Task B), Frequency Comparison (Task C), Label Analysis (Task D), and Zoom/Pan and Labelling (Task E). To ensure consistency the same notation is used for the remainder of the document.

For the list of tasks presented by McEachern, we test aspects of *identification* (in our tasks B and C), and *localization* (in our tasks D and E). In particular, our task E is high level in that it tests *when* behavioural patterns occur, and also *ordering* and *relationships* of alternate behavioural contexts. Specific examples of localisation tasks (similar to synoptic tasks) are: Temporal Pattern: How often does a (behavioural) pattern occur? (Task E); Sequence: What order do (behavioural) patterns occur? (Task E); We also study Rate of Change (Task C).

3.5.1 Tasks and Stimula Design

Hierarchy Navigation (Leaf Counting) - Task A. The objective of this task is to measure how well the user is able to traverse the hierarchy represented by the stimulus. The stimulus is a simulated hierarchical data interrogation where the user has made multiple selections drilling down to some detailed behaviour in several leaf nodes. The top level depicts all of the data with up to three segments preselected and expanded in the second level of the hierarchy. Further selections are made resulting in internal nodes or leaf nodes. The task is to count the leaf nodes for each segment, entering the answer in the corresponding text box. The time is measured from the presentation of the stimulus until the user clicks on the submit button. There are six different hierarchies presented using the two visualisation styles. Each question is presented twice. These twenty four stimuli are presented in random order with a constraint that the same stimulus must be at least three questions apart. Accuracy is measured as pass / fail on whether the user counted the correct number of leaves for that segment.

Comparison (Amplitude) - Task B. Biologists will compare the amplitude of behaviours across the data set. For example, the strength of a wing beat. This task is designed to measure the effectiveness of our new overlay function. The stimulus is a simulated hierarchical data interrogation where two leaf nodes are brought into close proximity to compare the amplitude of the signal. We compare bookmark charts with and without the overlay function, and stack zoom charts. In this task there are three visualisation types, six hierarchies, and each question is presented twice. These thirty six stimuli have the same constraint on random order as above. Timing is from stimulus presentation until clicking on the submit button. The user must select whether the left or right signal has the highest amplitude. Accuracy is measured as pass / fail.

Comparison (Frequency) - Task C. Feature frequency within the behaviour are com-

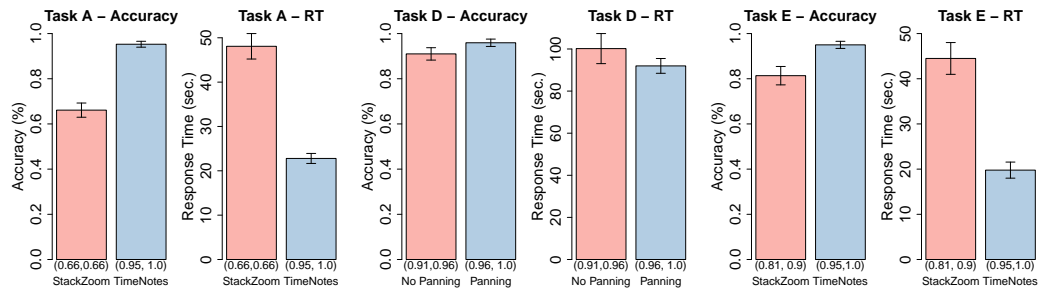


Figure 3.8: Analysis of performance results for Tasks A, D and E, (mean, median) values are indicated below each bar. Error bars show 95% confidence intervals.

pared across the signal (e.g. the speed of a wing beat). This task is designed to measure the effectiveness of our new overlay function. The stimuli follow the same pattern as task B except the user now determines which leaf node has the highest frequency.

Hierarchy Navigation (Zoom/Pan and Labelling) - Task D. We simulate the behavioural labelling process that biologists undertake with this task. To simplify the task we highlight the behaviours in the signal with a grey block. Users are required to indicate whether each one is behaviour A or B (by right clicking in the block and selecting the appropriate label). The block turns to the colour representing that behaviour. Within this task we increase participant degree of freedom as they are allowed to interact with the hierarchy via panning, selection and labelling. The introduction of pan and zooming feature can lead to loss of context [GGW13] we therefore decided to test the effectiveness of this feature within our system. We present twelve stimuli where half enable the pan function.

With the pan function enabled, users are able to grab the segment at a higher level of the tree and move it left and right, scrolling the signal at the lower level of the tree. It provides a mechanism to traverse the time-line allowing inspection of the data at a zoom level the user feels comfortable with. We present the whole time-series data, and allow the user to interact with the time-series in any way they choose, apart from the constraint that only half of the stimuli allow the pan function.

We provide a counter of the number of segments left to label. We ask users to target getting this to zero, but not to spend minutes looking for the last remaining one or two segments. We time from the presentation of the stimuli until the user clicks next. Accuracy is measured according to how many behaviours are correctly labelled.

Hierarchy Navigation (Label Analysis) - Task E. The objective of this task is to measure a more complex use of the hierarchy. Firstly, we test how well the user is able to locate target behaviours in the hierarchy. This involves scanning the hierarchy for a specific pattern. Secondly, we test how well the user is able to relate the found pattern to the overall data time-line which is a critical function for understanding time-series data. Thirdly, we test how well a user is able to perceive temporal ordering of the remaining patterns.

Our chosen task to fulfil these conditions is to present a hierarchy to the user where the data has been segmented and labelled using two contrasting behaviours (A and B). The user must locate the first occurrence of a segment labelled as A. This tests scanning and relation to the time-line. They must also count, and label, how many occurrences of segments labelled B precede it. This tests the temporal ordering implied through the hierarchy. It also requires hierarchy navigation.

We enforce the last condition by setting line transparency to 10% in the second highest level of the hierarchy. At the top level, participants are unable to discern whether labelled segments are A or B because the signal is too dense. At the second level, this may be possible, and therefore the segments could be counted without referring to the hierarchy. By setting lines to 10% transparency, participants cannot use this short-cut and are forced to refer to the presented hierarchy. In real situations data would be dense, and therefore the hierarchy would be used, or it would not matter if a short-cut is found and taken. It is for the purposes of the user study where we want to test the effectiveness of the hierarchies that we must employ this.

3.5.2 Study Hypothesis

In the comparison between TimeNotes and stack zoom we formulated the following hypothesis:

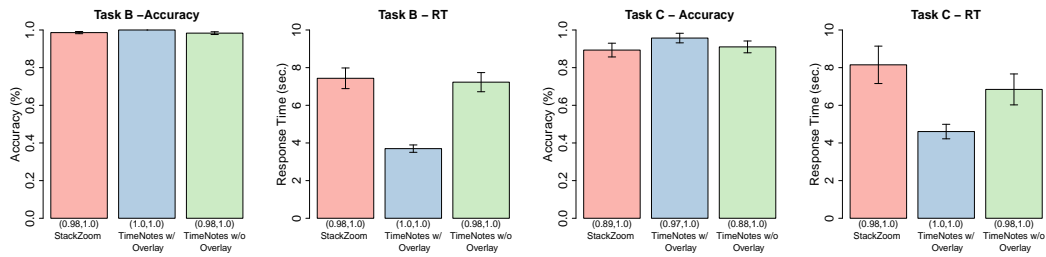


Figure 3.9: Analysis of performance results for Tasks B and C, (mean, median) values are indicated below each bar. Error bars show 95% confidence intervals.

- H1 Task A - TimeNotes will perform faster than stack zoom. We think that being a standard counting operation, with no time limit, both visual designs will perform equally in terms of accuracy; however the increase in clarity with respect to hierarchy linking structure, will help participants to find targets more quickly with TimeNotes.
- H2 Task B and C - TimeNotes with Overlay will perform faster and more accurate than both TimeNotes without Overlay and stack zoom. Since overlay exploits basic Gestalt principles such as grouping effect, the automatic alignment and overlapping of features should ease the mental task of computing differences in amplitude, which is reduced to measuring distance between inner and outer boundaries, and distances between frequency peaks. Without overlay the user needs to mentally perform both tasks of alignment and translation into a unique system of reference. We therefore think that the overlay feature will allow participants to be both faster and more accurate than similar visual designs without such feature. No difference is expected between TimeNotes without Overlay and stack zoom.
- H3 Task D - Visual design with panning option will perform faster than same visual design without panning option. We think that panning (present in both stack zoom and TimeNotes) is a crucial feature when performing analysis of charted data. Panning allows to reduce the hierarchy growth, inevitable when only zooming option is available, and increase space usage.
- H4 Task E - TimeNotes will perform faster than stack zoom. We think that being a standard searching operation, with no time limit, both visual designs will perform equally in terms of accuracy; however the increase in clarity with respect to hierarchy linking structure, will help participants to perform traversal and target search more quickly with TimeNotes.

3.5.3 Study Analysis

A pilot study was carried out involving eight participants including: co-authors, four post-graduate students and one member of our research staff. Together with the collection of preliminary results we aimed at testing length of tasks and study, to avoid confounding effects due to tiredness, randomization of stimula, to ensure that repetitions of the same stimula were not apparent within a task, robustness of the study interface. The five non-author participants were unaware of any of these factors. Pilot study results were positive and revealed trends in the data supporting our initial hypothesis, minor issues with the interface, especially with respect to Task D the only one involving direct interaction, were also noted; all issues were addressed for the main study.

The final study therefore consisted of five tasks, 128 stimula, two visual designs. Supplementary material contains the presentation used for participants training.

3.5.4 Experimental Setting

Participants. A total of 30 participants (2 females, 28 males) took part in this experiment in return for a \$10 book voucher. Participants belonged to both the student and academic communities. Pre-requisites to the experiment were basic knowledge of Calculus such as line charts, phase, frequency, amplitude and familiarity with concepts such as hierarchies and hierarchical organization of data, for this reason recruitment was restricted to the departments of Mathematics, Physics, Computer Science and Engineering, and in the case of students, year 2 and above. We excluded the biologists in the study as we had already worked with them previously in designing the visualisation and user study, we therefore

did not include them to avoid bias. Ages ranged from 20 to 54 (Mean=25.34, SD=8.27). All participants had normal or corrected to normal vision and were not informed about the purpose of the study prior to the beginning of the session.

Apparatus. The visual stimuli and interface were created using custom software written in C++ with OpenGL and QT as the graphics library. Experiments were run using Intel 2.8GHz Quad-Core PCs, 4GB of RAM and Windows 7 Enterprise. The display was 19" LCD at 1440 × 900 resolution and 32bit sRGB colour mode. Each monitor was adjusted to the same brightness and level of contrast. Participants interacted with the software using a standard mouse at a desk in a dimmed experimental room. The absence of windows in the room allowed us to maintain a constant and uniform lighting environment.

Procedure. The experiment began with a brief overview read by the experimenter using a predefined script. Detailed instructions were then given through a self-paced slide presentation (Appendix A). The presentation included a description of the study and also a briefing on how to interpret each visual design and, in the case of Task D, how to interact with both designs for labelling purpose. Participants also received a colour copy of the presentation for reference during the study if desired. The experiment was divided into 5 main blocks with a chance of resting between each block.

All five tasks were completed in sequential order. Given the nature of the experiment each section assessed a different aspect of the analytical process performed by biologist as described in section 3.5.1. Maintaining the same section order for each participant meant that each participant experienced similar experimental conditions. This increased robustness of the analysis of the collected data. Randomness was introduced at trial level. Within a task, trials were randomized to avoid learning effects. A training section preceded each task to familiarize the participant with both task and visual layout.

For Task A, D and E a total of 4 practice trials (two per visual layout) were completed, for Task B and C a total of 6 trials (two per visual layout, with 3 layout options presented in these tasks) were completed. Each training trial included a feedback to the participant regarding the correct answer. Participants were invited to take a short break at the end of each task, if needed. Participants were invited not to take breaks once a task had commenced.

The study was closely monitored, at least two experimenters were always present in the room and participants followed the study requirements. At the end of each task a short multiple choice questionnaire was presented to collect qualitative information from the participant. At the end of the study each participant completed a short post-experiment debriefing interview and questionnaire to collect demographic and further qualitative information. The purpose of questionnaire and debriefing was to obtain comments and recommendations concerning both the experimental procedure, design and usability of both visualisations. Due to the qualitative nature of the feedback, results were used to support the discussion of quantitative results gathered from the testing phase. Both visualisations were at all times presented as valid options, especially during post-processing interview, to maintain unbiased judgement and preserve validity of the collected qualitative feedback.

3.5.5 Analysis Results

In our analysis we mainly considered the effect of task vs. visual encoding. We focused on a comparison of the newly designed TimeNotes performances against the stack zoom approach as this was our primary research question. To perform our analysis we first tested data normality via a Shapiro-Wilk test, more appropriate for small sample sizes. For normally distributed data a repeated measure analysis of variance (ANOVA) was used to test for differences between groups, when data characterized a non-normal distribution, instead of a non-parametric distribution, the Friedman test was used. Both tests were performed assuming a standard significance level $\alpha = 0.05$ to determine statistical significance between conditions. For non-normally distributed data post-hoc analysis was performed via separate Wilcoxon signed rank-tests on related groups for which significance was found. Significance threshold was adjusted using a Bonferroni correction, with corrected significance value of $\alpha = 0.016$ for Task C and D. No cases were found in which both time and error data produced significant results, therefore no correlation analysis, testing the presence of a trade-off effect (e.g. less time leading to more errors) was required.

Hierarchy Navigation (Leaf Counting) - Task A Performance in Task A, summarised in Figure 3.8 as a function of visual design, revealed a noticeable variation between conditions, the Shapiro-Wilk test revealed a normal distribution of performance for both TimeNotes ($p \leq 0.8$) and stack zoom ($p \leq 0.3$). The ANOVA's test showed a significant main effect in response time ($p \leq 0.02$). Accuracy data revealed a non-normal distribution, the Friedman's test showed a significant main effect ($\chi^2 = 25.13, p \ll 0.02$). A closer analysis showed:

- Mean Accuracy: TimeNotes (mean = .95) significantly more accurate than stack zoom (mean = .66) ($p \ll 0.001$);
- Mean Response Time: TimeNotes (mean = 2.27) significantly faster than stack zoom (mean = 4.81) ($p \ll 0.001$);

Comparison (Amplitude) - Task B. Performance in Task B, summarised in Figure 3.9 as a function of visual design revealed a noticeable variation across conditions, Friedman's test showed a significant main effect in both accuracy ($\chi^2 = 5.43, p \leq 0.02$) and response time ($\chi^2 = 42.07, p \ll 0.001$). A closer analysis showed:

- Mean Accuracy: TimeNotes with Overlay (mean = 1.0) significantly more accurate than TimeNotes without Overlay (mean = .98) ($p \ll 0.016$) and stack zoom (mean = .98) ($p \ll 0.015$);
- Mean Response Time: TimeNotes with Overlay (mean = 3.69) significantly faster than TimeNotes without Overlay (mean = 7.23) ($p \leq 0.001$) and stack zoom (mean = 7.43) ($p \ll 0.001$);

No other significant differences were found.

Comparison (Frequency) - Task C. Performance in Task C, summarised in Figure 3.9 as a function of visual design revealed a noticeable variation across conditions, Friedman's test showed a significant main effect in both accuracy ($\chi^2 = 7.68, p \leq 0.02$) and response time ($\chi^2 = 32.07, p \ll 0.001$). A closer analysis showed:

- Mean Accuracy: TimeNotes with Overlay (mean = .97) significantly more accurate than TimeNotes without Overlay (mean = .88) ($p \leq 0.016$) and stack zoom (mean = .89) ($p \ll 0.012$);
- Mean Response Time: TimeNotes with Overlay (mean = 4.46) significantly faster than TimeNotes without Overlay (mean = 7.66) ($p \ll 0.001$) and stack zoom (mean = 8.15) ($p \ll 0.001$);

No other significant differences were found.

Zoom and Pan (Labelling) - Task D Performance in Task D, summarised in Figure 3.8 as a function of visual design with panning option and without, revealed a significant variations across conditions with respect to response time. Friedman's test showed a significant main effect ($\chi^2 = 3.2, p \leq 0.05$) with mean response time for visual design with panning option (mean = 91.9) significantly more accurate than without panning option (mean = 105.68) ($p \ll 0.04$). Further analysis was performed on accuracy with respect to correctly labelled versus wrongly labelled events. Friedman's test showed a main effect ($\chi^2 = 3.00, p = 0.059$). Post hoc analysis with Wilcoxon signed-rank tests revealed mean accuracy for visual design with panning option (mean = 0.96) significantly more accurate than without panning option (mean = 0.91) ($p \leq 0.046$).

Hierarchy Navigation (Label Analysis) - Task E Performance in Task E, summarised in Figure 3.8 as a function of visual design, revealed a noticeable variation between conditions, the Shapiro-Wilk test revealed a normal distribution of performance for both TimeNotes ($p \leq 0.13$) and stack zoom ($p \leq 0.24$). The ANOVA's test showed a significant main effect in response time ($p \ll 0.001$). Accuracy data revealed a non normal distribution, the Friedman's test showed a significant main effect ($\chi^2 = 4.0, p \leq 0.046$). A closer analysis showed:

- Mean Accuracy: TimeNotes (mean = .95) significantly more accurate than stack zoom (mean = .81) ($p \ll 0.001$);
- Mean Response Time: TimeNotes (mean = 19.78) significantly faster than stack zoom (mean = 44.5) ($p \ll 0.007$);

3.5.6 User Study Discussion

All hypotheses stated in section 3.5.2 were confirmed by our study. We also reached significant differences in accuracy across all tasks without any trade-off effect.

Task A's and E's unexpected increase in accuracy results further confirmed the effectiveness of hierarchical visual layout for data navigation and target identification, also noted by Javed et al. [JE10]. Differences in both accuracy and response time of TimeNotes versus stack zoom in Task A and E can be explained by the increase in the TimeNotes design of information grouping. Information grouping has a strong effect on both perception and attention [Tre82], in TimeNotes this is achieved by strengthening the semantic relationship of the node-link structure. The use of colours in stack zoom might have also added an extra layer of visual complexity which would affect the process of information decoding, e.g. colour interpretation for every hierarchy level. Post-experiment interviews also confirmed participants preferences towards TimeNotes visual encoding of the hierarchy ("*Counting leaves in Task A was hard with arrows as it made parents ambiguous.*" (anon.), "*I tried using colours [instead of arrows] and a depth first search approach.*" (anon.) "*While the stack plot is usable, the way it subdivides the data does not feel natural. You have to think for a second about where the links go. With the bookmark plot traversing the hierarchy feels automatic and natural.*" (anon)).

Task B and C reached significance in both accuracy and response time, this leads to the conclusion that overlay is an important feature when considering comparison/estimation tasks. It is fair to note that, especially in Task B, accuracy results are close to optimal. It would be of interest to see how an increase in the sample size would affect the emerging trend. It is also worth noting that in both tasks we only tested pairs, it would be interesting to also test the effect on accuracy when increasing the number of comparisons to more than two signals.

In Task D we tested the effect that increasing the degree of freedom by introducing panning, zooming and labelling would have on user performances in the context of our visualisation interface. Panning is a ubiquitous style of navigation in present-day user interfaces, however in the analysis of large sets of data increasing interactions can lead to loss of context [GGW13]. In Jetter et al. [JLG⁺12] a comparison of panning vs. zooming with panning is performed. Results showed increasing in performance for the former but not for the latter. Task D confirmed the effectiveness of panning, integrated within a hierarchical layout, when dealing with visual search tasks. When sufficient level of detail is reached, while preserving context through the hierarchy linking structure, panning facilitates quick scanning of the zoomed in data (post-experiment feedback: "*Task D. Where panning was not available I had to add larger views and then repeatedly remove them after labelling*" (anon.), "*The zooming ability made it very easy to zoom in to a usable level and then simply slide across the plot, labelling the data as you come to it in an efficient manner. Without panning however was very much a guessing game to find the data, and with continuously adding and removing levels many times I miss clicked and removed a node when I meant to label a segment instead.*" (anon)). Data collected also showed a steep difference between the number of events missed with panning enabled (53 miss) and no panning (110 miss).

Task D and Task E were the most complex tasks as they allowed participants to actively interact with the hierarchy through zoom (e.g. creation of lower hierarchy levels), pan and labelling. Both tasks feature a high accuracy rate with faster response when data segmentation is provided (Task E), e.g. participants need to restrict target search only to highlighted regions. Results in response time of Task D vs. Task E confirms the complexity of visual search tasks even more prominent when handling dense collections of data.

3.6 Field Study Discussion

For the field study, we provided training on the new TimeNotes feature in the already familiar software package. The biologists utilised the software (already installed on their machines) to inspect their own animal data. We observed them using the software and answered any questions which were raised. We held a session after everyone was well acquainted with using the software and held a focus-group like discussion where they discussed features. We summarise their feedback below. We meet with the biologists on a regular basis (every couple of weeks). They provide us with a constant source of feedback which we use to provide innovative solutions to their problems. TimeNotes was inspired by this continual

cycle of changes.

Zooming and panning provides overview and efficiency. At an overview level they need to know whether the deployment was successful. Did the sensor collect the data? Did all data channels collect? Does it look right? Did it work for the duration of the collection period? Does the attachment on the animal shift at all? All these can be answered in a short time by inspecting the sensor trace. Either at a global level for the former questions, or by zooming and panning the data for the latter question.

We demonstrated in the formal user study (task D) that labelling benefits from zooming and panning. Feedback from biologists concurs this. *“Its very cool that you can zoom in without losing context”*. *“We can see from the signal that the collar has shifted at this point in time”*.

Hierarchical layout aids side-by-side comparison and thinking of behaviour at different scales. To support behaviour discovery they use an overview of the data (for time of day and duration of behaviour over larger scale) and zoom down to finer behaviours, for example a wing beat. These are commonly hierarchical. For a sea bird, the top contains many feeding sequences. Each sequence contains behaviour to hyperventilate, dive, swim, prey capture, ascend and rest. Biologists label the overall sequence, then consider each dive sequence, label each component part, and compare across sequences. They view different sensors to see if the behaviour has been captured more fully in a different axis to others, or for example environmental sensors to observe water pressure for dive depth. Our bookmark design provides a clearer link between levels of the hierarchy.

We demonstrated in the formal user study (tasks A and E) that the bookmark hierarchy provides richer context than the stack zoom hierarchy through clearer organization of the hierarchy. They can pan at any level in the hierarchy with lower levels either moving in synchronization (padlocked together) or float on their own. *“This is so good for picking out individual behaviours”*.

Overlay aids comparison. By enabling the *snapping* together of two chart windows, two disparate parts of the signal can be brought into immediate focus for comparison of amplitude or phase. For example, a tag is often deployed on a few animals in quick succession, and utilising this function the same behaviour in different animals can be compared for speed and vigor. We demonstrated in the formal user study (tasks B and C) that the overlay function is more accurate and faster than side-by-side comparison.

Factoring in presentation functionality aids communication. When communicating behaviour, the typical work flow in previous software was to locate the behaviour, export that segment as raw data, read it into Excel, create a line chart, export that to powerpoint and annotate the behaviour. Likewise for inclusion in publications. This loses contextual information, and also typically required a new data (visual) scanning to locate the behaviour. We built in the idea of presentation by allowing the user to reposition bookmarks. Levels can be minimised and maximized. Text can be associated with bookmarks to add annotation. The work area may be saved for future reference. Line charts can be exported as raw data or as SVG. All figures in this chapter are generated using the software, with the benefit that the SVG is vectorized, aiding zooming within publications. This is not tested in the formal user study since it is a feature specific to this software and trivially faster and preferable to the above-mentioned work flow. Feedback is that this will be useful for presentations, teaching and will speed up results inclusion in publications.

3.7 Summary

The analysis of smart sensor data, consists of manually exploring multiple time-series graphs of several data attributes to gain an understanding and map data signals to animal behaviour. The large nature of this data introduces several challenges which currently obstructs effective exploration, analysis, and presentation. In this chapter, we comparatively evaluated existing methods for exploring large time-series data, giving a graphical overview of each and classifying their ability to explore and interact with data. Based on this, we introduced TimeNotes, a hierarchical navigation technique for visualising and interacting with time-series data. We undertake and report an empirical study and a field study with our collaborators. We test both static and interactive features of our environment confirming validity of state of the art techniques and their integration into a novel approach. We also presented RiverLens (Figure 2.12) as a combination of River Plot [BPS⁺07] and SignalLens [Kin10]. We did not comment or evaluate this any further in the chapter because it arose as

part of our discussion and evaluation with biologists, who wished to see this combination in a future version of the software.

In TimeNotes we provide functionality for all of Shneiderman's tasks, along with addressing identification and localization tasks (of McEachren [Mac95, AMST11]) with regard to behavioural patterns. We also provide functionality for presentation intent. TimeNotes allows biologists to explore smart sensor data across multiple time scales while retaining context with the overall data series illustrating the environment a behaviour occurs in. Built for purpose interaction methods allow interesting data regions and patterns found during exploration to be bookmarked for later access. During analysis the bookmarked data regions can be inspected by repositioning bookmarks in tandem with minimising and maximising levels to formulate and prove hypotheses. Finally, when the biologists present and communicate their findings to a wider audience they can construct custom interactive visualisations by annotating and repositioning their work environment. Work areas can be saved for future reference as well as exporting to SVG format for direct inclusion in material. Our findings illustrate that TimeNotes enhances the biologists working environment for the exploration, analysis, and presentation of time-series data, specifically, smart sensor data.

The TimeNotes visualisation is still inherently confined by the available screen resolution. Utilising a hierarchical layout enables the user to decompose the data along both the horizontal and vertical display axes making more efficient use of the screen space available, enabling multiple graphs on the vertical display axis for each zoom level. The user is limited by the number of nodes to be displayed vertically, along with the traditional problem of the number of data items inside each plot to be represented horizontally. In both of these cases an occlusion and overplotting problem may occur. We have tested TimeNotes on data sets consisting of up to 10 million data items in size, which exceeds the current size of the data files in the biological domain. However, as sensor technology evolves, the duration along with recorded frequency and therefore size of the data will increase (100's of millions of items). Further work is essential in this area to insure the scalability of TimeNotes as the technology progresses. Additionally, the tree layout can become complicated when dealing with many simultaneous behaviours or interesting regions. In such a scenario, collapsing the bookmarks and other tree operations are a useful visual aide, but it is not possible when a large number of nodes need to be opened at once for performing cross-comparisons. An interesting area of research will be integrating more expressive techniques, one example is the combination of ChronoLenses and TimeNotes into one framework. TimeNotes has been evaluated against the current state-of-the-art technique, stack zoom, in future work it would be of value to the community to perform studies on different hierarchical techniques, along with a formal user study evaluating TimeNotes for presentation intents.

4. VISUAL ANALYTICS FOR MULTI-DIMENSIONAL TIME-SERIES CLASSIFICATION

"We must use the light of reason to banish the darkness of superstition."

– Brian Cox, *Physicist*

Contents

4.1	Introduction	70
4.2	Background	71
4.3	Precision and Recall	72
4.4	Tasks and Design	73
4.4.1	Problem Statement	73
4.4.2	Domain Requirements	74
4.4.3	Tasks	74
4.4.4	Algorithm	74
4.4.5	Interface Design	78
4.5	Evaluation	83
4.5.1	Case Study 1 - Turtles	83
4.5.2	Case Study 2 - Soaring Birds	84
4.5.3	Formal User-Study	85
4.5.4	Domain Expert User Study	85
4.5.5	Evaluation Against Traditional Data Mining	86
4.6	Discussion	87
4.7	Summary	88

IN the previous chapter, we presented TimeNotes, an effective method for navigating and communicating through large time-series data. While this might seem empowering, the reality of researchers faced with perhaps 10 channels of data recorded at sub-second rates, is that it is time consuming to manually decode behaviour, once the biologist has sufficient knowledge of the data i.e. they have formulated a question (through exploration) and now want to mine the data (analysis) to extract knowledge in a reasonable time-frame. There is currently no effective protocol for the derivation of animal behaviour from smart sensor tags [BCGL⁺14]. In this work, we focus on algorithmic machine learning with visual analytics to assist in accelerating the labelling of animal behaviour. We operate through a separate study and do not build on the TimeNotes platform, although it is incorporated into the same software suite. In future work, we seek to validate the combination of TimeNotes with the work we present in this chapter where we believe it would be of benefit to incorporate our accelerated analytic methods for labelling behaviour into the exploration, analysis, and presentation visualisation and interaction techniques which TimeNotes offers.

The development of animal-attached 'smart' tags have revolutionised biologists understanding of the ecology of wild animals [PTWS14]. Researchers utilising this technology typically manually inspect several time-series graphs for known wave forms corresponding to behaviour. Animal behaviour is derived by applying the biologists' knowledge and experience from previous studies along with synchronised video to search and interpret known wave forms in the data through combining attributes together. Currently this is a manual task, completed by visual inspection, integrating between separate time-series graphs of each data attribute. A typical deployment spans anywhere from two days and up to a week, this can range from 6,912,000, to 19,353,600 measurements. Given such a volume of data, analysis can typically take several hours for the domain expert to undertake the labelling process.

This chapter is based on the publication 'TimeClassifier: a visual analytic system for the classification of multi-dimensional time series data.' [WJL⁺15a].

In this chapter, we present TimeClassifier, a visual analytics system for the semi-automatic classification of animal behaviour, combining the search of large time-series data with the semi-automatic classification of signals using user-defined templates. In this chapter, we target analysis to aide users in efficiently classifying animal behaviour once a mapping from signal to data has been identified. Our system requires one instance of behaviour for the matching process to take place. We utilise visualisation to search for matching behaviour classifications and interaction to facilitate a user in the loop approach for the checking, rejecting, and accepting of results to maintain a high accuracy. Our system provides biologists with a working solution which they have confidence in, and can analyse large complex data sets in a shorter time-frame. TimeNotes allowed researchers to “note” (label) their data. TimeClassifier includes classification in the traditional sense of labelling but classification suggests automation and so extends the previous chapter.

In Chapter 2, we find no work on the interactive classification of time-series data which support all user tasks. More relevant to this work is the information retrieval of time-series data, specifically search-by-example where a number of techniques exist [BAP⁺05, HF09, GS12, RLL⁺05b]. The link between how the signal relates to animal movement is complex, it is therefore not trivial to specify expected variances and shapes in the signal in our domain. We incorporate visualisation, matching, and human interaction into one system for the classification of time-series data and show it is effective through our longitudinal case studies and comparison with machine learning techniques.

Our work consists of the following contributions:

- We introduce a generalisation of the associated user tasks when performing classification of time-series data. The tasks outline the operations a user seeks to undertake when labelling time-series data.
- We develop TimeClassifier, a system that classifies large data with good precision and recall which provides fast, consistent, and accurate classification of biological animal behaviour. We design TimeClassifier using our user tasks for classification.

The rest of this chapter is arranged as so. In section 2, we introduce precision and recall terminology for evaluating machine learning algorithms. In section 3, we present background to the project and how TimeClassifier was created. In section 4, we introduce associated user tasks for classification, our visual analytic system and associated algorithms. In section 5, we report results from our evaluations and case studies. In section 6, we present a discussion. Finally, in section 7, we summarise our findings and outline future work.

4.2 Background

In chapter 3 we observed our colleagues in the SMART group while they explored and analysed their time-series data. During data analysis, a high proportion of the duration is spent labelling behaviours throughout the series, in part this is because it is an entirely manual process consisting of exploring the whole data set linearly, identifying each behaviour in turn by marking the beginning and end time-stamps of the behaviour in a separate file before continuing searching the data. This is a time-consuming and cognitively demanding process. Initially, we asked the biologists if they had considered machine learning to assist them in this process, we were informed they had but the results obtained were of little value. We assumed the lack of belief in machine learning was because these methods typically require a level of expertise and are difficult in the hands of end users because of the black box nature of algorithms.

We created a first iteration of the software which allowed searching the data series for user-defined behaviour instances to obtain feedback on this approach. Figure 4.1 illustrates the core user interface consisting of an overview and detail time-series display. In essence, the user selects subsets of data which represent a specific behaviour, otherwise known as behaviour templates. The user either stores these in the software, or performs a direct search on the data to match the template throughout the dataset, finding all occurrences where a specified behaviour occurs. The searching element is performed using an interactive wizard (later reused in the final software solution, described in section 4.4.5). The results of the search are presented to the user overlaid on the original time-series graph using colours corresponding to the behaviour instance they represent.

While the ability to search behaviour is empowering to the biologists it soon became apparent that this was of little use in reality for research studies. Occasionally a small

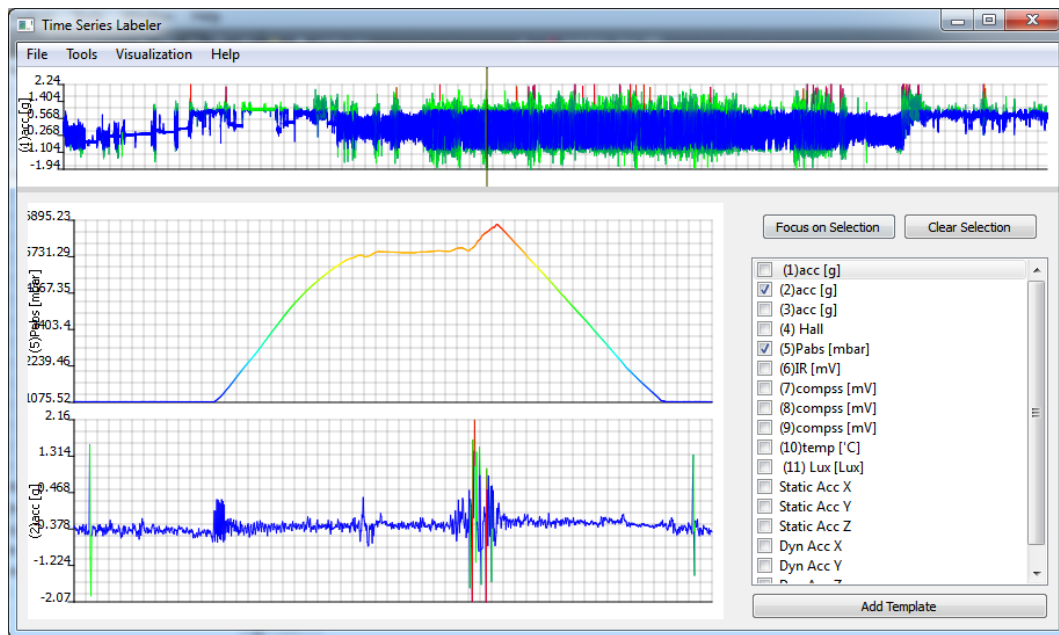


Figure 4.1: This image shows an early prototype of the TimeClassifier user interface. Initially the software comprised of a focus and context display which only allowed users to search for similar patterns across the series. There was no facility to validate classified behaviour.

percentage of behaviour was not correctly identified due to the variability of animal behaviour. At this point it was unclear how to move forward as there was no mechanism to validate and correct the result set obtained. The biologists require 100% accuracy for their studies and therefore abandoned the software. After working extensively with the biologists we saw beyond their criticisms of machine learning. They encounter insignificant results from these systems and therefore have low confidence in them and express frustration with the black box phrase [vLAA⁺13].

In such a situation a feedback loop is vital, allowing the user to inform the software to facilitate the validation of classified results. Thus, the idea for TimeClassifier was born, a readily available visual analytics system for the semi-automatic classification of animal behaviour which incorporates a user in the loop approach to handle user feedback based on classifications obtained. The goal of visual analytics is to marry together human judgement and expertise with computational power to enhance data analysis [TC06]. TimeClassifier realises the goals of visual analytics through the identification of relationships in the data. The user initially defines what constitutes animal behaviour prior to the computer finding similar occurrences guided by user judgement. The remainder of this chapter details the design and implementation of TimeClassifier.

4.3 Precision and Recall

The success of algorithms in the machine learning field are measured using precision and recall and computational time. Precision (P) is the fraction of instances retrieved which are correct. While recall (R) is the fraction of correct instances successfully retrieved [Pow07]. In terms of classification, a high recall means the algorithm classified most of the relevant results. A high precision means the algorithm correctly classified most of the retrieved results. Precision and recall can be combined to one quantitative measure, called an F-Score (F_1), defined as a measure of test accuracy. These are defined as:

$$P = \frac{|C|}{|B|} = \frac{|A \cap B|}{|B|} \quad R = \frac{|C|}{|A|} = \frac{|A \cap B|}{|A|} \quad F_1 = 2 * \frac{P * R}{P + R}$$

where given an input template, A is the set of manually classified behaviour instances (taken from ground truth data manually classified by our domain experts), B is the set of retrieved entities (including those not correct), C is the set of correct classifications among the retrieved entities.

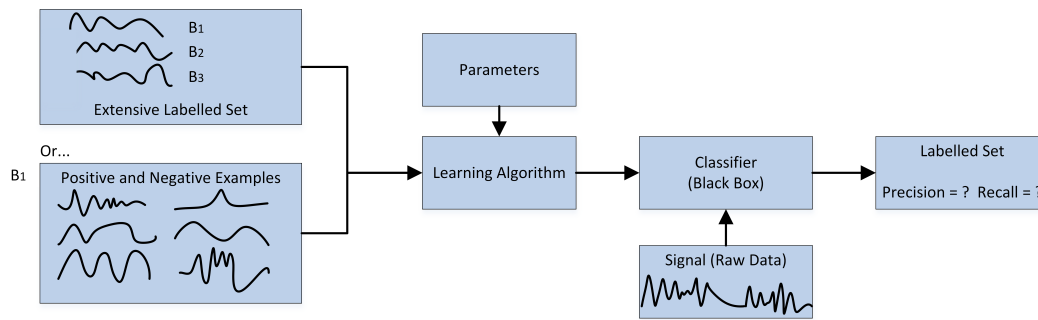


Figure 4.2: This figure shows the classification workflow typically utilised when performing supervised learning. A model (classifier) learns from extensively labelled data to generalise relationships. The model operates on new unseen data to classify behaviour.

Existing machine learning approaches do not have sufficiently high precision and recall for biologists to have confidence in using them. After working extensively with biologists we have seen beyond standard criticisms of machine learning (i.e. ‘black box’). The low precision and recall from these systems and therefore low confidence is expressed in frustration using the black box phrase [vLAA⁺13].

When faced with low precision and recall, the available algorithms do not have an accessible means by which to rectify any mistakes beyond supplying an unknown quantity of further training examples, choosing alternative algorithms or changing parameters. The latter two are only available options for machine learning experts, and the former requires a huge investment in expert time to label data with no promise that precision and recall problems will be solved. We provide a visual analytics system that acts as an interface to all of the above problems and utilises a single positive example to start the process (i.e. low expert time). In our evaluation, we outperform machine learning techniques. Biologists can use our system to build up extensive labelled data that could ultimately be used to improve machine learning algorithms in this area once effective tagging protocols are introduced.

4.4 Tasks and Design

In this section, we present the domain level tasks and design of our system, refined through an iterative development with our collaborators at the SLAM research centre.

4.4.1 Problem Statement

Matching signals in time-series data is well established [KK03]. Classically this is approached in the data mining community by having extensive labelled data demonstrating positive and negative instances of the template signal. Classifiers are built from that corpus of data, and then operate on unlabelled data to provide the classifications. Techniques such as random forests [Bre01], dynamic time warping [KR05b] and support vector machines [CV95] are all used. The success of machine learning algorithms are measured using precision and recall and computational time [TK07] (detailed in section 4.3). These algorithms have been considered, but have been difficult to introduce because of the large numbers of training sets required and their low discriminating precision in practice.

Figure 4.2 shows the classification work-flow typically applied when utilising supervised learning. Classification is approached in the data mining community by having extensively labelled data demonstrating positive and negative instances of the template signal. Obtaining this data is time consuming, requires domain expertise, and the undertaking of field studies to gather video synchronised data. Often, it is not possible to obtain large quantities of such data due to logistical and environmental constraints. Secondly, choosing the classification algorithm and parameters introduces its own class of problems. Typically, in this process, the data dimensionality is reduced to a few parameters which contain the relevant information to perform classification; feature extraction. Good classification results rely heavily on the features chosen, however, extracting a desirable feature set is considered more of an art than a science and takes a great amount of skill along with trial and error [Smi97]. Once the data is classified, if the precision and recall are less than desired, decisions must be made

as to whether it is useful to invest more time creating additional training input, modify the parameters, or use a different learning algorithm. It is not obvious what the next best step to take is without expert knowledge of the underlying algorithms. Our system exploits and incorporates the knowledge of the domain expert to guide the classification process.

4.4.2 Domain Requirements

To be most useful to the movement ecology community, software to help in the analysis of smart tag-acquired data needs to be able to deal with large quantities of data to identify and classify behaviours quickly to a high accuracy. Validating results is essential to be able to see what was classified, along with the ability to manually accept and reject results via applied domain knowledge. Extensively large collections of labelled data do not exist in the marine wildlife domain in-part this is because smart sensors are an emerging technology, as such there is currently no precise protocols for deployment, which is an active area of research in the biology domain. The result of this is the positioning of sensors differs between deployments as it is challenging to locate precisely the same location, along with the variability in animal behaviour which can lead to sensors moving (i.e. scratching). Therefore, any tool must be able to operate effectively on a low number of provided instances of a behaviour.

4.4.3 Tasks

From our observations with machine learning algorithms we decompose classification into six primitive tasks. The combination of these allow the user to perform effective classification. We give a generalisation of the tasks associated with classification and apply them to the context of time-series data.

Identify: Find data subsets which correspond to the specified behaviours in the time-series. *Identify(t)* returns all occurrences of a template (t) throughout the data series. The result is a set of unlabelled subsets from the time-series which are of the same classification as the input sequence t .

Associate: Classify behaviour to a specified group. *Associate(us, s)* associates an unlabelled subset of data (us) to a specified class (s). This is used to manually classify data, and for accepting found behaviours from an *Identify* to the correct group.

Reject: Remove a previous result or classified instance from the system. *Reject(ls)* rejects a labelled subset (ls) from the current group it is classified in. *Reject(us)* removes an unlabelled subset (us) from the result set of found behaviours (from an *Identify*).

Move: Relocates a classified instance to a specified classification group. *Move(ls, s)* moves a labelled subset (ls) to the classification group (s). This is equal to applying a *reject(ls)* and an *associate(ls, s)* operation.

Membership: Inspects which classification group a subset belongs to. *Membership(ls)* returns the classification group s_i the labelled subset (ls) belongs to.

Compare: Compare is further decomposed into two facets. Firstly, comparing signals between classified instances. *Compare(a, b)* enables two data subsets a and b to be compared for similarity. Secondly, comparing a signal to the rest of the set of classified instances. *Compare(a, s_i)* allows the comparison of a subset a to the list of classified subsets contained in the specified classification group s_i .

We build TimeClassifier from these user intents to facilitate an effective software protocol for classification of time-series data.

4.4.4 Algorithm

We consider classification as an extended form of search in the time domain. For example, given a set of labelled data, classification operates by searching for matches for each labelled item and assigns the results accordingly to the corresponding classification group. A user oriented approach involving search to keep the user in-the-loop exposes control to the user for specifying the results they require for each classification [BAP⁺05].

Signal matching is a process for determining the presence of a known waveform in a larger dataset. In essence this works by sliding the specified template across the data set, computing the similarity of the template at each position in the data series corresponding to how similar the sample was at each position. This allows the user to select a single positive example of a behaviour and classify all occurrences of it in the data.

A distance measure is used to determine a quantitative value corresponding to similarity or dissimilarity between time-series (section 2.3). The choice of distance measure is important when considering the ability to deal with outliers, amplitude differences, and time axis distortion [KR05b]. The most common form of distance measure is Euclidean distance, although it is heavily subjective to noise, and is not capable of depicting objects stretched or compressed along the x or y axis. Dynamic Time Warping [KR05b] (DTW), and Longest Common Subsequence [AAMHK12] (LCS), have been introduced to overcome these issues, however, these come as a trade-off with execution time (both quadratic complexity $O(N^2)$). Recent optimisations to DTW have been applied to finding a singular most similar sub-sequence [KR05a]. We require the similarity of a sub-sequence at each position in the series, therefore this optimisation is not suitable to support our tasks.

Correlation is the optimal technique for detecting a known waveform in random noise [Smi97]. In signal processing it is well known that correlation has a linear complexity frequency space implementation which other techniques like dynamic time warping cannot offer. Although, cross-correlation cannot handle compression and stretching in time to detect behaviours with varying frequencies (e.g. running vs. jogging) to overcome this the user must re-sample their signal in our system to detect such variances. Ralanamahatana et al. [RG⁺05] state it is necessary to normalise data to extract meaningful matches. However, normalised cross-correlation does not have a simple frequency space representation. We introduce a fast normalised cross-correlation method for our visual analytics system to maintain a fast and highly interactive application.

Matched filtering is the process of utilising correlation as the underlying algorithm for signal matching. Correlation of a specified template is performed over a data set. The results produced indicate the similarity of the template at each position in the dataset. We adopt the matched filtering approach in our system to maintain interactivity with results.

There are a number of variations of the cross-correlation technique for different purposes. We firstly introduce the traditional cross-correlation algorithm for one dimension. Table 4.1 lists the symbol notations utilised throughout the remainder of this section to introduce the methods.

Notation	Description
$c(x)$	Cross-correlation of the data item at position x with the remainder of the data series.
$nc(x)$	Normalised cross-correlation of the data item at position x with the remainder of the data series.
N	Length of the data series.
a	Length of the template signal.
t	Template signal.
f	Data signal.
μ	Mean value of a given input series.
\bar{t}	Mean of the template signal.
\bar{f}	Mean of the data values under the template signal.
σ	Standard deviation of a given input series.
σ_f	Standard deviation of the input signal under the template series.
σ_t	Standard deviation of the template series.

Table 4.1: Notation of the symbols used to detail our fast normalised cross-correlation method.

In time-series space, cross-correlation is the sliding dot product of a template t over a signal f . Results at each position $c(x)$ indicate the similarity of the template t of length a in the signal f at the position x . The formula for cross-correlation follows:

$$c(x) = \sum_{s=-a/2}^{a/2} t(s) * f(x+s) \quad (4.1)$$

Cross-correlation in time-series space is a slow operation, but convolution in time-series space corresponds to point-wise multiplication in frequency space [Smi97]. This can be used to speed up the cross-correlation process. Cross-correlation in frequency space is $O(2*N)$ computational complexity, where N is the length of the data series. This is implemented

by transforming both the template and signal into the frequency domain, these are then multiplied together in frequency space. The inverse Fourier transform of the result forms the filtered answer.

$$nc(x) = \mathcal{F}^{-1}[\mathcal{F}(f')\mathcal{F}(t')] \quad (4.2)$$

The traditional cross-correlation method is sensitive to linear changes in the amplitude of two compared signals [Lew95]. This is not a desired property when considering extracting matches based on their shape, excluding differences in amplitude. We therefore also introduce normalised cross-correlation an approach to excluding linear changes in amplitude, both signals are effectively normalised first before performing cross-correlation. Normalised cross-correlation results are confined to the range -1 and 1. The formula for the normalised cross-correlation follows. In the equation, \bar{f} denotes the mean value of the signal f under the template t , \bar{t} denotes the mean value of the template t , σ_f denotes the standard deviation of the signal f under the template t , and σ_t denotes the standard deviation of template t .

$$nc(x) = \frac{1}{n} \frac{\sum_{s=-a/2}^{a/2} (f(x) - \bar{f})(t(x+s) - \bar{t})}{\sigma_f \sigma_t} \quad (4.3)$$

Fast Normalised Cross-Correlation The normalised cross-correlation method (Eq. 4.3) is a slow operation in time-space. To maintain interactivity in our application it is essential this process is an almost real time operation. The normalised cross-correlation method does not have a direct frequency space expression [Lew95].

We present a fast normalised cross-correlation method in one-dimensional space utilising pre-computed lookup tables to speed up the process. Our technique is loosely based on that presented by Lewis [Lew95]. Utilising our technique reduces the normalised cross-correlation complexity from quadratic $O(N^2)$ time in time-space to linear $O(N)$ in frequency space. Memory usage is traded-off for algorithm speed to make the system usable for its purpose. A total of five look-up tables of the length of the data series (N) are used, and two of the template length (a), with a maximum of four tables required simultaneously. We list the look-up tables required below:

- Data array (Length: data size) - Array containing the input data.
- Template array (Length: template size) - Array containing the template to match in the data series.
- Mean under the template for the data (Length: data size) - Array containing a running mean of the template size for the input data.
- Data minus mean under the template for the data (Length: data size) - Array containing the input data minus the previously computed mean under the data.
- Data minus standard deviation (Length: data size) - Array containing the input data minus the standard deviation of the template length under the data.
- Normalised cross-correlation result (Length: data size) - Array containing the normalised cross-correlation of the input data with the input template.
- Template minus template mean (Length: template size) - Array containing the input template minus the template mean.

The fast normalised cross-correlation approach splits the normalised cross-correlation formula up into two sub-problems, the numerator, and dominator. We first start with the numerator of equation 4.4, defined as:

$$num(x) = \sum_{s=-a/2}^{a/2} (f(x) - \bar{f})(t(x+s) - \bar{t}) \quad (4.4)$$

The mean of the template (\bar{t}) can be precomputed when storing the template. An array of the same size as the dataset is created containing the mean of data values under the template (\bar{f}) for each index in the data set. This is precomputed upon executing the normalised cross-correlation function. Utilising the mean values we can compute a further array containing the signal minus the mean ($(f(x) - \bar{f})$) and another containing the template

minus the mean ($(t(x+s) - \bar{t})$). This reduces the numerator to the cross-correlation formula (Eq. 4.1) which can be computed using fast Fourier transform in frequency space (Eq. 4.2).

Next we move on to efficiently computing the denominator of equation 4.5, defined as:

$$den(x) = \sigma_f \sigma_t \quad (4.5)$$

The standard deviation (σ) of the signal under the template (σ_f) and template (σ_t) needs to be computed. The standard deviation is computed as follows where μ is the mean input value and N is the length of the input array.

$$\sigma^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \mu)^2 \quad (4.6)$$

The earlier computed running mean under the template (\bar{f}), and the mean of the template (\bar{t}) are multiplied together to get the resulting denominator value. The numerator (equation 4.4) and denominator (equation 4.5) are then divided together to get the final result set.

$$nc(x) = \frac{num(x)}{den(x)} \quad (4.7)$$

Listing 4.1 details the pseudo-code for deriving our fast normalised cross-correlation utilising look-up tables. At any time during the run-time a maximum of four look-up tables of length N are utilised.

Listing 4.1: Algorithm for computing fast normalised cross-correlation utilising look-up tables to reduce computational complexity.

```
function * normCorrelation(double[] dataArray, double[] signalArray, int attrID)
{
    // Mean of the signal
    float signalMean = GetSignalMean(signalArray);

    // Standard deviation of the signal
    float signalStandardDeviation = GetStandardDeviation(signalArray, signalMean);

    // Compute running mean vector under template (using cross-correlation in fourier space)
    double * dataMeanVec = GetFastRunningMean(data, attrID, length_of(signal));

    // Data minus mean
    double * dataMinusMean = new double[data_size];
    for (int i = 0; i < data_size; i++) {
        dataMinusMean[i] = data[i][attrID] - dataMeanVec[i];
    }
    delete [] dataMeanVec;

    // Signal minus mean
    double * signalMinusMean = new double[length_of(signal)];
    for (int i = 0; i < signal->GetSignalLength(); i++) {
        signalMinusMean[i] = signal[i] - signalMean;
    }

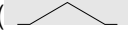
    // Standard deviation of data under template
    double * dataDeviationVec = GetFastRunningDeviation(dataMinusMean, length_of(signal));

    // Convolve in fourier space signalMinusMean
    double * results = new double[data_size]
    results = convolveFast(dataMinusMean, signalMinusMean);

    delete [] dataDeviationVec;
    delete [] signalMinusMean;
    delete [] dataMinusMean;

    // Data divided by standard deviation of data under template multiplied
    // by signal standard deviation
    for (int i = 0; i < data_size; i++) {
        float div = (dataDeviationVec[i] * signalStandardDeviation);
        results[i] = (results[i] / div) / length_of(signalArray);
    }

    return results;
}
```

Extracting Matches The results returned by the normalised cross-correlation algorithm may feature multiple matches within the same part of a matching signal. This occurs because of a drop off in similarity values from a matching signal. This creates what we refer to as a bell shaped curve result for a matching signal. A match will typically return a high value and then slowly drop off with time (). Directly returning all matches above a set threshold, will result in multiple matches for the same signal at neighbouring points in time. We only want each matching signal to be represented once.

To ensure each matching signal is represented only once, we extract the maximal point of a match, referring to the position in the signal the template was most closely matched for a positive match. When a result is over the user-defined threshold, we search the entire length of the template for the most closely matched point along the signal. The highest value in the region is the matching signal and is the only point to be included in results for that match. Further to this, we merge extracted matches together if the distance between them is under a user-defined threshold as proposed by Hao et al. [HMJ⁺12].

Multi-Dimensional Data - The correlation theorems introduced are designed for one-dimensional signals. The data we deal with is of a multi-dimensional nature, and as such the biologists require the ability to define behaviour templates consisting of multiple attributes (e.g. acceleration in three axes). The presence of a behaviour may be more dominant in one dimension, for this purpose we compute the sum of the cross-correlation values for each attribute multiplied by an attribute coefficient. This is defined by the strength of a match ($x_i = \sum_{j=1}^N c_j \times s_{i,j}$) at position i , where N is the number of data attributes, c_j defines the user-defined coefficient for attribute j , and $s_{i,j}$ defines the similarity value for the associated attribute (j) and position (i). By default, each attribute is given an equal weighting ($c = 1/N$), such that, an average of the similarity values for each position is used.

Performance - We have tested the algorithm on data sets up to 10 million rows in size (using a 64bit edition of the software) on a laptop containing an Intel 2.5ghz i7 processor featuring 8GB of RAM. The times taken for data set size are highlighted in Table 4.2 (note: this does not include use of interaction, and is purely based on the algorithmic aspect):

Data size (megabytes)	Number of rows	Time taken (milliseconds)
13.4MB	1 Million	1750
66.5MB	5 Million	26410
122MB	10 Million	84512

Table 4.2: Evaluation of the normalised cross-correlation algorithm in respect to data size versus time taken to evaluate.

4.4.5 Interface Design

The design of TimeClassifier has been guided by our domain characterisation and classification tasks. The result is a visual analytic system which supports the manual labelling process. We extend the manual labelling process using an extended form of search by applying visual analytics to signal matching for the semi-automatic classification of animal behaviour. Visualisation is utilised to assist in finding mis-classifications and interaction techniques to facilitate the checking, rejecting and accepting of results for maintaining a high accuracy. We now detail our system.

The user interface is split into three components (see Figure 4.3 for overview). Firstly, a data view at the top, being composed of the data in a stacked time-series graph format. Coloured segments overlaid on the graph indicate classified animal behaviour (*Membership*). A search panel is located in the bottom left which the user can perform searches on the data utilising the template search wizard (*Identify*). Results are shown in this panel for the user to verify, reject (*Reject*) or accept (*Associate*) results before moving them to the appropriate classification in the bottom right panel, where the classification widget is situated. Classified behaviours are shown to the user in this tabulated panel (*Membership*). Each tab represents a behaviour group with visualisations for the corresponding set of classified behaviours (*Compare*) and buttons to move (*Move*) or reject (*Reject*) classified instances. The categorical colours assigned to each tab correspond to those overlaid transparently on the time-series graph (*Membership*).

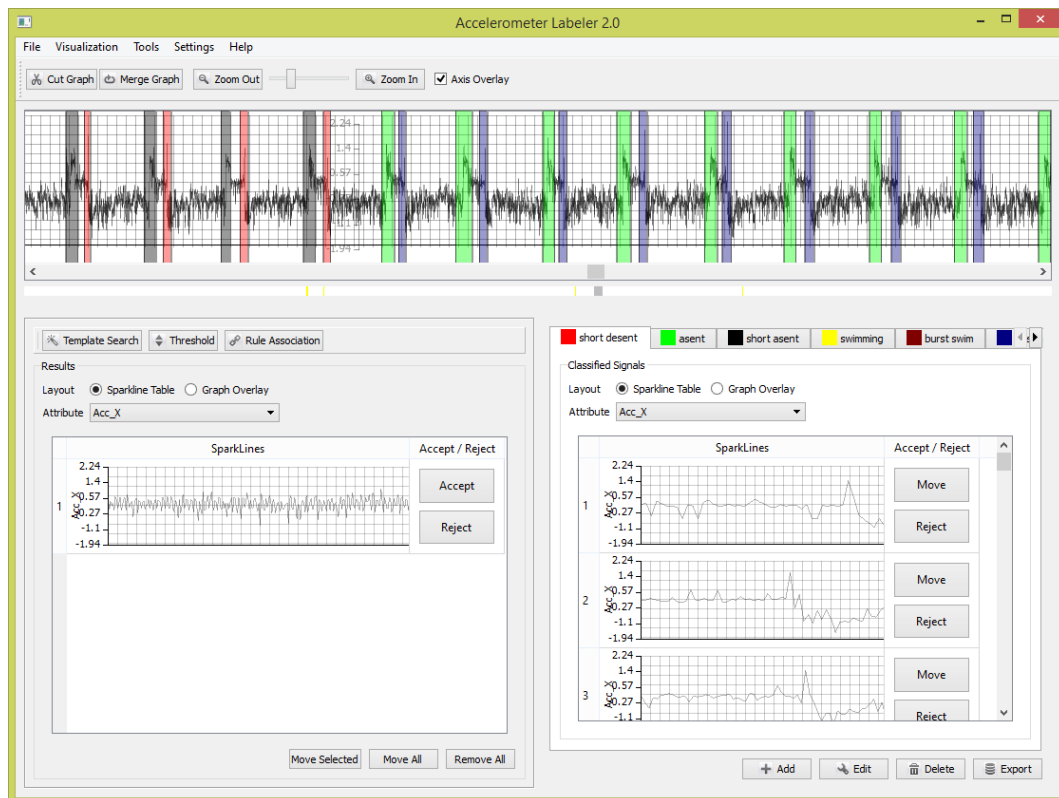


Figure 4.3: This figure shows the main user interface of TimeClassifier operating on data from a deployment on an Imperial Cormorant. Top: A stacked time-series graph of the whole data set with overlaid coloured regions illustrating labelled animal behaviour. The user can apply zooming to see more detail. Bottom-left: The results widget is used for searching for behaviour instances and displaying results. Bottom-right: The classification widget encapsulates classified results. Each tab represents a classification group with each classified behaviour instance represented using embedded time-series and overlaid plot views.

System Work-flow

Figure 4.4 illustrates our work-flow for time-series classification. The user initiates the process by selecting a behaviour instance from the raw data or from the template database. The signal may be re-sampled to capture behaviours at different frequencies. Our classification wizard is used to identify instances of the user specified behaviour template. The results from the classification wizard are exported into the results panel (Figure 4.3 bottom left). Visualisation is used to present the results to the user. Our feedback loop allows the user to improve precision and recall by manipulating the results set to reject, accept, and modify returned instances or by providing secondary examples into the classification wizard. Accepted results are moved to the correct classification tab in the classification panel (Figure 4.3 bottom right), this stores information about the classifications and associated classified data regions. The user may improve precision and recall further by manually labelling data samples by dragging regions from the time-series into the correct classification group. We now detail each of the components of the system work-flow for the classification of animal behaviour.

Behaviour Selection

The first step is for the user to select a behaviour to classify in the data. There are two methods for this in the application. Firstly, query-by-example, and secondly selecting previously saved behaviour instances from the template database.

Query By Example - Query by example allows the user to directly execute the classification wizard from the time-series display by applying rubber band brushing to subsets of data corresponding to a known behaviour. Specific attributes will be dominant for identifying a behaviour, therefore after selection, a dialog is displayed where the user can select which data attributes to utilise for the template. For templates consisting of multiple dimensions, the

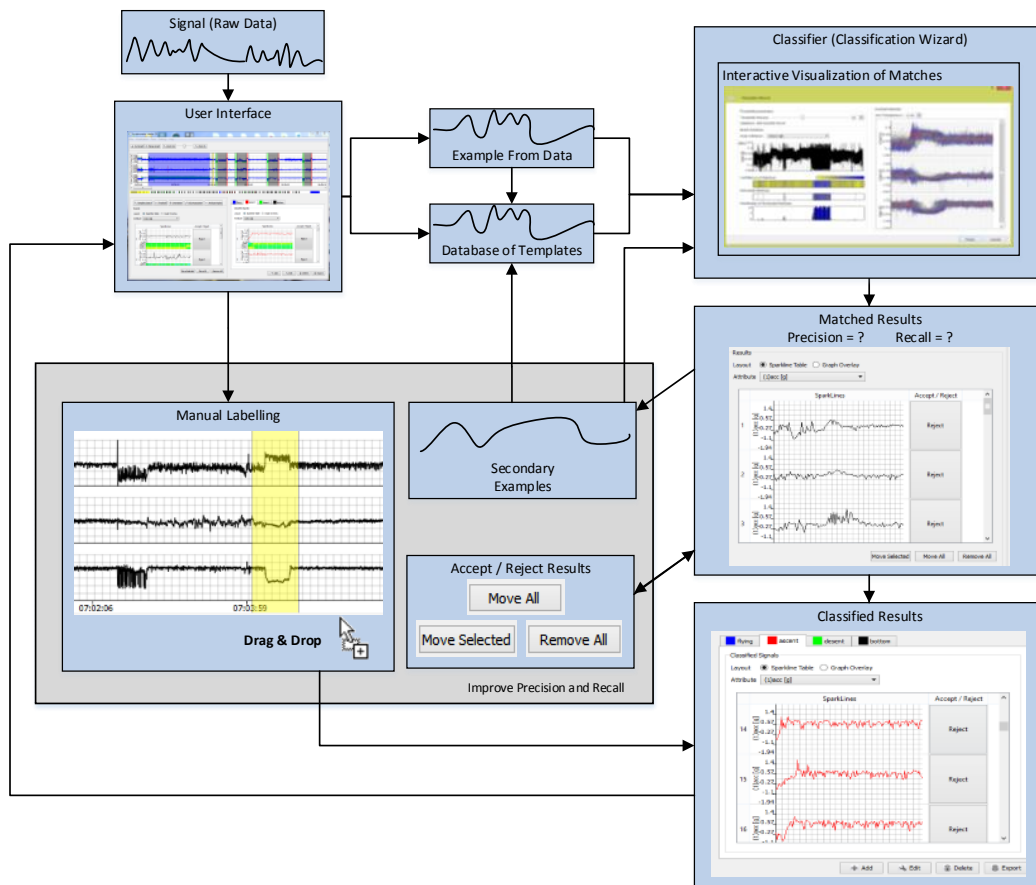


Figure 4.4: This figure shows our proposed system work-flow. Firstly, the user provides one instance of a behaviour from the time-series or from our template database. The classification algorithm then finds all instances of specified behaviour at the selected frequencies on the unlabelled data. The results are presented to the user in the results view. Using our feedback loop, precision and recall can be boosted by providing secondary examples to feed into the classifier or by manually accepting / rejecting matches. Accepted behaviours are moved to the classification panel. The user can manually label data samples to further improve precision and recall is necessary.

user specifies a weighting (algorithm specified in Section 4.4.4) to apply to each dimension which contributes to the overall search.

Database of Templates - Behaviour templates used in the system can be stored in a database for future use. The database is relational with behaviour templates assigned to classes of animals. The user can query for all patterns present for a specific animal or select an existing behaviour template previously saved in the database by navigating to the animal of interest and then selecting the appropriate behaviour template.

Signal Re-Sampling - The signal may be re-sampled to capture events at different frequencies as some behaviours occur at different speeds, for example running. To capture these events independently of the time duration we can store and search for the signal at different time-intervals using re-sampling [Smi97]. Re-sampling is implemented by specifying an irrational factor consisting of an interpolation factor (rate of up-sample) and a decimation factor (rate of down-sample) prior to search. This process must be carried out in the order specified, otherwise part of a desired signal may be removed.

Decimation is a process for decreasing the sampling rate of a signal. A low pass filter is first applied to the signal. We utilise the Gaussian filter due to its smoothing ability through a weighted average and keep every M th sample. The collection of remaining samples forms the re-sampled signal down-sampled by a factor of M .

Interpolation is the process of inserting samples to increase the sampling rate of a signal. Increasing a signal by a factor of L is achieved by inserting $(L-1)$ linearly interpolated samples between each pair of samples to create a higher rate signal.

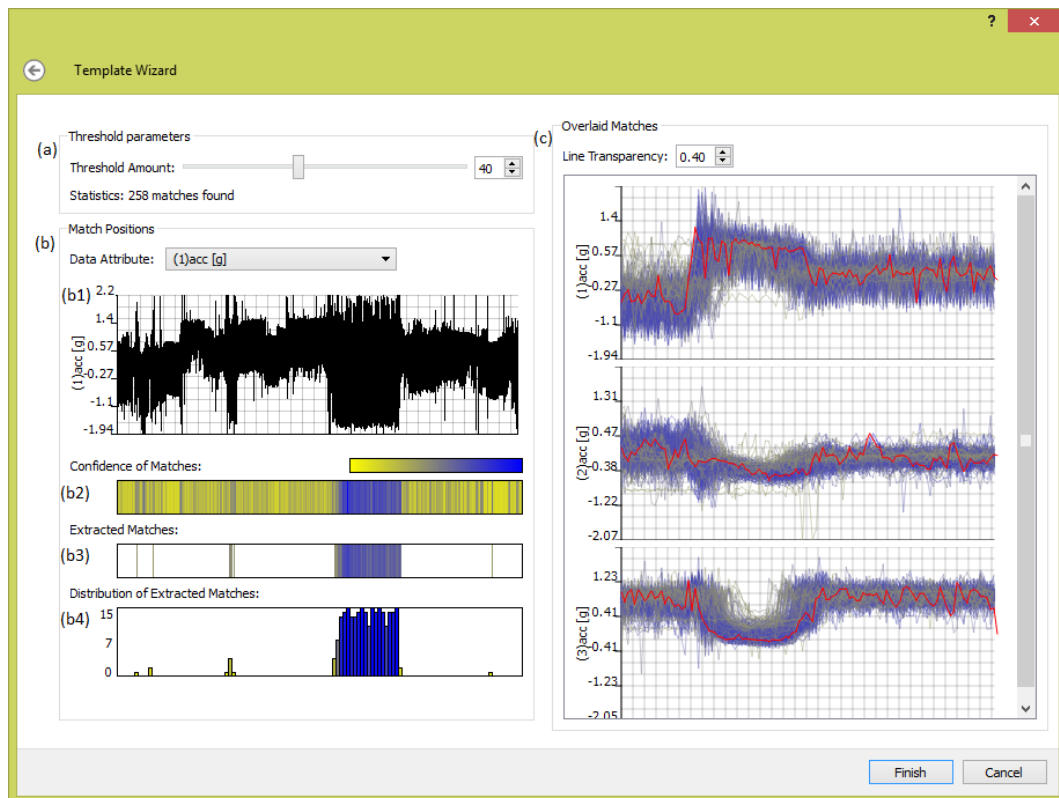


Figure 4.5: This figure shows the classification wizard. (a) illustrates the wizard parameters for dynamically adjusting the threshold. (b) shows the density based visualisations to gain an understanding of where matches occur in the data series. (c) shows the overlaid signals visualisation of all the extracted matches in a stacked graph format, with one graph for each attribute of the template. The template signal is overlaid in red to show a direct comparison. A yellow to blue colour scheme is used, yellow representing low similarity matches, while blue encodes high similarity matches.

Classification Wizard

The algorithm discussed in section 4.4.4 is applied to the data set. The result is the similarity of the specified template at each position in the time-series. Similarity is represented as a percentage of the match, with one hundred percent similarity representing an exact match, whilst for zero there are no matching features. Once the normalised cross-correlation algorithm is executed, the user is presented with the pattern matching results in the classification wizard (Figure 4.5). This is used to guide the user through refining a similarity threshold to test, reject and accept matched signals by applying their expert knowledge of behavioural patterns and temporal position in the data set.

We introduce two views to assist in extracting matching behaviours. On the left (Figure 4.5 (b)) are visualisations to show the temporal positions of matches in the data series, while on the right (Figure 4.5 (c)) we depict all of the extracted matches overlaid on top of each other to show the variance between matches. The visualisations are updated as the threshold value is refined by adjusting a slider corresponding to the threshold percentage (Figure 4.5 (a)). We now detail these visualisations and the associated user-options for accepting and rejecting matches.

Positions of Matches - We contextualise behaviours over the time domain since the biologist may know additional information through prior exploration of the series. We visualise this using three graphical views (Figure 4.5 (b)) spatially aligned with a time-series graph of the data series (Figure 4.5 (b1)). The large nature of our data means the time-series graph is dense. Often this means there can be more than one value per pixel which results in overplotting. We adopt density estimation to represent multiple data items per pixel. The x axis encodes segments of time, such that, each pixel in the display represents many data elements in the series. We compute statistics from each segment of time and represent them using different visual encodings to define our three views.

The confidence of a match visualisation (Figure 4.5 (b2)) depicts a heatmap showing

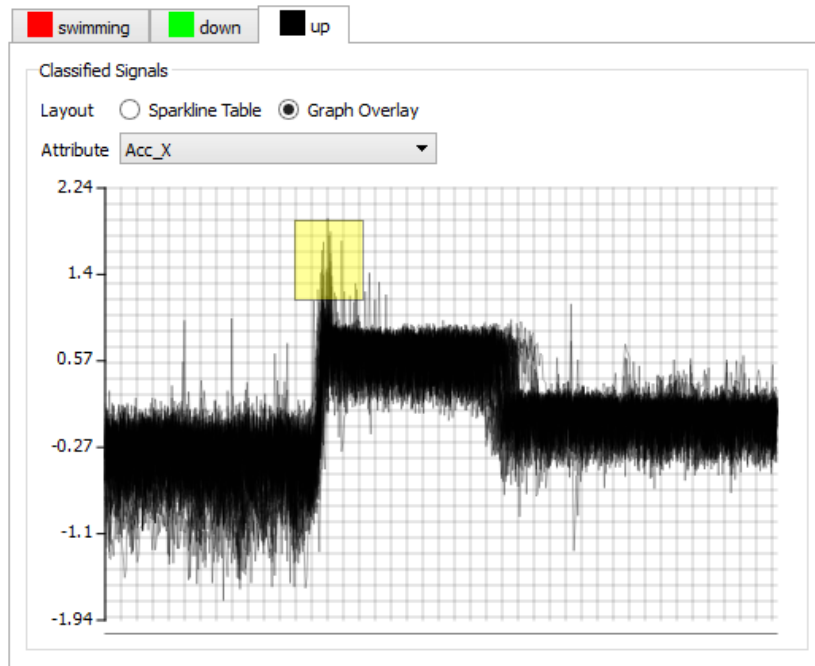


Figure 4.6: The overlaid plot view overlays the classified instances in a time-series graph. The user can accept or reject results by rubber-band selection (yellow) on the time-series.

an overview of the pattern matching results to encode where high (blue) and low (yellow) similarity matches occur in the data series. We map maximum match strength to colour. The extracted matches view (Figure 4.5 (b3)) depicts only extracted matches (above the threshold) and is updated as the similarity threshold is adjusted. Colour encodes average extracted match strength. Finally, a distribution of extracted matches (Figure 4.5 (b4)) which utilises a histogram to map the number of matches at each position to bar height.

The user may refine the result set to reject results from the data series via rubber band brushing across the density displays. Additionally, details on demand can be obtained to delve deeper into the data.

Overlaid Matches - All of the extracted matched signals are overlaid in a stacked time-series graph format, one graph for each data attribute of the pattern (Figure 4.5 (c)). The user can gain an overview of the general shape of the extracted signals from the graphs. This allows the verification of the shape of extracted matches by the domain expert. Furthermore, most outliers will stand out immediately as they will not fit into the general shape of the extracted results. Some matches will get lost in the overall trend, we therefore utilise line transparency to aid in this. Our users indicated that this is one of the most powerful interactive elements of TimeClassifier. Changes in the threshold introduce signals which deviate closer or further away from the pattern template (overlaid in red), enabling the user to directly see the cause and effect of modifying the threshold on the general shape of matched signals.

Results can be rejected in this view by the manual selection of lines on the time-series graphs. All results falling within the selection are removed from the result set.

Matched Results

After the user finds an appropriate threshold value, the results are extracted and added to the results widget in the bottom left of figure 4.3. The user can further inspect the results using our two views. Firstly the separated display, this puts the classifications in a tabular format, with each row corresponding to an identified instance of a behaviour visualised using an embedded time-series plot. The user can accept or reject results by selecting the corresponding button on each row. Secondly, the overlaid plot view overlays the classified instances in a time-series graph. The user can accept or reject results by rubber-band selection on the time-series (Figure 4.6). The overlaid plot view is useful where the signals shape is similar amongst results, conversely the separated display is best suited where each behaviour signal varies widely. Matches displayed in the results view are also shown in the data view overlaid on top of the time-series graph in grey.

Improving Precision and Recall (Feedback Loop)

It is widely accepted in the machine learning community that achieving 100% precision and recall is a difficult, if not impossible task. The variability and inconsistency of animal behaviour is further challenging. In order to support the classification process and boost the precision and recall we propose that the domain expert is involved in the data analysis loop via feedback with the result set. We introduce three methods for this purpose. Firstly, the user can provide secondary examples of a behaviour to find more behaviour instances. Secondly, the user can directly manipulate the result set to accept and reject matches. Finally, the user can manually classify behaviour.

Secondary Examples - Where the user believes recall to be low, boosting can be used to retrieve more instances [FSA99]. Boosting is a machine learning meta-algorithm where-by more instances of a classification are input into the classifier to increase the accuracy of the learning algorithm and therefore the matching ability. To this effect, in TimeClassifier, more examples are selected by the user and input into the search wizard. This widens the search space to find patterns related to the secondary retrieved patterns but may not be directly related to the initial search pattern.

Accept / Reject Results - The results panel provides an effective means to inspect the newly found behaviour classifications. This view allows the user to examine spherical coordinates and parallel coordinates visualisations to assist in determining the correctness of the behaviour classification [WGJL12]. Using these additional visualisations along with the time-series views, the user can choose to accept or reject results for the associated classification. Results are accepted by moving them to an appropriate classification tab in the classification widget, or rejected by clicking the reject button. The user should aim to keep accepting / rejecting results until this panel is empty. The user can manually label any part of the time-series by dragging and dropping it onto the classification label

Classified Results

Classified behaviours are shown in two views. Firstly, the classification widget which displays classified behaviour in a corresponding tabulated view. Secondly classified instances are aligned and overlaid on top of time-series graphs as coloured rectangular regions identifying where in the data a match for the behaviour has occurred (Figure 4.7). We allow the user to collapse the labelled regions of the time-series in this view so they can keep track of the remaining unlabelled signal. Each behaviour is identified by a unique colour assigned to each classification tab in the classification widget. The final result set of labels can be exported to CSV format aligned with the original data file. This is an important feature requested by the domain experts to enable them to utilise results in further experiments.

4.5 Evaluation

The construction of TimeClassifier has been an iterative design process with many user interface and algorithmic improvements to support the biologists way of working. The system has been deployed with 14 biologists using data from animals such as turtles, penguins, condors, and badgers, to name a few. Feedback was collected and improvements were made throughout the project. Many small requests (like derived data and CSV export) were made to increase convenience and reduce the time of use. We present the results of two field trials on the current evolution of our system to demonstrate the usefulness of our technique from a movement ecology context in comparison to the current manual process implemented in our software. The feedback we received has been positive, to quote one expert, *'We are very excited about the power of the iterative approach'*. We then discuss a formal user study to evaluate against the manual approach, followed by a timed user study, and an evaluation against traditional data mining algorithms.

4.5.1 Case Study 1 - Turtles

Biologists are interested in identifying wild animal locomotion because it is one of the most energetically costly behaviours [Ale05]. This is particularly germane in air-breathing vertebrates such as penguins, seals and turtles because the costs of locomotion are dependent on depth, being modulated by the degree of compression of air spaces within the body and resultant upthrust [WHR⁺92]. Specifically, knowledge of when air-breathing vertebrates engage in limb stroke behaviour helps understand power, and therefore oxygen,

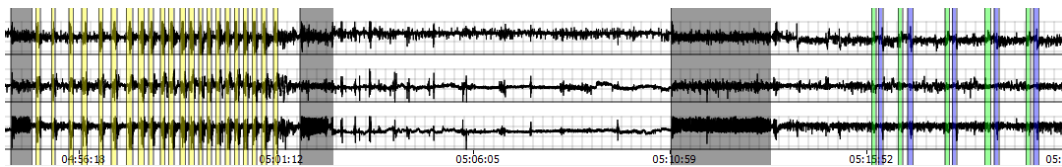


Figure 4.7: This figure shows a subset of approximately 1 hour of labelled ascent (green), descent (blue), swimming (black), and burst swimming (yellow) patterns extracted from data obtained from a Magellanic Penguin by a domain expert using TimeClassifier.

use during dives, and can help biologists understand strategies for maximising performance [SCBN04].

Identification of limb strokes is, however, challenging. Although it can be undertaken manually using acceleration data, with limb beat frequencies of up to several Hertz, in datasets that can last days or even months this is effectively impossible. A particular challenge is that the acceleration channel values change with animal posture [SH08] so that the limb beat signal is superimposed on this variable baseline.

A domain expert analysed a section of a turtle dataset, 154,090 data points of surge acceleration recorded at 40 Hz (using a Daily Diary tag [WSL08]), both by manual inspection and using the program to identify flipper beating, and compared the two performances. In a first, simple run based on a single flipper beat, the program took 12 minutes to set up, modify parameters, and finally run to produce an output compared to 86 minutes for the manual option. TimeClassifier correctly identified flipper beating, and absence of flipper beating 79% of the time. However, manual determination of flipper beating was considered to be highly subjective for at least 20% of the time because turtle flipper beats vary greatly in amplitude, period and even in the extent to which one stroke (wavelength) is executed fully. Thus, the program performed extremely well in identifying effectively almost all clear-cut cases. In a second run, the program was run iteratively with, first, a single turtle flipper beat being used as a template to find its equivalence in the dataset using a high similarity threshold. The data set was then examined to determine which flipper beats had not been found by this process before another single flipper beat was added and the process repeated. Five iterations of this type identified all but an estimated 4 flipper beats. This gave a time-based accuracy of 99.86%. This process also took 12 mins (the faster time being largely due to greater familiarity with the program). Assuming that 12 minutes is the standard time allocated to using the program, and that this effectively varies little with the size of the dataset, the iterative approach (provided by TimeClassifier) takes the same time to categorize about 9 minutes of dataset (21,600 data points) into behaviours as the manual approach. With any comparable data set longer than this, the program is therefore more efficient than the manual approach by a factor of the dataset length (mins)/9 in terms of time.

4.5.2 Case Study 2 - Soaring Birds

Soaring birds such as vultures rely almost entirely on environmentally generated lift to remain aloft [LMT⁺12]. These updrafts may be broadly classified as either slope-generated lift or thermal lift, the latter providing a much greater energetic gain for birds. Identification of where and when birds use the two different lift types is important for conservation but problematic for biologists. Multiple transducers in animal-attached tags can help identify slope-soaring and thermal soaring behaviours but barometric pressure sensors and magnetometers are most useful because pressure transducers can indicate height gain while magnetometers can indicate circling. However, manual inspection of multiple sensors operating at high frequencies is time consuming and requires a great deal of expert knowledge.

A domain expert analysed data recorded during 7 flights of a condor totalling 4 hours (577,770 data points), manually inspecting primarily the pressure and magnetometry data for patterns indicative of thermal and slope soaring. TimeClassifier, was then applied with a threshold in the barometric pressure to identify where lift occurred and repetitive waveforms in the magnetometry data to differentiate slope soaring from thermal soaring. The manual inspection took 110 mins to complete while the program, including parameter modification, took 23 mins. The concurrence between the manual with one iteration of the program performance was 76%, an excellent result given that, as with the penguin data set, there was a substantive amount of subjectivity in the manual interpretation. The second, iterative,

approach (described above) with the vulture data set led to a behavioural accuracy of 100% after 18 mins (the faster time again being largely due to greater familiarity with the program second time around). Using the same criteria for time efficiency as with the turtle case study, the iterative approach (provided by TimeClassifier) is more efficient than the manual approach by a factor of the dataset length (mins)/50 in terms of time for the simple (first) approach.

4.5.3 Formal User-Study

A user study was carried out to assess the effectiveness of our novel approach for labelling animal activity data with respect to the existing manual solution by simulating the behavioural labelling process that biologists undertake. To simplify the study we highlight the behaviours in the time-series with a grey block. Users are required to indicate whether each highlighted signal is behaviour A or B (by right clicking in the block and selecting the appropriate label). The block turns to the colour representing that behaviour.

We provide a counter of the number of segments left to label. We ask users to target getting this to zero, but not to spend minutes looking for the last remaining one or two segments. We time from the presentation of the stimuli until the user clicks next. Accuracy is measured according to how many behaviours are correctly labelled. We obtained a total of 30 participants who each received a \$10 book voucher. They were all computer scientists with average age of 25. The experiment began with a brief overview read by the experimenter using a predefined script. Detailed instructions were then given through a self-paced slide presentation. The presentation included a description of the study and also a briefing on how to interact with the software for labelling purpose. Participants also received a colour copy of the presentation for reference during the study if desired. The study was closely monitored, at least two experimenters were always present in the room and participants abode to the study requirements

We report the average time to label of 91.9 seconds and an average accuracy of 95% - so they get 5% of labels incorrect. There are around 25 labels per stimuli and 12 stimuli overall. Each stimuli was constructed from a small data set (between 25,000 to 30,000 data items in length) where we pre-segment the data and just require users to choose between two labels for each segment. The computed template version achieved 100% accuracy in an average of 1.2 seconds.

4.5.4 Domain Expert User Study

In addition to the formal timed user study with 30 participants on a subset of the data we also conducted a domain expert evaluation with 3 participants on the full data set. We were unable to perform the study on experts in the marine wildlife field (due to the limited number of experts available), we therefore performed the evaluation with computer scientists and biologists involved in smart sensor research. The formal user study (section 4.5.3) used already segmented data and only needed a decision on whether each was to be labelled A or B, the domain expert study used the full unsegmented and unlabelled data set and thus shows a significant difference in the time required to manually label the data.

We obtained ground truth data from domain experts who manually analysed data from a deployment on a Penguin totalling 30 hours (864,319 data points recorded at 8 Hertz). Manually inspecting primarily the accelerometer attributes for patterns of *ascent*, *descent*, *burstswim*, and *swimming* behaviours took approximately seven hours. Our software was augmented with logging capabilities to undertake an informal user study to identify these behaviours. Each participants precision and recall was logged over time, we combine these measures and visualise each participants F-Score (see section 4.3 for derivation) in Figure 4.8. On average, each participant obtained 97.3% precision and 88.9 % recall and took an average of 43 minutes to complete. We believe 100% precision and recall was not obtained because the participants were not experts in this area, and therefore missed the behaviours which required manual and interpretive intervention. It is likely that expert biologists in the field of marine wildlife users would be able to get closer to the 100% (as indicated in the field trials in the chapter) due to their domain knowledge of the signals. Also, the work with biologist experts indicated there is a learning effect using the program, such that it becomes faster with more familiarisation.

Rapid jumps occur in the precision and recall when our pattern searching was used. Slow increases and decreases indicate manual rejection and labelling. The time was notably

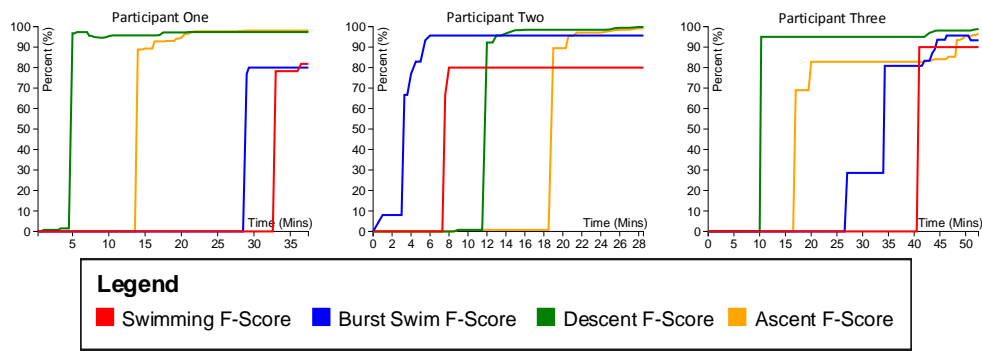


Figure 4.8: This figure shows the three graphs for each of our participants performing behaviour labelling using TimeClassifier. We report the f-score (a combination of precision and recall) achieved by each participant over time, as they label each of the behaviour groupings of swimming, burst swimming, descent and ascent. The X axis encodes time, while the Y axis encodes the F-Score percentage. Line colour corresponds to the F-Score of the specified behaviour detailed in the graph legend.

split between locating templates and applying them to label the data. After applying our classification wizard, the users often went through the data rejecting misclassification's before manually adding missing results. Participant number one had prior knowledge of the data, and therefore produced the highest precision and accuracy, completing the task in the shortest time frame. Participant two and three, had no prior knowledge of the data and therefore initially spent some time familiarising themselves with the data. Interestingly, participant two experimented with different templates often using two or three templates for each behaviour to increase the likely-hood of retrieving all the instances. Conversely, participant three opted to search for all the behaviours before rejecting / accepting results. They then applied additional templates if after visually inspecting the time-series the number of matches was not sufficiently high enough.

4.5.5 Evaluation Against Traditional Data Mining

We compare our visual analytic approach to several other traditional data mining approaches. These are, hierarchical clustering with dynamic time warping (DTW), one nearest neighbour (1NN) with DTW, k-nearest neighbour (KNN), support vector machines (SVM), random Forests and the Time Searcher 2 application. The Penguin data from the case study was further utilised to evaluate TimeClassifier against these methods. Table 4.3 displays results for precision and recall along with time for our approach without utilising any interaction after the search. We applied a constant threshold of 80% throughout the pattern searching wizard with one template instance for each behaviour utilising the x , y , and z axes of acceleration with an equal weighting assigned to each. Near perfect scores are achieved for ascent and descent. Notable is the lower precision score for swimming and burst swimming, due to a low number of occurrences resulting in a low discriminating template.

Behaviour	Time (Milliseconds)	Precision	Recall
Ascent	3244	257 / 269 (96%)	257 / 265 (97%)
Descent	4741	259 / 277 (94%)	259 / 265 (98%)
Swim	5885	9 / 12 (75%)	9 / 11 (82%)
Burst	4975	21 / 25 (84%)	21 / 24 (88%)

Table 4.3: Table illustrating the time taken (in milliseconds), precision, and recall results for TimeClassifier to label Ascent, Descent, Swimming, and Burst Swimming behaviours on a data set obtained from a Penguin deployment.

We trained each of the supervised learning algorithms using six instances of each behaviour, ranging in size from approximately 30 to 2000 data samples in length for each instance. The x , y , and z axes of the accelerometer attribute form the feature vectors for each algorithm. As the data is temporal, each time index was considered as a separate entity to train the classifiers. There was no refinement of the training process, for consistency.

The hierarchical clustering and 1NN algorithms required segmented data as input, to avoid conflicting the experiment we utilise manually segmented data collected from our ground truth data. Therefore, those results are placed in italics and are only included for time and accuracy comparison and to show that even with ideal segmentation they would not perform as well as our approach. The results with respect to precision, recall and time are shown in table 4.4.

Algorithm	Time (Seconds)	Precision	Recall
<i>Hierarchical clustering with DTW (Minimum length: 30, maximum length: 2000)</i>	91	*	*
<i>1NN with DTW (Minimum length: 30, maximum length: 2000)</i>	19	39%	83%
KNN (K = 6)	7	22%	56%
Multi-class SVM (6 classes)	6871	21%	71%
Random Forest	392	96%	16%
Time Searcher 2 (Threshold of 80% with no interaction)	8	35%	47%
Our System (Threshold of 80% with no interaction)	1.9	87%	91%

Table 4.4: Table comparing time (in seconds) along with average precision and recall scores for our TimeClassifier system results with state-of-the-art machine learning methods. We utilise the x , y , and z axes of acceleration as feature vectors for input into the algorithms. We specify the parameters for each machine learning method in the table. 1NN with DTW and hierarchical clustering with DTW are placed in italics since they required segmented data as input. Hierarchical clustering failed to return any meaningful results, we therefore place a star for the results of this algorithm.

Results obtained show low precision and / or recall results for all existing classification techniques. High precision and recall scores are required to avoid a negative trade off. For example, random forests produces the highest precision of 96% but recall is low, only returning 16% of all behaviour instances in the data. Conversely, 1NN returned 83% of all instances, however, only 19% were correct. Hierarchical clustering failed to return any meaningful results. We suspect the results for KNN, SVM, and random forest are lower because they do not consider the temporal ordering of variables, instead operating on the values of each data index as a separate (and therefore distinct) feature for the classification, therefore, an overlap of values between templates increases misclassification. Time searcher 2 is only capable of operating on a subset of 30,000 data items (3% of the original data size). Our solution produces robust precision and recall scores, higher than any of the existing methods utilised. Our results can be increased through interaction, and sufficient domain knowledge. In future work we seek to further validate these findings with experts in the machine learning field who are able to fine-tune the parameters and training process.

4.6 Discussion

Throughout this project our views of the problem faced by the domain experts was redefined as we learned more about the problems faced. Our experts had never consistently used any machine learning techniques previously. We assumed this was a classic problem caused by the opaque nature of machine learning techniques and lack of access to the appropriate techniques. One challenge was no access to extensive labelled collections of behaviour instances, and often contained little value due to the lack of standardised tagging techniques.

We opted to use pattern searching, such that our domain experts can provide one example of a behaviour, of which the software will locate similar instances. Our technique achieved high precision and recall results. However, we realised the biologists required software which would give them the precision and recall rates for classification they require for further analysis. Out-of-the-box solutions do not offer this. Achieving near perfect precision and recall rates is widely accepted as extremely difficult if not impossible in the machine learning community. Our initial system displayed the results to experts but there was no process to interact with the result set to apply their domain knowledge to improve

the precision and recall rates.

We introduce a new classification work-flow which incorporates visual analytics to steer the classification process. Interaction with the result set allows experts to reject, and accept results to improve precision and therefore achieve effective labelling. While users can search using more pattern templates for a specific behaviour or manually add classification to improve the recall rate. A direct comparison with an existing machine learning approach shows the benefit of a visual analytic approach making it possible to achieve high precision and recall compared to much lower values for state of the art machine learning techniques. We provide a solution biologists have confidence in and trust as they can verify results obtained. This new comparison demonstrates the clear benefit of the visual analytics approach, that is, we can achieve higher precision and recall compared to much lower values for state-of-the-art machine learning techniques by involving the domain expert.

4.7 Summary

After working extensively with biologists we have seen beyond standard criticisms of machine learning (i.e. 'black box'). They encounter low precision and recall from these systems and therefore have low confidence in them and express frustration with the black box phrase [vLAA⁺13]. When faced with low precision and recall, the available algorithms do not have an accessible means by which to rectify any mistakes beyond supplying an unknown quantity of further training examples, choosing alternative algorithms or changing parameters (relevancy feedback). Involving the user-in-the-loop overcomes these issues which leads to more trust in results, along with increased efficiency and accuracy.

In this chapter, we present a visual analytics system for the semi-automatic classification of animal behaviour which brings together the expert knowledge and understanding of animal behaviour from the biologist with those of the pattern matching capabilities of the matched filtering algorithm applied to classification. We deployed our system with movement ecologists and report a number of real-world case studies. The results demonstrate the value of visual analytics to making a positive impact on the work of movement ecologists, as the two case studies demonstrate substantial improvement to work-flow efficiency for labelling animal behaviour. We make TimeClassifier available as part of the complete software tool available at <http://www.framework4.co.uk>. It also includes the usability features of the previous chapter, and further details about additional functionality of the software is given in the next chapter.

The normalised cross-correlation algorithm we utilise does not (by default) search for behaviours occurring at different speeds, for example running. Instead, the user has to re-sample their behaviour signals to search at different frequencies. State-of-the-art algorithms like dynamic time warping and longest common sequence apply search using a minimum and maximum length (or duration) bound, although these algorithms incur a greater computational complexity ($O(n^2)$). In future work it would be valuable to investigate an algorithm which can extract behaviours occurring at different frequencies in lower computational complexity. Additionally, the biologists currently have no protocol for tagging animals, this results in the same behaviour producing a different signal between deployments (because the device is mounted at a different position), we offer no method to overcome this issue, other than searching with more templates. In future work, it is essential that a protocol is introduced, or methods are introduced in the software which normalise the recorded sensor data, such that, it is independent of where the device is mounted. The investigation of more complex search parameters is required for detecting advanced behaviours which are independent of the shape of the signal produced. In such a scenario, the user needs to search via other means, for example, threshold analysis, rule based search, and decision trees may be of use to this end.

5. SOFTWARE BEYOND HARDWARE FOR COMPREHENSIVE SMART SENSOR ANNOTATION

“Statistical thinking will one day be as necessary a qualification for efficient citizenship as the ability to read and write.”

– H. G. Wells, *Novelist*

Contents

5.1 Introduction	89
5.2 Background	91
5.3 Implementation	92
5.4 Framework4	92
5.4.1 Loading Data	92
5.4.2 Derivation of Animal Trajectory by Dead-Reckoning	93
5.4.3 Derivation of Animal Behaviour	107
5.4.4 Derivation of Animal Energy Expenditure	107
5.4.5 Derivation of the Physical Characteristics of the Environment	109
5.4.6 Exporting data	109
5.5 Evaluation	109
5.5.1 Software System	109
5.5.2 Impact of Framework4	110
5.5.3 Usage and Statistics	112
5.6 Summary	114

A common feature facilitating the advanced understanding of the biology of wild animals has been the increase in numbers and types of sensors used in smart animal tags, as well as increases in the frequency with which they can be sampled and concomitant increases in memory capacity [ASJM⁺08, HBB⁺11]. Thus, the ability to answer critical questions in biology appears to have been driven to a large extent by advances in technology [RCW05]. These advances in methodology come under two broad areas. One relates to methods that allow tags to be physically attached to animals for increasing lengths of time [WPP⁺97, HBAO⁺01, RA07] while minimising animal detriment [VSGW12, VWW13], the other relates to the physical production of the complex solid-state units in smart tags [MJT04].

The excitement at the potential inherent in sophisticated animal tags has, however, been tempered by a new limiting factor. This is a methodology to deal with the problem of the analysis of the high resolution, multiple channel (and therefore multi-dimensional) data acquired by the tags in short, software [JMF03]. To be most useful to the movement ecology community, software to help in the analysis of smart tag-acquired data needs to be able to deal with large quantities of multiple sensor data and, ideally, should be able to merge different derived analytical outputs together into one output file so that various elements derived from the primary data can be interrelated. Currently, the smart tag community has witnessed a number of software innovations, most of which are concerned with determination

This chapter is based on the following publications:

- Section 5.4.2 is based on ‘Step by step: reconstruction of terrestrial animal movement paths by dead-reckoning’ [BWJ⁺15]
- Section 5.5.2.1 is based on ‘A new perspective on how humans assess their surroundings; derivation of head orientation and its role in ‘framing’ the environment’ [WHW⁺15]
- Section 5.5.2.2 is based on ‘In search of rules behind environmental framing; the case of head pitch’ [WNW⁺15]
- The remainder of the chapter is based on ‘Prying into the intimate secrets of animal lives; software beyond hardware for comprehensive annotation in Daily Diary tags’ [WJL⁺15b]

of behaviour, i.e. from dive profile data [HBH07] or, most notably, from dual-axial acceleration data [WIK⁺05, SSI⁺09b]. Analysis of acceleration data is particularly welcome because inspection of raw acceleration values over time to derive behaviours is not particularly intuitive [SWH⁺08]. Thus, solutions for this have involved a suite of different approaches including cluster analysis [SSI⁺09b], support vector machine classification models [MJS⁺09] and artificial neural networks [NSFR⁺12].

In 2008, Wilson et al. [WSL08] put forward a concept for a particular sensor configuration within a tag that they called the Daily Diary (DD)[WSL08]. Specifically, the suggestion advocated the combination of tri-axial accelerometers, tri-axial magnetometers, and pressure and speed transducers together with environmental sensors such as temperature, light and humidity. The theory was that this configuration of sensors, sampled at infra-second rates, would allow tag users to be able to derive four key attributes of animal ecology seamlessly. These are: (1) animal trajectory, and therefore position [WLD⁺07] (2) animal behaviour [SWQ⁺08] (3) energy expenditure [WWQ⁺06] and (4) the environmental conditions to which the tag carrier is exposed [WGS⁺02]. Although this original work did indicate avenues by which these elements might be achieved, there was no specific suggestion of software that might do this. In short, currently, although some software is available to help determine some aspects of that advocated by the daily diary concept [BCGL⁺14], there is nothing that binds the concepts together.

In the previous chapter, we introduced TimeClassifier, a visual analytic system for behaviour recognition of sensor data ((2) of Wilsons [WSL08] key attributes). In this chapter, we proceed TimeClassifier to create a new software package which computes the remaining three pivotal data attributes advocated by the daily diary concept (this corresponds to the domain level challenge 5, identified in chapter 1 section 3). We call this software package, Framework4, since it is a framework for the four core attributes advocated. This chapter describes the structure and functioning of Framework4 that allows the users of smart tags to calculate all four key attributes advocated by the daily diary and then to bind them together into one single output file so that workers can subsequently obtain geo-reference behaviours, energy expenditure and environmental parameters merged into one data file to gain a more holistic picture of how animals react to and within their environment. To achieve this, we introduce a novel algorithm for deriving animal movement reducing accumulated errors from inertial sensors otherwise limiting this approach. We demonstrate our software on data recorded using a daily diary, however, the software can be applied to any data formatted to our input specifications (see section 5.3.1).

Previous solutions for deriving Wilson's core attributes have been disjoint and largely distributed as packages for statistical programming languages which require advanced experience with the associated languages (further discussed in section 5.4.2). Framework4 operates using a wizard approach informed by usability practice and guidelines [Dix09]. The aim of Framework4 is to provide an all-encompassing software solution which abstracts from the underlying mathematical operations and algorithmic procedures with user interfaces such that no knowledge of the underlying process is required. The resulting solution allows biologists to focus on their research and achieving the results required to advancing knowledge. We hope this software will assist in the understanding of wild animal ecology, providing new insights as a result of advanced computing knowledge.

Our work consists of the following contributions:

- We conceive and validate a corrected dead reckoning algorithm which corrects cumulative errors from the estimates of derived heading and speed associated with conventional dead reckoning.
- The impact of Framework4 in the movement ecology community has allowed the realisation of a head tracking device which enables researches to derive and inspect previously unattainable metrics.

The rest of this chapter is arranged as follows. In section 2, we investigate the current challenges faced by our collaborators and the benefits of Framework4. In section 3, we outline the design and implementation of framework4. In section 4, we identify the methods and user interfaces behind framework4 which bridge the gap between users and underlying algorithms to make them readily accessible to movement ecologists. In section 5, we evaluate the methods behind framework4. Finally, in section 6, we summarise our findings and outline future work.

	A	B	C	D	E	F	G	
1	East	(1, 0, 0)	Assumption is that the longitudinal axis of the animal is the x- axis					
2	North	(0, 1, 0)						
3	Up	(0, 0, 1)						
4								
5		Degrees	Minutes	E/W	Radians			
6	Declination	0	44.439284	W	-0.012926864			
7	Inclination	-5	-40.89978		-0.099163726			
8								
9								
10		East	North	Up				
11								
12	h	-0.012863	0.995004165	0.099001286				
13								
14	g	0	0	-1				
15								
16	h x g	-0.99500417	-0.012863	0				
17								
18	h.g	-0.09900129						
19								
20								
21	-0.09900129	scalar product of g and h (unit vectors)						
22								
23	Observed data							
24	Accelerometer	Accelerometer	Accelerometer	Magnetometer	Magnetometer	Magnetometer		
25	X	Y	Z	X	Y	Z		
26								
27	-0.523241	-0.851869	-0.0231812	0.591853	-0.226263	-0.773638		
28	-0.900005	-0.425664	0.093812	-0.143097	0.691243	0.708312		

Figure 5.1: A spreadsheet developed by movement ecologists in the SLAM research group for deriving animal trajectories. The spreadsheet consists of several configuration parameters. The coupling of how the configuration maps to the resulting trajectory is not clear.

5.2 Background

In the previous chapter, we introduced TimeClassifier, a visual analytic tool for identifying and labelling animal behaviour. The capabilities of TimeClassifier were empowering to the biologists, accelerating the process for labelling animal activity which was otherwise an entirely manual and time-consuming process. However, behaviour is just one of the derived attributes which is typically required as a precursor before any advanced knowledge and insights can be drawn from the data.

Wilson et al. [WSL08] specify four pivotal attributes to the smart sensor SLAM concept for understanding animal activity, these are: (i) The derivation of animal behaviour (supported via TimeClassifier) (ii) Animal trajectory to understand animal movement. (iii) Energy expenditure to understand how much energy the animal is using. (iv) Environmental conditions by accessing the data together in tandem to understand the context in which animals operate.

The current method of derivation for these attributes consists of exporting labelled data directly from TimeClassifier into a commercial spreadsheet software tool. The biologists have a set of pre-defined spreadsheets which they configure for their smart sensor tag and execute on the exported series. The spreadsheets have had limited success, especially for complicated procedures where it is not clear how to utilise the spreadsheet effectively. Figure 5.1 illustrates an example configuration for deriving animal movement. The computation of animal trajectory relies on a number of configuration parameters which is difficult to interpret and validate the impact of different parameter settings, especially where guidance is needed for appropriate values.

The spreadsheets facilitate basic functionality but are difficult when the solution requires specialist software and complicated algorithms. The biologists now have many challenges (e.g. dead reckoning coupled with GPS correction) which require advanced software solutions made accessible by purpose built user interfaces.

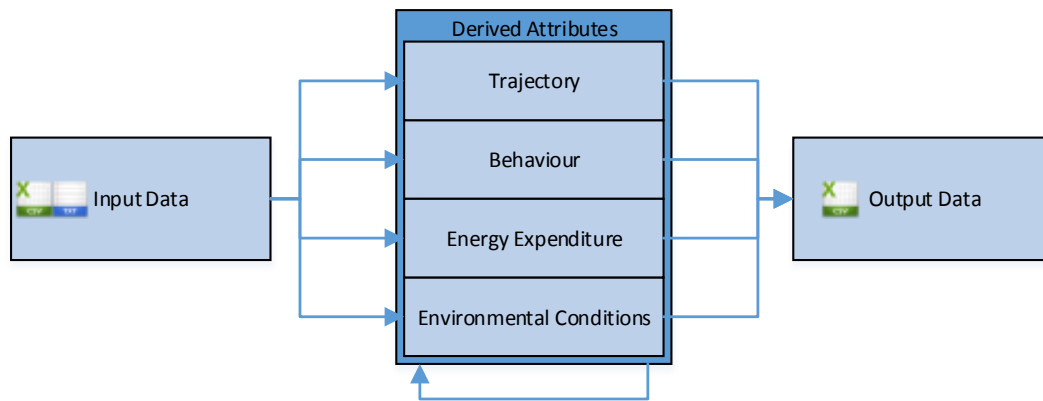


Figure 5.2: This figure shows the Framework4 analysis pipeline. The user inputs their sensor data in CSV or tab delimited format. After this, the user applies any combination of our behaviour analysis, energy expenditure, trajectory and environmental condition derivation tool kits. Finally, the data is exported into a unified CSV file with the derived data channels time aligned with the original input data.

5.3 Implementation

Framework4 is a software package for the Microsoft Windows operating system for analysing smart sensor data. We create a desktop application as it allows us to handle large data files effectively (tested on over 10 million data samples) on standard computing equipment. Utilising a desktop application, we directly communicate with the CPU for efficient data handling, and the GPU for visualisation purposes. Our tool can be used in the field, in remote areas during deployment where there is often no internet access.

We implement Framework4 using the programming language C++, with Qt [BS06] which provides an application framework for developing software (with user interfaces). We develop using these languages as they directly compile to machine code which is optimised for efficiency when performing data intense operations. We distribute both 32 bit and 64 bit versions of the software, since 64 bit allocates the software a larger memory capacity which is necessary when analysing larger data sets, commonplace given the storage capacity and logging durations utilised.

5.4 Framework4

Figure 5.2 illustrates the work-flow of Framework4. The software accepts as input, comma separated values (CSV), or tab delimited formatted data. Once imported, the user can apply our behaviour analysis, energy expenditure, trajectory, and environmental condition derivation frameworks to derive the associated data dimensions. After the user has obtained the data attributes they require for analysis, the data can be exported into one unified CSV file with all the attributes aligned. Design choices have been made at each stage to assist the user in their tasks, for example, making use of wizards to lead the user through various processing tasks. Each of the software features and the underlying methods by which they operate is now discussed.

5.4.1 Loading Data

The software supports two file formats; comma separated values (CSV) and tab delimited formats. These are two of the standard file formats for storing flat file data in text format and are common outputs from commercial smart sensor tags. The daily diary exports its data in a binary format which gets segmented into multiple files and converted to tabular delimited format post-deployment. Framework4 loads and operates on the individual tabulated files.

We incorporate an import wizard in the application to import data files (Figure 5.3). Here, the user can specify the names and data types of each data attribute, which are used as a reference to the data channels throughout the application.

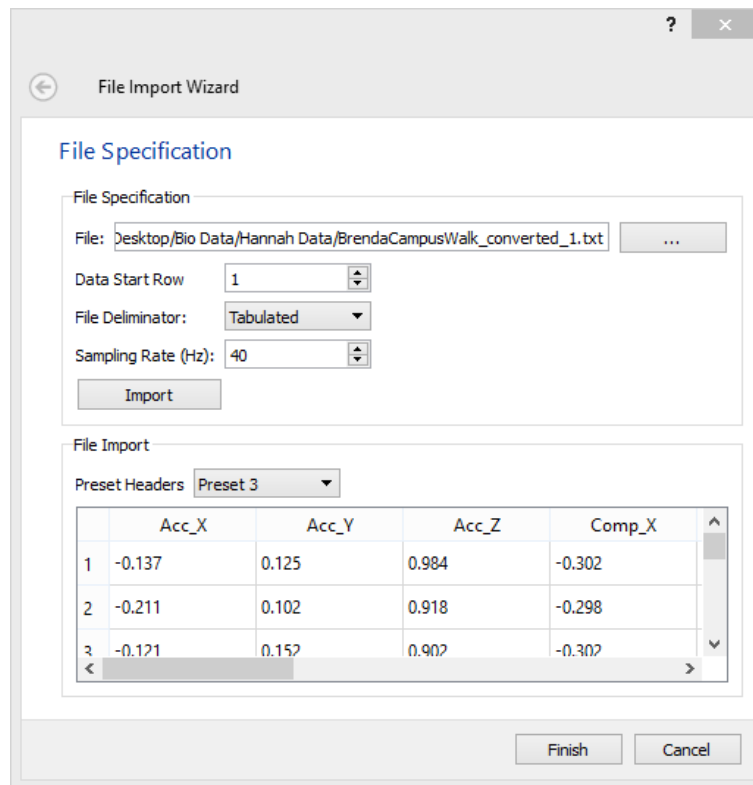


Figure 5.3: The data import wizard provides an intuitive, but flexible user interface for importing data from CSV and tabular delimiter file formats. The user specifies the start row of the data, along with the file delimiter. A preview of the imported data is shown in a tabular layout, the user can rename headers using a pre-set or by extracting them from the imported file. The specified attribute names are used to reference the data channels in the software.

5.4.2 Derivation of Animal Trajectory by Dead-Reckoning

The way in which animals use the environment is fundamental to understanding their behavioural ecology [BDF08] and, as such, many different systems have been developed to examine animal movements (see Wikelski et al [WKK⁺07] for a comprehensive survey). To date, GPS has been popular for animal movement. However, fine infra-second scale behaviour cannot be obtained as it has an approximate accuracy of 5 meters [BWJ⁺15]. A relatively recent addition to the field which can achieve this is dead-reckoning [WSL08] which operates on the basis that a position at time (t) can be derived given the position at a previous time ($t - 1$) coupled with the distance and heading taken between the two time intervals. Dead-reckoning has received little interest until now, partially because early systems for dead-reckoning were crude [IDDB94, WW88, WCDA91] but with the development of sensors and techniques, headings can be computed to within 1 degree of accuracy utilising accelerometer and magnetometer sensors [Car00, WSL08] by measuring the earth's magnetic field.

The earth's magnetic field is constructed of field lines approximating a magnetic dipole. Each field line originates at a point near the magnetic South Pole and terminates at a point near the magnetic North Pole (Figure 5.4). Measuring the strength and direction of the field lines using a tri-axial magnetometer can obtain a relative compass heading in respect to magnetic north. The relationship between magnetic north and geographic north is defined by a declination angle which varies across the earth's surface and with time. The angle of declination can be obtained from a reference table provided by the National Geophysical Data Centre (<http://www.ngdc.noaa.gov/geomag-web/>). Applying the declination angle to the magnetic north heading obtains a geographic heading. Dead reckoning couples heading with speed estimates from accelerometer data to derive position and therefore trajectories.

Dead reckoning has particular value in not being dependent on transmission technology, and therefore can be used in challenging environments (e.g. underwater) since magnetic field lines can permeate through water where other technologies (e.g. GPS) fail to operate. However, location cannot be inferred accurately from inertial sensors due to error accumula-

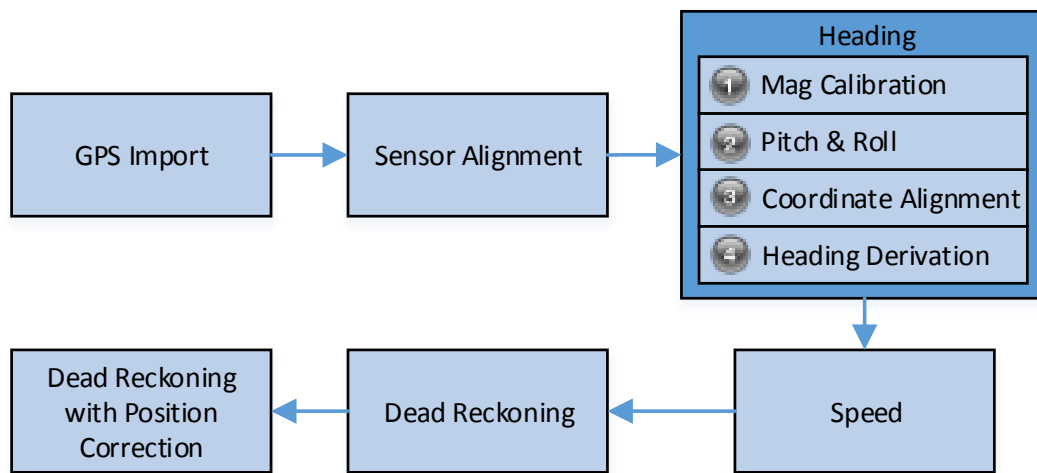


Figure 5.5: Framework4 performs dead reckoning in the following way. Firstly GPS data is imported. The data is synchronised by comparing the time-stamps with the original data file. The sensor coordinate systems are then aligned to ensure all sensors are registering the same frame of reference. Heading is derived by performing several processing stages of; calibration of the magnetometer sensors, computing pitch and roll, coordinate alignment, and finally heading derivation. After this, speed is derived. Finally, the heading and speed are combined to derive a dead reckoned trajectory. The dead-reckoning path is corrected using external positioning information to eliminate accumulated errors in the final trajectory obtained.

tion. Even small errors (less than 1 degree) in the data recorded from the sensors can lead to large displacements in the resulting trajectory over time. What we achieve in this software is a framework that allows researchers to experiment with location fixing data channels combined with motion channels to provide corrected dead reckoning. It is the hypothesis of this work that integrating sensor data into dead reckoned trajectories enable the accurate derivation of fine scale behaviour. We introduce a novel algorithm coupling together external positioning with traditional dead reckoning for resetting error accumulation, a facet of the traditional dead reckoning approach. We expose these methods to the wider community through the framework4 software.

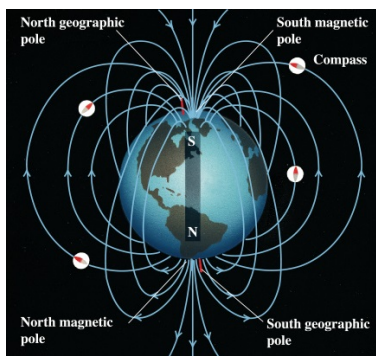


Figure 5.4: The earth's magnetic field is constructed of field lines. A declination angle determines the offset between magnetic and geographic north. Image Credit: Addison Wesley Longman, inc.

user to see and access the underlying derived data at each step of the dead-reckoning procedure to provide data transparency. Furthermore, the dead-reckoning aspect is tied in closely with behaviour analysis functionality which is not offered in any existing tools.

In Framework4, we introduce a user-friendly wizard for performing standard dead-

Many algorithms have been introduced to this end for dead-reckoning [JT03, MS1⁺03, WLD⁺07] to provide new insights into animal movement. These methods have been made available to the research community through statistical software packages. Firstly Shimomi et al. [SNS⁺10] introduce “ThreeD_path”, a library for the Matlab statistical package based on the dead-reckoning algorithm by Johnson and Tyack [JT03] for reconstructing 3-D paths. More recently, Battaile [Bat14] present the “TrackReconstruction” R package to perform dead-reckoning and enable visualisation of the derived trajectories. Both of these packages hold value for those with experience with the associated statistical programming languages, allowing direct manipulation of the methods used and data supplied. However, for those without any background with these applications, the learning curve can be steep, and may appear non-trivial to those with limited experience. We provide a user-friendly protocol which requires no programming experience, allowing the

reckoning on data with tri-axial magnetometer and tri-axial accelerometer components (Figure 5.5) accessible via the tools menu. Dead-reckoning is subject to cumulative errors and, as the heading and speed are estimates, any systematic deviations from the actual heading and speed will lead to increasing errors. To assist in reducing such errors, we introduce a novel corrected dead-reckoned algorithm which utilises supplementary positional information often recorded alongside the daily diary data (i.e. GPS fixes). Our algorithm applies such positional data as ground truth positions to force the dead-reckoned path to align. This resets any cumulative error at each positional fix, but does not factor in any errors associated with the underlying ground truth position. For example, the GPS signal may feature its own set of errors as its accuracy is only to within 5 metres [BWJ⁺15]. Incorporating external errors into the underlying algorithm is of interest for future work in this area, although not a focus of this work since the ground truth positioning data can come from any source, and therefore future technological developments in this area can be integrated into the software without any changes (e.g. differential GPS). The advantage of dead reckoning in comparison to traditional positional logging techniques in the community (e.g. GPS) is that they are unable to detect fine-scale animal movement. Dead reckoning is high frequency and able to detect such movement, however, cumulative errors occur over time which can lead to unacceptable deviations in the resulting trajectories. Combining the two technologies to create a hybrid approach allows us to obtain fine-resolution movement with cumulative errors reset using low frequency location fixes.

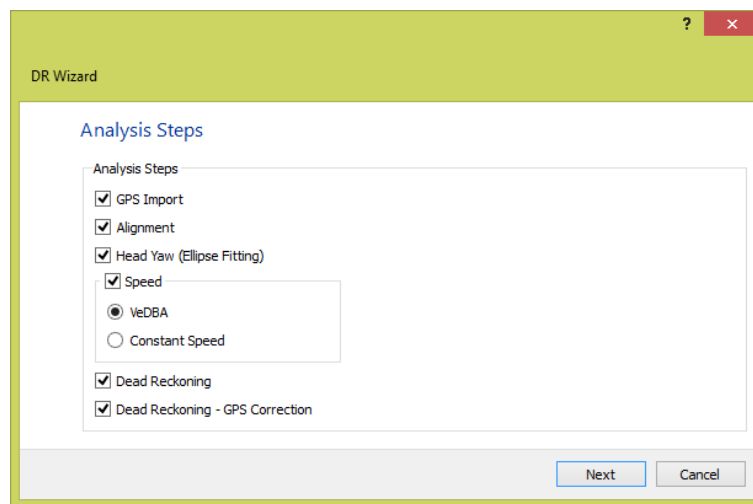


Figure 5.6: The dead-reckoning wizard features a number of modular classification steps. These are; GPS import, alignment, heading, speed derivation via VeDBA or applying a constant, dead-reckoning, and dead-reckoning with GPS correction

Each of the components of the dead-reckoning wizard are modular so that the user can select which analysis steps are required (we expose these in the main wizard page shown in figure 5.6). The steps are; (i) GPS import for synchronising time-stamped GPS (or similar) data with the original data file (ii) alignment correction for the accelerometer and magnetometer coordinate frames (iii) heading derivation from the magnetometer channels (iv) speed derivation via a proxy derived from acceleration for obtaining an estimate of the speed, or constant speed options (vi) dead-reckoning to combine the heading and speed information to derive a trajectory and (vii) dead-reckoning correction using external positioning information to eliminate drift in the final dead-reckoned path. We now introduce each of these methods in turn, in addition to detailing the software front-end.

5.4.2.1 GPS Import

It is often the case that supplementary positional information is recorded from separate data sources to the rest of the data. For this reason we can import positional data (e.g. GPS which we use as an example throughout the remainder of this chapter) from separate data files and merge them into the current operational data set, although both are assumed time-synchronised. The same import wizard is used as in the file importer to import the geo-referenced data. Once the user has imported the data they then select the relevant time fields in both the data sets to use for synchronisation (i.e. day, hours, minutes, seconds,

and milliseconds). The wizard (Figure 5.7) matches the time index columns and appends the additional data fields wherever a matching data item with the same time stamp is found. Where no aligning row exists, 'null' values are used, which are ignored during the dead-reckoning procedure.

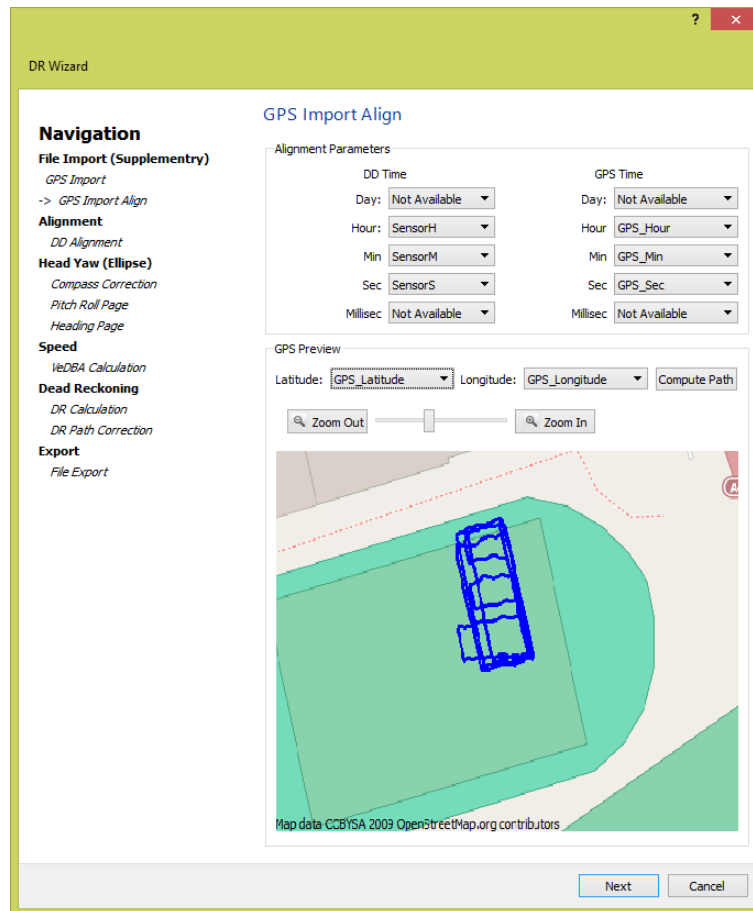


Figure 5.7: The user selects the relevant time fields (i.e. day, hours, minutes, seconds) in both the geo-referenced and smart sensor data sources to synchronise them into one data set. A graphical preview of the geo-referenced data is shown in the bottom of the panel.

5.4.2.2 Alignment

Alignment of the accelerometer and magnetometer coordinate systems is vital for computing heading as rotational information derived from the accelerometer channels is applied to the magnetometer channels. Within the daily diary system, the coordinate frames of both sensors are not aligned (that is, the magnetometer and accelerometer channels are not pointing in the same directions), therefore adjustment must take place. We advise users to check their sensor documentation or contact the system manufactures for this information such that the transformation which must take place for alignment is identified. Framework4 offers a pre-specified transformation for the daily diary system. In addition to this, to compute the correct pitch and roll of the device along with device heading, the orientation of the device on the attached body must be known and corrected. To deal with this, the program asks the user to specify the orientation of the device. As the user selects different orientations, an image of the coordinate frame is updated appropriately (Figure 5.8). After the correct settings have been chosen, the sensor attributes are transposed by applying an appropriate transformation matrix to the accelerometer and magnetometer data channels [BWJ⁺15].

5.4.2.3 Heading

Deriving the heading necessitates that the data attributes be sanitised via; (i) magnetometer calibration, (ii) pitch and roll computation, (iii) coordinate frame adjustments, and finally (iv) heading derivation. These processes are executed across separate pages of the wizard.

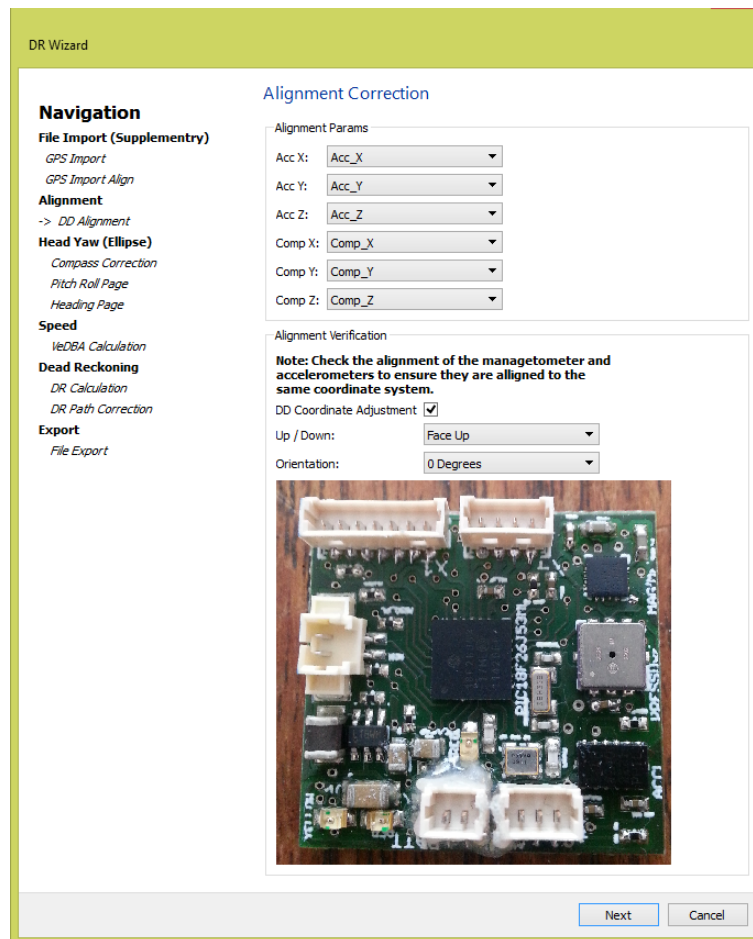


Figure 5.8: Since the accelerometer and magnetometer coordinate systems may not be aligned. We allow the user to adjust the data channels by a pre-defined transformation function for the Daily Diary system. The user selects the accelerometer and magnetometer data channels, along with the orientation of the tag during deployment. When adjusting the orientation, the image of the sensor (bottom) is updated to reflect the settings specified.

The user is exposed to the algorithm parameters at each step with the corresponding results visualised in tandem to allow the user to obtain direct feedback on the effect of the configuration selected. For clarity, we now introduce the methods for each process, while defining the rationale at each step.

Magnetometer Calibration

The measurements obtained by commercial magnetometer sensors are corrupted by several sources of errors [RAL10]. Proper calibration of the magnetometer is required to obtain high accuracy heading measurements. Inconsistencies are usually introduced by instrumental errors, such as scale factors, non-orthogonality between axes, offsets and magnetic deviations caused by perturbations and interference with the magnetic field lines.

Magnetic measurements are subject to sources of error primarily caused by hard iron and soft iron deposits acting on the magnetic field (Figure 5.9 illustrates the problem simplified to two-dimensions, although the problem requires a solution in three-dimensions). Rotating a magnetometer around 360 degrees in all orientations under no sources of error should produce a perfect sphere centred on the origin. Hard iron effects are created by objects which produce a magnetic field with a constant bias in the output, resulting in a sphere displaced from the origin. Soft iron deposits are caused by ferrous materials which are more permissive to the magnetic field influencing the magnetic field lines as they pass through causing distortion and stretching in the surrounding field. This distorts the sphere into an ellipse as hard iron errors are independent of the orientation of the device and can therefore occur across the sphere. Hard iron distortions are caused by metals such as nickel and iron and commonly have a much larger contribution towards the total error [CAM01].

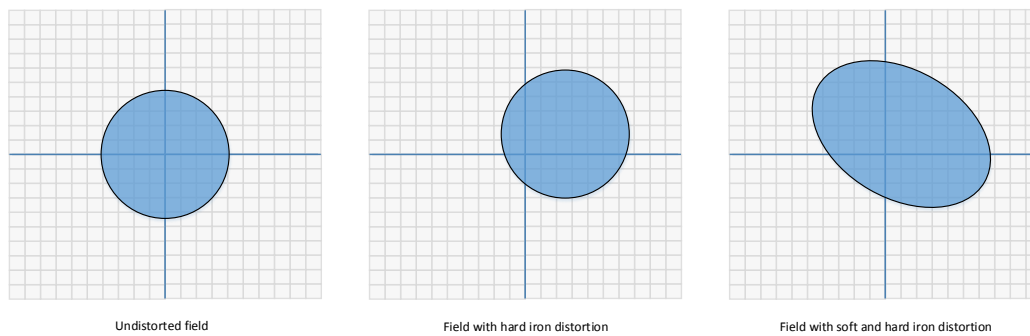


Figure 5.9: Rotating a magnetometer around 360 degrees should produce a perfect sphere centred on the origin (left). When hard iron effects are present a constant bias is present (centre), resulting in a displaced sphere from the origin. Soft iron deposits distort or stretch the field (right).

We utilise the state-of-the-art error model presented by Renaudin et al. [RAL10] consisting of an ellipsoid-fitting algorithm based on an adaptive least squares estimator which calibrates the magnetometer readings for both instrumental errors and magnetic deviations. Prior to deployment the user is required to reposition the device at a number of orientations to obtain a complete range of values which are then fitted to an ellipse across the data series. A transformation function is computed to transform the ellipsoid to a sphere and therefore remove any errors caused by hard and soft iron distortion. The transformation matrix is applied to the magnetometer data channels after. The user selects the magnetometer data channels which are used to compute the correction matrix (Figure 5.10). The user can view and adjust the given matrix which can be exported and applied to other data sets from the same deployment.

Pitch and Roll Computation

To determine the heading of a device affixed to an animal, the magnetometer should ideally have an $x y$ plane that is parallel to the surface of the earth, something that is not always possible given the mounting position and uncertainty of animal behaviour. This is problematic when considering that tilting the device alters the output of the magnetometers registered declination and inclination angles. The pitch and roll values are used to perform tilt correction which rotates the magnetometer values onto an $x y$ plane parallel with the earth's surface. Calculating pitch and roll requires decimating the acceleration channels into static and dynamic components. We outline the process for deriving the static and dynamic acceleration before identifying obtaining device orientation which is then used to adjust the coordinate frame appropriately.

Static and Dynamic Acceleration The static component of acceleration is derived by passing a sliding moving average over the acceleration axes [SWH⁺08]. The central moving average filter derives a set of averages by iterating a moving window over the whole data series. Upon each iteration, an average of all the data items within the given window is computed. Given a data sample S and a window size w , a moving average can be computed using formula 5.1. Shepard et al. [SWH⁺08] provide guidelines on selecting an appropriate window size. Although, often a window size of twice the recording frequency is appropriate.

$$S_i = \frac{1}{w} \sum_{j=i-\frac{w}{2}}^{i+\frac{w}{2}} S_j \quad (5.1)$$

The raw components of acceleration ($RawAcc_x$, $RawAcc_y$, and $RawAcc_z$) are smoothed using formula 5.1 to obtain the associated static components of acceleration ($SAcc_x$, $SAcc_y$, and $SAcc_z$). The dynamic components of acceleration ($DAcc_x$, $DAcc_y$, and $DAcc_z$) are computed by subtracting the static acceleration components away from each associated raw component.

$$DAcc_x = RawAcc_x - SAcc_x \quad (5.2)$$

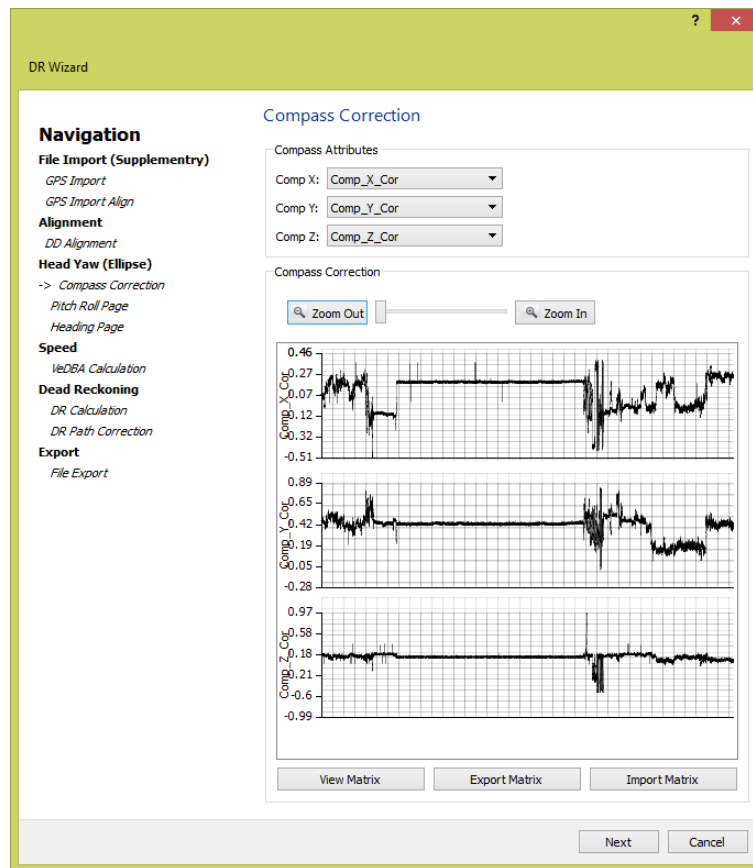


Figure 5.10: The user selects the magnetometer data channels to perform error correction on. A graphical preview of the corrected channels is shown in a stacked time-series graph. The user can view the correction matrices applied, along with exporting / importing them from other data sets where the initial calibration took place.

$$DAcc_y = RawAcc_y - SAcc_y \quad (5.3)$$

$$DAcc_z = RawAcc_z - SAcc_z \quad (5.4)$$

Device Orientation Pitch and roll are calculated as rotations in the heave (x) and surge (y) axes. For clarity, a tri-axial accelerometer records acceleration in the heave, surge, and sway axes, corresponding to the dorso-ventral, anterior-posterior and lateral axes of the animal respectively [SWQ⁺08]. If the static acceleration of heave, surge, and sway are denoted by $SAcc_x$, $SAcc_y$, and $SAcc_z$, then pitch is calculated using formula 5.5 and roll using formula 5.6. This calculation provides pitch and roll in radians, so the presence of $180/\pi$ converts the result to degrees.

$$Pitch(\beta) = \left(\arctan \frac{S_y}{\sqrt{S_x \cdot S_x + S_z \cdot S_z}} \right) \cdot \frac{180}{\pi} \quad (5.5)$$

$$Roll(\gamma) = \left(\arctan \frac{S_x}{\sqrt{S_y \cdot S_y + S_z \cdot S_z}} \right) \cdot \frac{180}{\pi} \quad (5.6)$$

Figure 5.11 illustrates the associated user interface for computing device orientation. Here, the user selects the accelerometer attributes, along with the window size to use for deriving the static component of acceleration (equation 5.1). A time-series graph shows a preview of the derived pitch and roll. The nature of the software allows the user to experiment with different window sizes to see the direct impact of adjustments on the resulting pitch and roll before continuing.

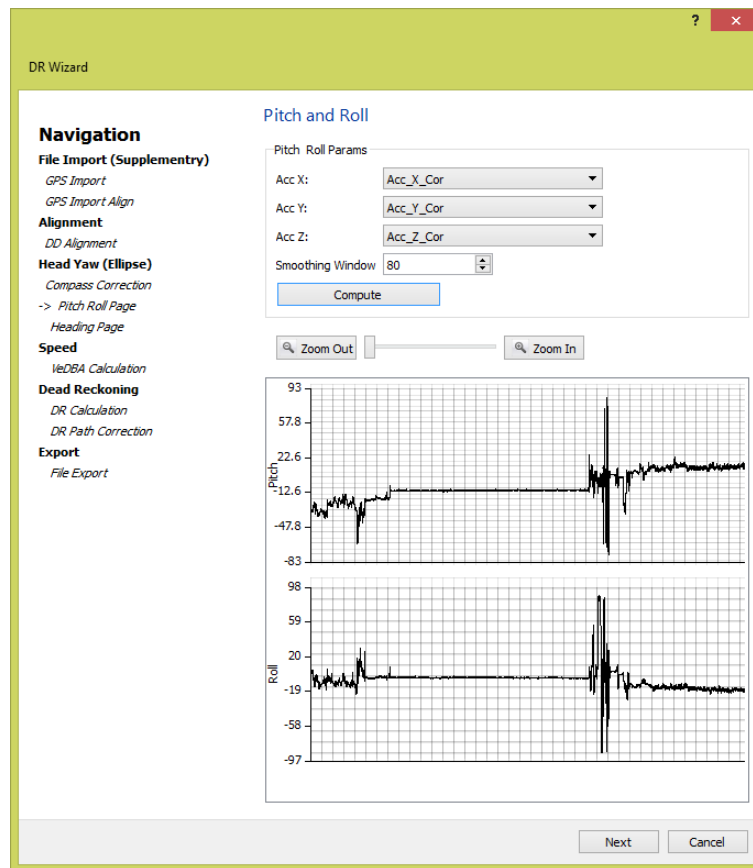


Figure 5.11: Tilt correction computes the pitch and roll of the device during deployment and reverses these offsets on the magnetometer data channel. The user specifies the accelerometer components, along with the smoothing window to use to compute orientation. A graphical preview of the pitch and roll are displayed in a stacked time-series graph for verification purposes.

Coordinate frame adjustment

Device attitude via pitch and roll can be used to rotate the magnetometer measurements to bring them back level with the surface of the earth. Within this step the pitch and roll measurements are used to obtain tilt corrected magnetometer channels ($RMag_x$, $RMag_y$, $RMag_z$) from the associated raw magnetometer readings (Mag_x , Mag_y , and Mag_z) using formula 5.7.

$$RMag_i = Mag_i \cdot R_y(\beta) \cdot R_x(\gamma) \quad (5.7)$$

The rotation matrices [BHSM04] for pitch (γ) and roll (β) are given as $R_x(\gamma)$, and $R_y(\beta)$, expressed by formula 5.8 for the x axis, and formula 5.9 for the y axis.

$$R_x(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{pmatrix} \quad (5.8)$$

$$R_y(\beta) = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \quad (5.9)$$

In the wizard, the user selects the pitch and roll channels to use for adjustment. This allows the user to select non-derived values, for example, pitch and roll from a gyroscope. Otherwise the pitch and roll values calculated prior should be selected. The user clicks compute and can preview the rotated magnetometer channels in a time-series graph view.

Heading derivation

The compass heading (H) with respect to magnetic north is determined using the pre-

viously computed $RMag_x$ and $RMag_y$ tilt- and error-corrected magnetometer components utilising formula 5.10.

$$H = (\arctan(RMag_y / -RMag_x)) \cdot \frac{180}{\pi} \quad (5.10)$$

In framework4 the user selects the adjusted magnetometer components and clicks compute (Figure 5.12). A preview of the derived heading is displayed for verification.

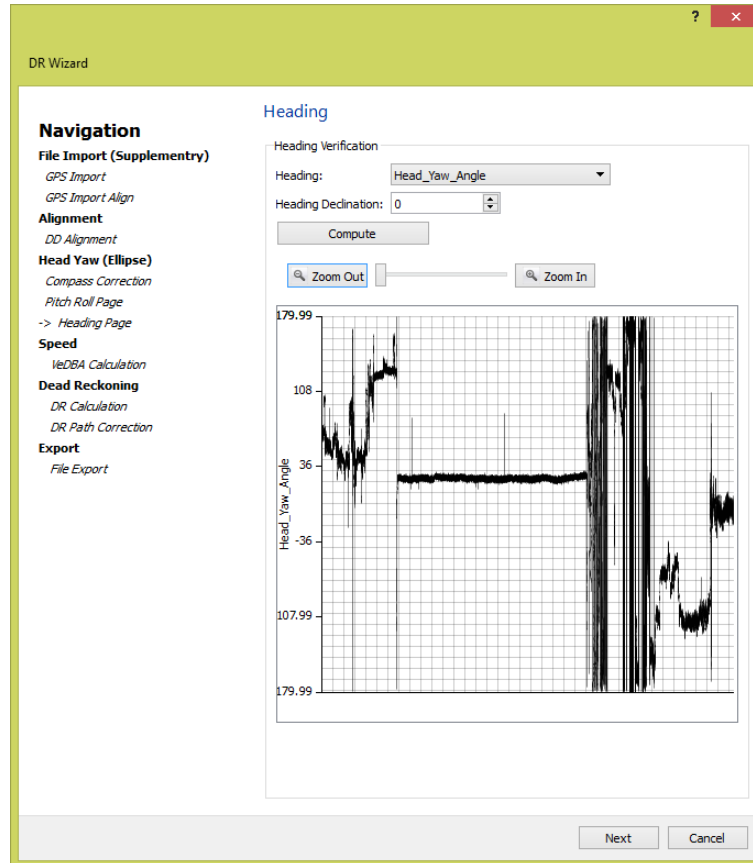


Figure 5.12: The user selects the orientation and error corrected magnetometer components to compute the heading from. A deviation angle corresponding to the heading declination angle can be applied, to transform the heading from magnetic north to geographical north. A graphical preview of the derived heading is shown in time-series graph format for verification.

5.4.2.4 Deriving speed

Dynamic acceleration [WWQ⁺06] is argued as a good measure for predicting speed in terrestrial animals [BSS⁺12] and has indeed been found an effective proxy for speed in 10 disparate species including geese, armadillos, penguins, skunks, ducks, beavers, cormorants and humans, during terrestrial locomotion [BSS⁺12]. However, the vectorial sum of the dynamic acceleration (VeDBA) appears more robust than the overall dynamic body acceleration (ODBA) in this context since it copes better than ODBA to variability in substrate [BQW12]. Framework4 calculates VeDBA using formula 5.11.

$$VeDBA = \sqrt{DA_x^2 + DA_y^2 + DA_z^2} \quad (5.11)$$

where DA_x , DA_y , and DA_z are the dynamic acceleration values derived by taking the absolute values of running means of the raw acceleration values of each of the 3 orthogonal measurement axes from the corresponding raw acceleration values. In this step, the user specifies a window size to use to derive the dynamic acceleration component (see Section 2.2.3.2) (Figure 5.13). Animal speed (s) can then be computed by applying a speed coefficient (m) and adding a constant (c) to the VeDBA value. The speed coefficient and offset can be interactively adjusted in the dead-reckoning wizard page. In addition, we

expose a threshold (t) whereby, if the VeDBA falls below this value, the VeDBA is set to zero (expressed by formula 5.12). This allows the user to remove noise from the resulting speed channel.

$$s = \begin{cases} (VeDBA \cdot m) + c, & \text{if } VeDBA > t. \\ 0, & \text{otherwise.} \end{cases} \quad (5.12)$$

One of the sources of error present when using VeDBA as a proxy for speed is that some behaviours may show the animal as moving when it is in fact stationary [BQW12]. For example, an animal eating (a stationary behaviour) will show energy use through animal jaw movement which will correlate to a speed being registered. Alternatively, running is a non-stationary behaviour. To overcome this, we extend the VeDBA protocol to incorporate behavioural information, such that, VeDBA is only used as a proxy for speed for specific, user-defined, behaviours (an advantage of having TimeClassifier captured in the same software suite). This is expressed by formula 5.13, where $b(i)$ is a function to obtain the behaviour for a given time index (i), and B is the set of all behaviours to utilise for the speed calculation. Further work is required to validate the effectiveness of this approach, although we hypothesise it will provide more accurate speed estimates by removing non-locomotive behaviours from the calculations.

$$s = \begin{cases} (VeDBA \cdot m) + c, & \text{if } VeDBA > t \wedge b(i) \in B. \\ 0, & \text{otherwise.} \end{cases} \quad (5.13)$$

For volant species, the dynamic acceleration is unlikely to correspond to the speed of the animal, invalidating VeDBA in this context. Until a satisfactory measure of speed is derived (such as a pilot tube), we suggest using the constant speed option, and correcting the speed coefficient by applying positional information (like GPS) in the dead reckoning path correction algorithm.

5.4.2.5 Traditional Dead-reckoning

Dead-reckoning combines speed and heading to compute a trajectory. There are several parameters which must be defined prior to the process taking place. These are; (i) an initial start position defining the point where the path starts (if GPS data is given, the initial geographic positions are taken from this), (ii) a number of speed parameters for the VeDBA threshold and speed coefficients (introduced prior) and (iii) a heading offset corresponding to the declination angle. The computed path is shown alongside in a map below the parameters so that the user can interactively adjust settings and see the result (Figure 5.14).

To compute the path, the speed (s) obtained from Section 5.4.2.4 must be converted to radial distance in terms of the radius of the earth R ($6.371 \times 10^6 m$). This is calculated to obtain a speed constant (q), identified in formula 5.14.

$$q = \frac{s}{R} \quad (5.14)$$

The latitude and longitude at time t_i (where t_0 is equal to the initialisation position of the trajectory) can be calculated using the previously converted speed (q) (equation 5.14), and heading (H) (equation 5.10) for a given time index. Formula 5.15 identifies the latitude, and formula 5.16 for the longitude component.

$$Lat_i = \arcsin(\sin Lat_{i-1} \cdot \cos q + \cos Lat_{i-1} \cdot \sin q \cdot \cos H) \quad (5.15)$$

$$Lon_i = Lon_{i-1} + \text{atan2}((\sin H \cdot \sin q \cdot Lat_{i-1}), \cos q - \sin Lat_{i-1} \cdot \sin Lat_i) \quad (5.16)$$

The complete set of latitude and longitude points define the trajectory of the animal movement across the earth's surface. The trajectory columns are appended to the data set using framework4. In future work we wish to also derive vertical movement using the pressure sensor.

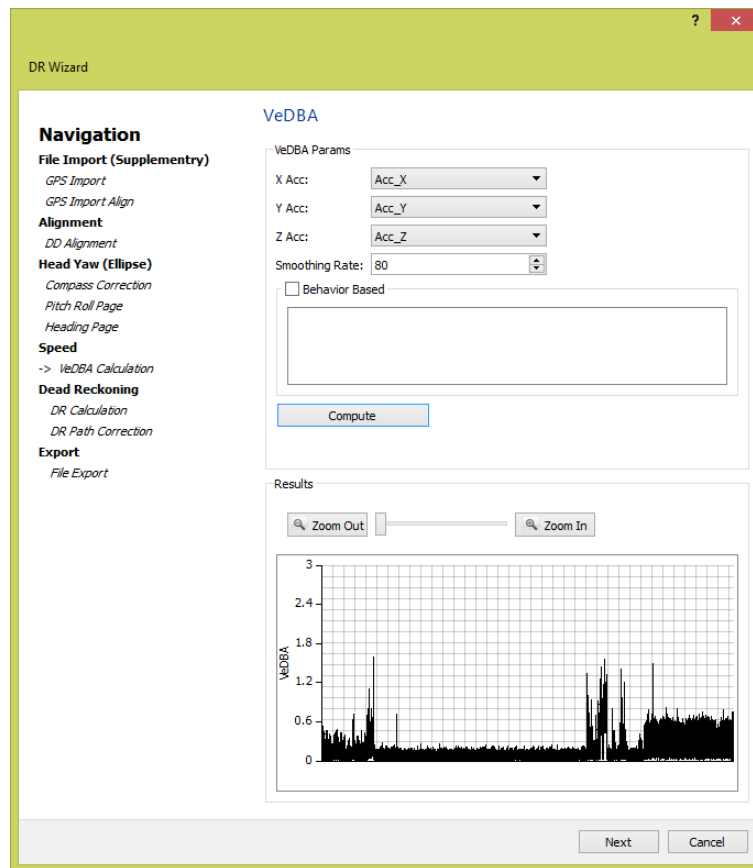


Figure 5.13: VeDBA is computed by selecting the accelerometer components and an appropriate smoothing rate to utilise. The user can select to perform behaviour based VeDBA computation, where VeDBA is only applied to estimate speed for specific behaviours derived in Framework4 (e.g. running). A graphical preview of the derived VeDBA channel is shown below the parameter settings in a time-series graph.

5.4.2.6 Dead-reckoning with position correction

The advantage of a dead-reckoned track is that it can give very fine detail in the route of an animal and do so without reference to external ground-truthing sources, although confidence in the precision of the route will inevitably decrease with increasing time between ground-truthed points [BWJ⁺15]. Nonetheless, the approach has particular value in being able to allude to trajectories where conventional tracking methods do not work (Figure 5.15). However, even small, but systematic, errors in the derived heading and speed channels will accumulate over time and thus can increase the resulting error correspondingly during a deployment.

To overcome these problems, we introduce a novel dead-reckoning correction algorithm which adjusts the heading and speed of the obtained dead-reckoned trajectory using time-aligned positional information from external sources (e.g. GPS). Combining the dead reckoned trajectories with that of supplementary positional fixes resets the accumulated error at each given position (Figure 5.16). The heading and speed is corrected by adjusting the length and orientation of the dead-reckoned path until they align to the same positions along each position fix of the external geo-referenced data. The algorithm we introduce is an iterative approach where-by each iteration forces the dead reckoned path to adhere more tightly to the ground truth fixes as the heading and speeds are adjusted. We allow the user to repeat the adjustment process a set amount of times or continue until each speed and heading adjustment amount is under a specified threshold difference.

We define the following steps of the corrected dead reckoning algorithm before specifying the underlying formulas for each step of the algorithm in turn. Listing 5.1 specifies the algorithm in pseudo-code. For each position in the dead reckoned and externally sourced trajectory (in this case GPS data):

1. the heading differences are computed between the current position and next location

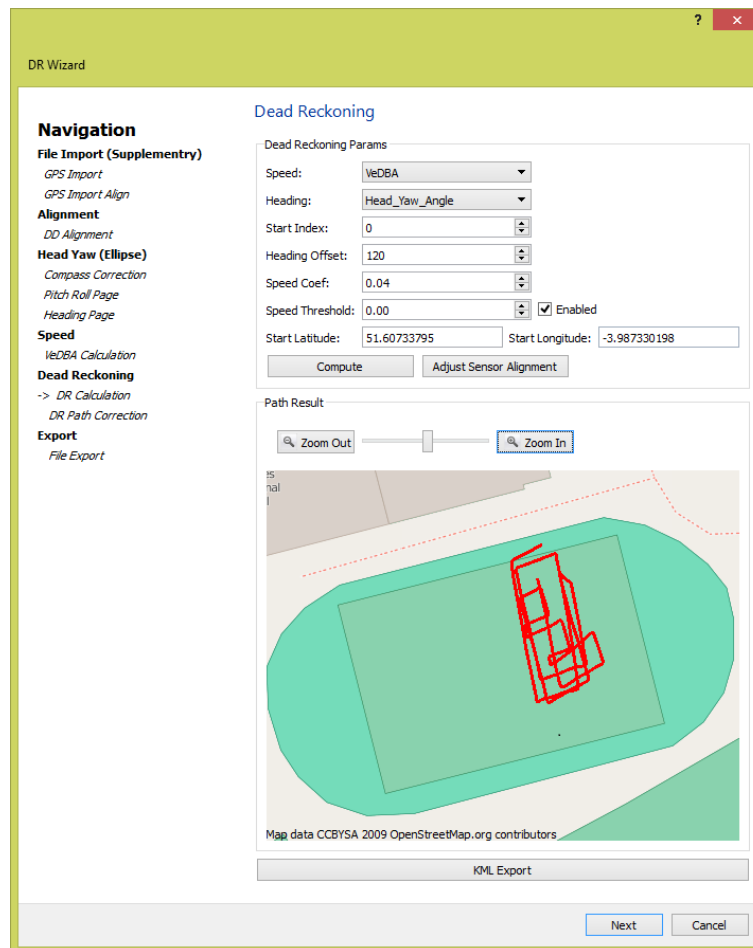


Figure 5.14: The dead reckoning step combines the derived speed and heading channels to generate a trajectory. The user specifies the speed and heading channels to utilise for this procedure. A start index in the data (default zero) is required since usually an initial portion of the data is used for calibration which is not desirable to include in the resulting path. A heading offset allows the user to apply an angle of declination such that the heading corresponds to true north. A speed coefficient and threshold amount adjusts the speed accordingly. The start latitude and longitude specify the initial position of the trajectory. Once compute is pressed the resulting path is shown in map below. The user can interactively refine parameters until the expected result is achieved.

- for the GPS and dead reckoned fix,
2. the difference in distances are computed for the same points on each trajectory,
 3. a heading coefficient is calculated between the GPS and dead reckoned heading differences to determine the offset of the dead reckoned trajectory in regard to that specified by the GPS fixes,
 4. a speed coefficient is computed to determine the differences in speed between the dead reckoned and GPS distances,
 5. finally, the corrected trajectory is determined by adjusting the original dead reckoned track using the derived speed and heading coefficients.

This process is repeated a user defined number of times, or until the heading and distance coefficients computed are less than a specified threshold.

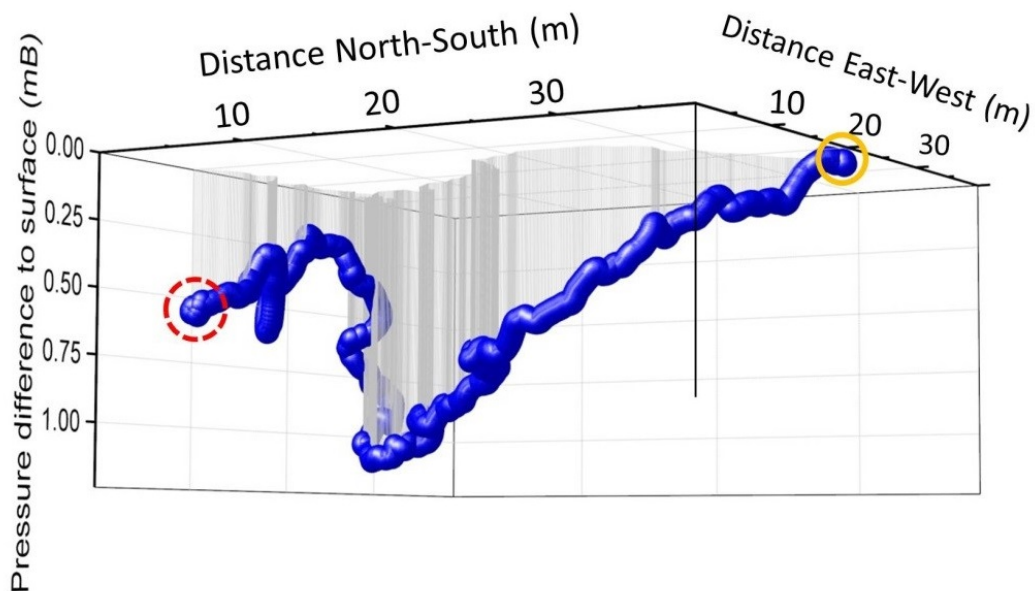


Figure 5.15: Dead-reckoned track of a European badger (*Meles meles*) in Northern Ireland leaving its sleeping quarters (red dashed circle) and moving through the underground sett to emerge at the entrance (yellow circle). The vertical axis representing depth is shown as the pressure difference between the surface and any time underground. The reconstruction assumes that animal speed is directly proportional to VeDBA [BQW12] underground in the same way it is on the surface. If this is not the case, the derived distances will be affected accordingly.

Listing 5.1: Algorithm for computing a dead reckoned corrected trajectory using an external ground truth position source.

```

maximumHeadingCoefficient = 0;
maximumDistanceCoefficient = 0;

do {
  For each position in trajectory (dr_pos, gps_pos) {
    gpsHeadingDiff = headingDifference(gps_pos{i}, gps_pos{i+1});
    drHeadingDiff = headingDifference(dr_pos{i}, dr_pos{i+1});

    gpsDistanceDiff = distanceDifference(gps_pos{i}, gps_pos{i+1});
    drDistanceDiff = distanceDifference(dr_pos{i}, dr_pos{i+1});

    headingCoefficient = gpsHeadingDiff - drHeadingDiff;
    maximumHeadingCoefficient = max(maximumHeadingCoefficient, headingCoefficient)

    speedCoefficient = gpsDistanceDiff - drDistanceDiff;
    maximumDistanceCoefficient = max(maximumDistanceCoefficient, speedCoefficient);

    trajectory{i} = correctedTrajectory(headingCoefficient, speedCoefficient, dr_pos);
  }
} while (maximumHeadingCoefficient < headingThreshold AND
         maximumDistanceCoefficient < distanceThreshold)

```

We now outline the underlying methods for computing heading difference, distance difference, heading coefficient, and a speed coefficient, before binding them together to derive an adjusted dead reckoned trajectory.

Heading difference - The heading (θ) between two points $(Lat_0, Long_0)$ and $(Lat_1, Long_1)$ is calculated as formula 5.17.

$$\theta = \arctan\left(\frac{\cos(Lat_0 + \sin(Lat_1) - \sin(Lat_0) + \cos(Lat_1) \cdot \cos(Long_1 - Long_0))}{\sin(Long_1 - Long_0) + \cos(Lat_1)}\right) \quad (5.17)$$

Distance difference - The distance (d) between two points $(Lat_0, Long_0)$ and $(Lat_1, Long_1)$ is calculated by first computing the latitude distance (equation 5.18) and longitude distance

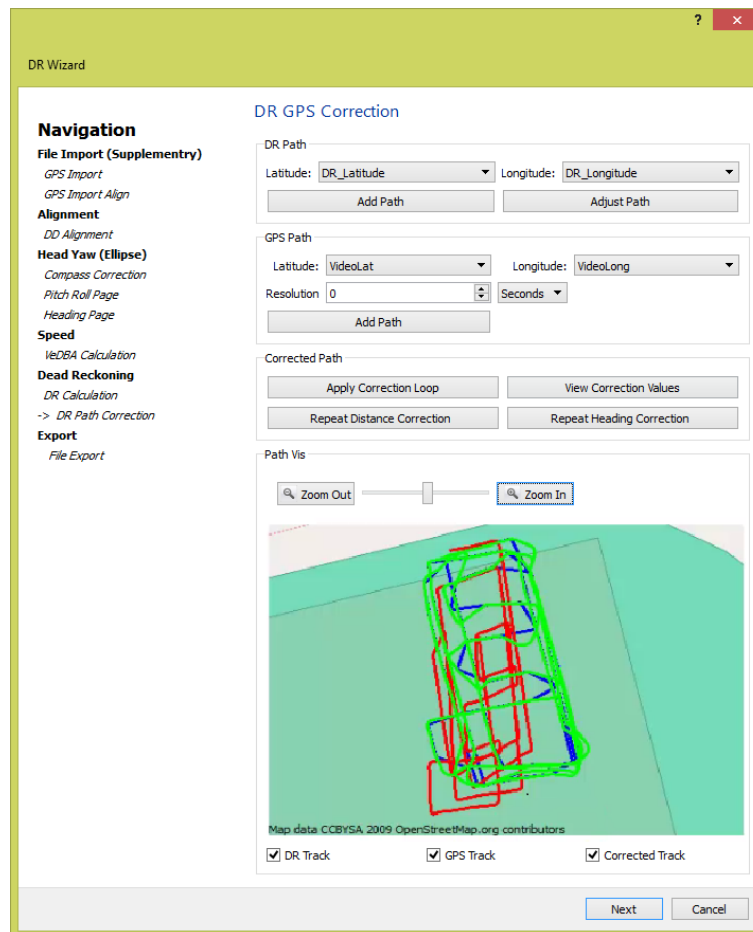


Figure 5.16: The dead reckoning correction stage adjusts the dead reckoned trajectory with positional fixes to reset accumulated error along the path. The user specifies the path to perform correction on, along with the positional data to use as a ground truth to force the dead reckoned path through. The user can down-sample the positional data to apply the correction at set resolutions. The resulting paths are overlaid in a map view. The coloured paths are associated as follows: Blue for the GPS trajectory, red for the dead reckoned trajectory, and green for the dead reckoned corrected trajectory.

(equation 5.19), which are then combined to compute an orthorhombic distance (equation 5.20), before finally being projected onto the surface of the earth for geographical distance (equation 5.21).

$$\text{latitudeDistance}(\text{lat}D) = \text{Lat}_1 - \text{Lat}_0 \quad (5.18)$$

$$\text{longitudeDistance}(\text{long}D) = \text{Long}_1 - \text{Long}_0 \quad (5.19)$$

$$\text{arcDist} = \sin\left(\frac{\text{lat}D}{2}\right) \cdot \sin\left(\frac{\text{lat}D}{2}\right) + \sin\left(\frac{\text{long}D}{2}\right) \cdot \sin\left(\frac{\text{long}D}{2}\right) \cdot \cos(\text{Lat}_0) \cdot \cos(\text{Lat}_1) \quad (5.20)$$

$$d = R \cdot 2 \cdot \arctan\left(\frac{\sqrt{\text{arcDist}}}{\sqrt{1 - \text{arcDist}}}\right) \quad (5.21)$$

Heading coefficient - The heading coefficient is the amount two adjacent GPS and dead reckoned points need to be adjusted by to be of the same orientation. For two sequential ground truth fixes, there are usually many dead-reckoned fixes in-between. Firstly the heading is adjusted (equation 5.22), this consists of adding a constant heading to all the dead-reckoned headings between the ground truth fixes (Figure 5.17 right).

$$\text{headingCoefficient}(\text{h}C) = (\text{gpsHeading} - \text{drHeading}) \quad (5.22)$$

Speed coefficient - A speed coefficient computes the amount to adjust the speed of the dead reckoned trajectory between the fixes such that they align (equation 5.23). This consists of multiplying the original speed values (derived from VeDBA) between the ground truth fixes so the speeds align.

$$\text{speedCoefficient}(sC) = \left(\frac{\text{gpsDistance}}{\text{drDistance}} \right) \quad (5.23)$$

Corrected trajectory - The formulae for dead-reckoning are adjusted to generate a dead-reckoned corrected path for each longitude (equation 5.24) and latitude (equation 5.25) position, where q is the original speed coefficient, and H is the original heading coefficient applied when computing the initial dead reckoned trajectory. The speed coefficient (sC) is multiplied by the original speed coefficient, along with an addition of the heading offset (hC), specific for that section of the track to counter the deviations from the ground truth geographical positions (Figure 5.17).

$$\text{Lon}_i = \text{Lon}_{i-1} + \arctan\left(\frac{\sin(H+hC) \cdot \sin(q \cdot sC) \cdot \cos(\text{Lat}_{i-1})}{\cos(q \cdot sC) - \sin(\text{Lat}_{i-1}) \cdot \sin(\text{Lat}_i)}\right) \quad (5.24)$$

$$\text{Lat}_i = \arcsin(\sin \text{Lat}_{i-1} \cdot \cos(q \cdot sC) + \cos \text{Lat}_{i-1} \cdot \sin q \cdot \cos(H+hC)) \quad (5.25)$$

Once finalised the user clicks finish where the corrected latitude and longitude components are appended to the data set.

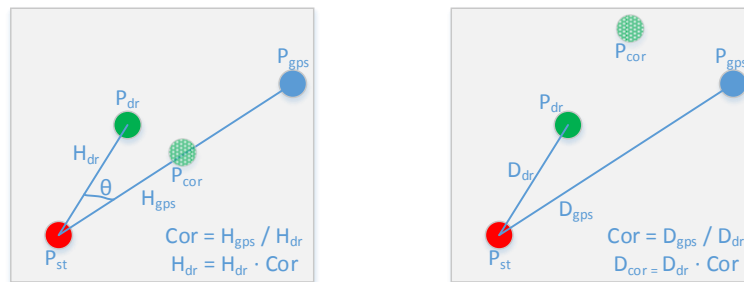


Figure 5.17: Illustration of the dead reckoning correction algorithm. A heading correction factor is first calculated and applied which adjusts the heading between two trajectory fixes obtained using the dead reckoning algorithm to those of a specified ground truth source (left). The distance correction factor is calculated and applied which adjusts the distance between two fixes to those of the specified truth source (right).

5.4.3 Derivation of Animal Behaviour

The derivation of animal behaviour in Framework4 utilises the TimeClassifier framework introduced in Chapter 4. Framework4 was built as an extension to TimeClassifier based on the feedback we obtained from our collaborators to enhance the capabilities of TimeClassifier to derive the additional components advocated by the daily diary concept [WSL08].

By extending TimeClassifier with the ability to derive additional attributes, and coupling them together into one framework, not only can animal behaviour be classified with respect to time, but that the occurrence of different behaviours can be represented on GPS-enabled dead-reckoned animal tracks in an obvious colour scheme (Figure 5.18).

5.4.4 Derivation of Animal Energy Expenditure

Since the suggestion by Wilson et al. [WWQ⁺06] that dynamic body acceleration could be used as a proxy for energy expenditure (measured via VO₂), there have been a number of studies that have confirmed its utility in species ranging from shellfish, through fish, amphibia and reptiles to birds and mammals (see Halsey et al. [HSW11] for review). Two measures have been used, Overall Dynamic Body Acceleration (ODBA) and Vectorial Dynamic Body Acceleration (VeDBA), which are essentially equivalent in terms of their power to predict VO₂ [QCW⁺12] although VeDBA has more utility for predicting speed [BSS⁺12]. Framework 4 uses VeDBA (see section 5.4.2.4 for calculation) as a proxy for VO₂ [HSQ⁺09] although it



Figure 5.18: The dead-reckoned trajectory of a cow (*Bos taurus*) in a field in Northern Ireland over 2 hours, colour-coded according to different activities green = grazing, black = walking, red = lying down

should be noted that dynamic acceleration-derived metrics cannot allude to metabolic costs associated with processes such as specific dynamic action and non-shivering thermogenesis [GWS11] so users should be cognisant of this in considering the limitations of this approach. Previously, this was derived using custom formulas in spreadsheet software.

Plots of animal trajectory can be colour-coded according to VeDBA and thus highlight the link between VeDBA-derived metabolic power and location (Figure 5.19).

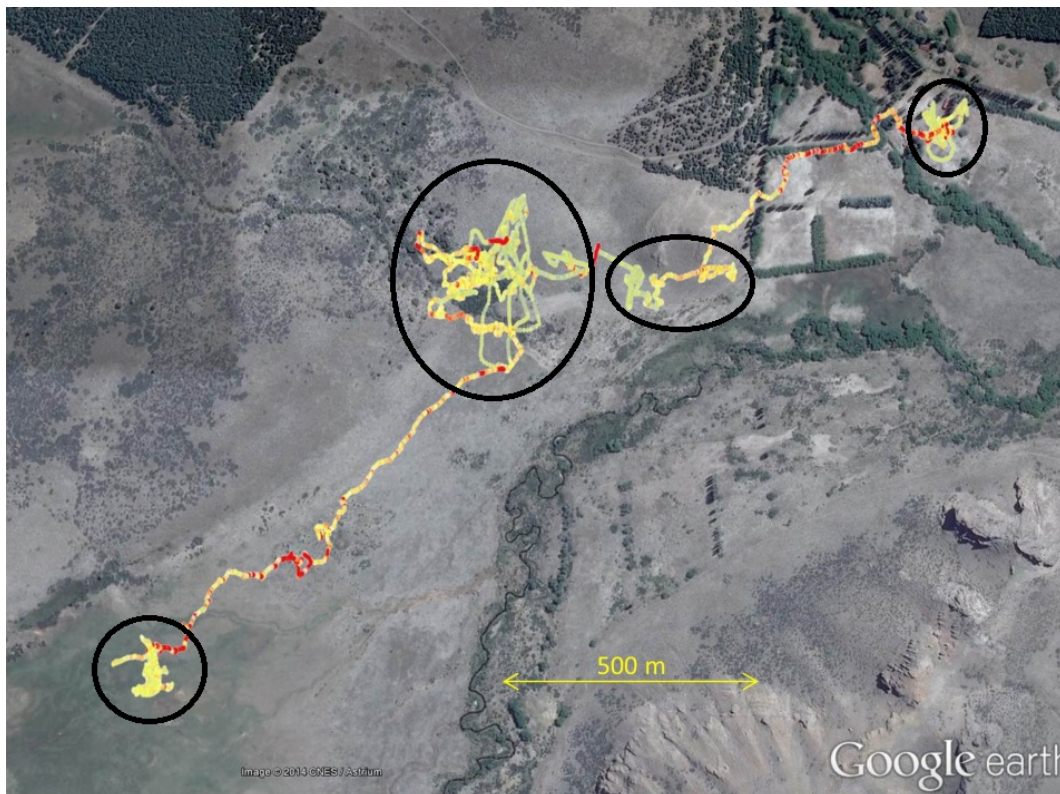


Figure 5.19: The dead-reckoned trajectory of a sheep (*Ovis aries*) in Argentina over 14 hours showing how VeDBA, a metric that correlates linearly with metabolic rate, (ranging from pale green to yellow [low values] through to red [high values]) varies with location and track tortuosity. Note how the twists and turns in the trajectory are generally linked to lower VeDBA (circled in black).

5.4.5 Derivation of the Physical Characteristics of the Environment

Many animals modulate their behaviour in the environment according to its physical characteristics. For instance, reptiles may associate with areas of elevated temperature or isolation to warm up [CC86] while many bird species are limited in their foraging capacities by light [WFHM99]. Thus, the ability to resolve the geographic position of animals in tandem with environmental variables can help explain the incidence or emergence of particular behaviours (e.g. Wilson et al. [WCBL94]). A key part of Framework4 is that it resolves animal space use and behaviour over time, inserting the positional and behavioural data into the original data file in columns. Given that daily diaries also record environmental variables, such as temperature [WSL08], this means that these environmental variables are consequently linked temporally and spatially to the behaviour.

5.4.6 Exporting data

The derived analytical attributes from the software can be outputted together into one data file. Exporting data is supported via navigation to the *'Export'* option in the *'File'* menu on the main tool bar. The resulting exported file is formatted to the CSV standard, where all the derived attributes from framework4 are appended as additional columns in the data file alongside the existing data channels. Each behaviour is assigned a unique numerical value where, if a data item falls within a labelled region, it is assigned this value.

5.5 Evaluation

In this section, we report an evaluation of the Framework4 software suite. Firstly we evaluate and validate the corrected dead reckoning algorithm. We then present two real-world case studies of recent publications which utilise Framework4 to uncover new knowledge about how humans interact and adapt with their surroundings. Finally, since Framework4 is publicly available, we report statistics on its usage in the research community.

5.5.1 Software System

Since we introduce behaviour and dead reckoning algorithms in Framework4 we evaluate their usage. Firstly, we provide an overview of our findings for the labelling aspect (TimeClassifier) from Chapter 4. We then validate the dead reckoning component performing a video synchronised experiment. The derivation of energy expenditure, and physical characteristics of the environment use well established methods [WSL08, BWJ⁺15], we therefore do not seek to validate these methods in this study.

5.5.1.1 Behaviour Derivation

In Chapter 4, we introduce TimeClassifier for labelling animal behaviour. Framework4 utilises the TimeClassifier methods, we therefore summarise our findings and refer the reader to Chapter 4 for the underlying validation methods. We present two case studies with domain experts applied to data obtained from turtles, and condors to determine specific behaviour events in the data. Using the software feedback loop, the experts were able to achieve 99.86% and 100% accuracy, respectively, in a short space of time. A direct comparison with machine learning techniques (hierarchical clustering, KNN, SVM, and random forests) is also presented using purely the algorithmic component of the software. TimeClassifier achieves a faster run time and accuracy in comparison to these methods. A visual analytics user-in-the-loop approach (as shown in the case studies) further lets accuracy be boosted to 100% using domain feedback through the softwares user interface, something which was previously not possible. Using the TimeClassifier methods allows Framework4 to achieve unprecedented levels of accuracy for biologists when operating on their smart sensor data.

5.5.1.2 Dead Reckoning

The novelty of Framework4 is the dead reckoning techniques we introduce, combining traditional dead reckoning with ground truth data to reset accumulated errors arising. In order to validate the dead-reckoning technique, we performed a simple experiment at Swansea University, UK. A human participant was equipped with a Samsung Galaxy S5

phone, set to record GPS position every second and accelerometer and magnetometer data at 30 Hz. In addition, the participants position was logged simultaneously using a video camera. Geo-referenced points were obtained by performing a perspective transformation to acquire a top-down view. In this instance, we were unable to perform the study using animal data where it would not have been possible to obtain the video synchronised data we required for the validation. We manually labelled the position of the participant on each frame of the video, and then transformed the points using known reference positions to latitudinal and longitudinal coordinates. The participant walked a grid pattern within a 40 by 20 meter area on a grass playing field, in front of the cameras field of view. The GPS data obtained was not sufficiently accurate in this instance to be used for ground-truthing the dead-reckoned track (see Figure 5.20 (b)). The GPS data acquired should have been good, the experiment took place in a sports field with good sight of the sky, not occluded with trees or in an urban environment. We also waited some time for GPS to lock. The results of this experiment are indicative of GPS precision and rings true to what researchers achieve in the field. There are more precise positional systems (e.g. differential GPS), however, usually these have sensors too large for animal attachment or require animals to be limited in geography (i.e. within a specific range from a transponder).

The video positions were sub-sampled down to a position every 1, 2, 5, 10, 15 and 20 seconds and the dead-reckoning procedure was performed using each of these data sets for ground-truthing. The distance between the dead-reckoned positions during the latent periods (i.e. the periods between ground-truth correction) and the concordant video-derived positions (the complete, continuous data without sub sampling) was measured and termed the distance error. Figure 5.20 (c-f) shows a comparison of the tracks dependent on the frequency of ground-truthing. GPS alone did not accurately reproduce the true pattern of movement at this scale.

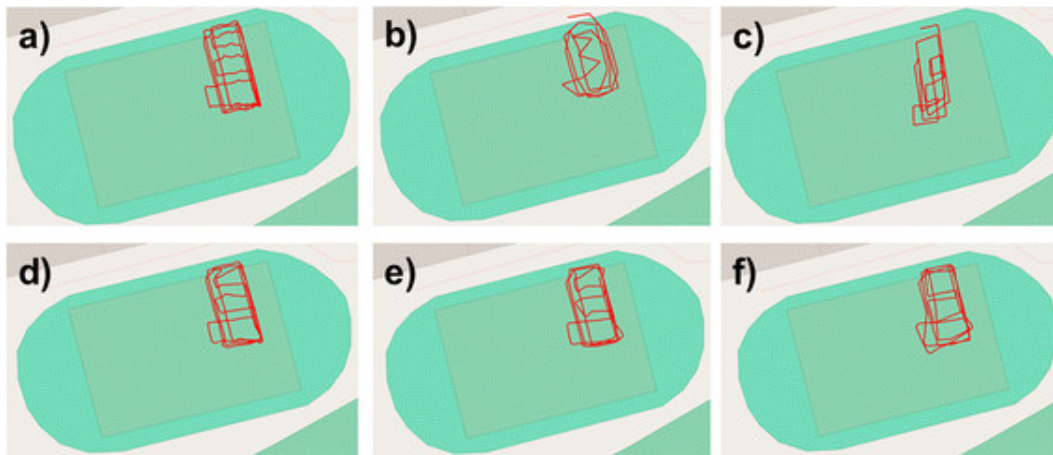


Figure 5.20: Two-dimensional paths of the human participant as determined by (a) video Recording, (b) GPS, (c) dead-reckoning without correction, (d) dead-reckoning with correction every 2 seconds, (e) dead-reckoning with correction every 5 seconds, (f) dead-reckoning with correction every 10 seconds.

Figure 5.21 shows the mean distance error over the entire experiment at each of the ground-truthing frequencies. As expected, mean distance error increased with longer ground-truth intervals. Using a linear regression, we estimate the error accumulation rate to be 0.194 m per second between ground-truth corrections. At this stage, this figure is merely advisory as the error accumulation of terrestrial dead-reckoning is likely to be highly variable according to animal behaviour, surface type and track tortuosity [WSL08, BQW12]. Further controlled trials are required to explore this issue. Bidder et al. [BWJ⁺15] introduce an error taxonomy model for the application and effectiveness of different resolutions on the position correction algorithm, it is beyond the scope of this chapter to investigate the application of parameter settings for dead reckoning for animal deployment in the wild.

5.5.2 Impact of Framework4

The utility of Framework4 has advanced researchers to derive and construct new insights into smart sensor data, including in new research areas. We now report two impact case studies

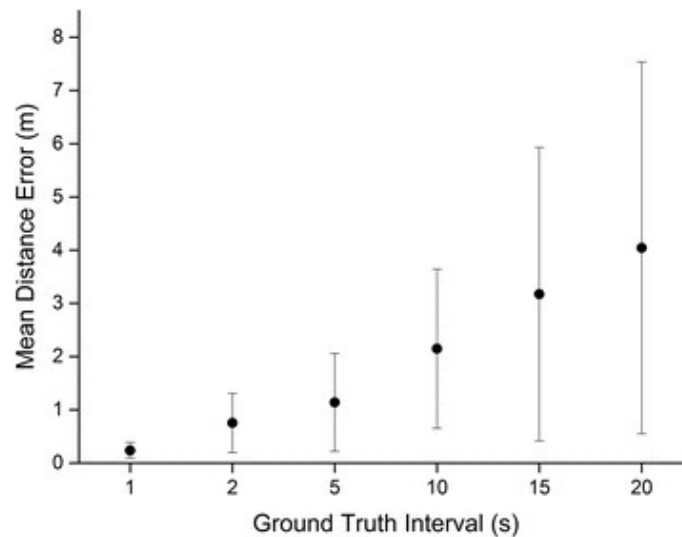


Figure 5.21: Mean distance error (meters) at each of the ground-truthing regimes. Error bars represent standard deviation of the mean.

of Framework4, which have resulted in contributions to international research journals in the movement ecology research community. Firstly, we introduce a new device supported by framework4 [WHW⁺15], and secondly insights framework4 have made possible into the data obtained [WNW⁺15].

5.5.2.1 A System for Identifying how humans assess their surroundings

The study of how humans move and behave in environments has led to a profusion of research efforts to identify the visual attention given to objects [ACMR00]. Conventional analysis has relied on eye tracking methods, where a camera is used to determine the orientation of the eye, and metrics, such as, fixations and scanpaths [Coo05] which detail statistical summaries. What we see is however determined by three things; body orientation, head attitude and eye attitude within the head [ACMR00]. The eyes can only track objects if the head is orientated such that the objects are within the field of view. Therefore, head attitude and movement play a core role in defining the environment. However, this avenue has not previously been studied due to operational and technological constraints.

The Framework4 software and methods were applied to creating a new device, called, the Human Interfaced Personal Observation Platform (HIPOP), which is a head mounted unit that logs magnetometer and accelerometer data combined with GPS data channels. The premise being that the head mounted unit can be used to derive heading and pitch of the wearer, for gaining an understanding of how humans assess the environment they are in through studying head altitude and movement channels.

The device features two daily diary devices, one attached to the back of the individual to obtain movement trajectories, and a second head mounted (embedded in a cap) to compute heading, along with head pitch (Figure 5.22). Using the movement trajectories, along with head angle, and head pitch, an estimate can be obtained of how the head is orientated in relation to gravity and geographic north, and to geographic position, which allows us to understand how the environment is being 'framed' by the wearer in regards to the environment. The HIPOP has particular uses in the psychology domain for understanding how people react to the environment they are in.

Framework4 bridges the gap between the data logged by the HIPOP and researchers utilising the device to identify new insights into human behaviour. The dead reckoning wizard allows the user to derive head pitch, heading, and a movement trajectory for the individual. The behaviour component allows the user to segment the data appropriately by behaviour, or study, which can be exported in a merged data format, or individually to allow the identification of summary statistics and comparison between individuals.

The device was deployed on individuals visiting a botanical garden. The results were analysed using Framework4 to derive the necessary dimensions using built for purposes visualisation software outside of Framework4. The interplay of head pitch and head heading show considerable variance along the paths taken by each of the participants at the

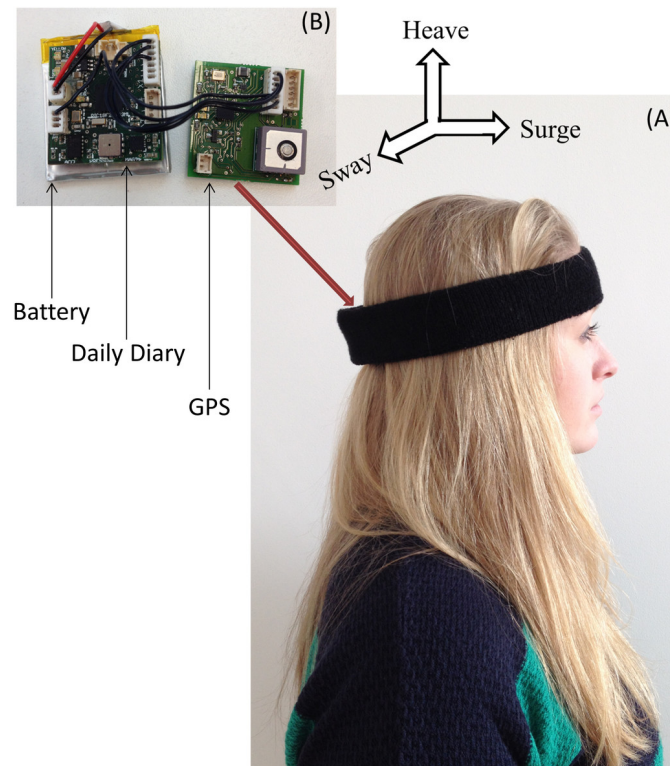


Figure 5.22: The HIPOP head mounted unit consists of a daily diary and GPS tag embedded into a cap to compute heading, head pitch, and movement trajectories. This allows an estimate of how the head is oriented to be quantified.

botanical gardens, and highlight the role that head orientation might be expected to play in exposing people to certain features of an environment. The application of Framework4 allows researchers to generate data dimensions previously inconceivable.

5.5.2.2 Rules

Since the premise of the HIPOP platform identifying correlations between head orientation and how the environment is framed in relation to environmental content, Framework4 was next challenged to identifying rules behind how head attitude is modulated by the environment. The behaviour recognition (template matching) aspect of the software was applied to the derived head pitch dimensions to identify and quantify distinct behaviours occurring when people moved through a botanical garden, a reef, and a featureless corridor.

The study identified three stylised behaviours; smooth, regular peaks, and troughs in head pitch over variable length fixations. Template matching quantified the variability of head pitch patterns, which accounted for 40% of the total time during deployment. Figure 5.23 illustrates the incidence of stylised peaks extracted using the behaviour labelling component of Framework4. The combination of Framework4 with the HIPOP device has identified that humans continually scan their environment which is influenced by the type of environment, specifically concentrating around the distribution of points of interest.

5.5.3 Usage and Statistics

The Framework4 software suite is freely available for download from the web address <http://www.framework4.co.uk>. Figure 5.24 captures the main page of the website, which users are greeted with when first entering. The website is sectioned into 4 pages providing; an overview of the features, videos which provide user tutorials on how to use the software, a download page, and a portal to contact us. An installation wizard is bound to the downloadable software which automatically creates desktop and start menu short-cuts for ease of use. We advise users only apply Framework4 to Microsoft Windows 7 or newer where it has been tested. We encourage its free use, with no permission or license required. We do however ask users to cite the paper this chapter is constructed from in resulting

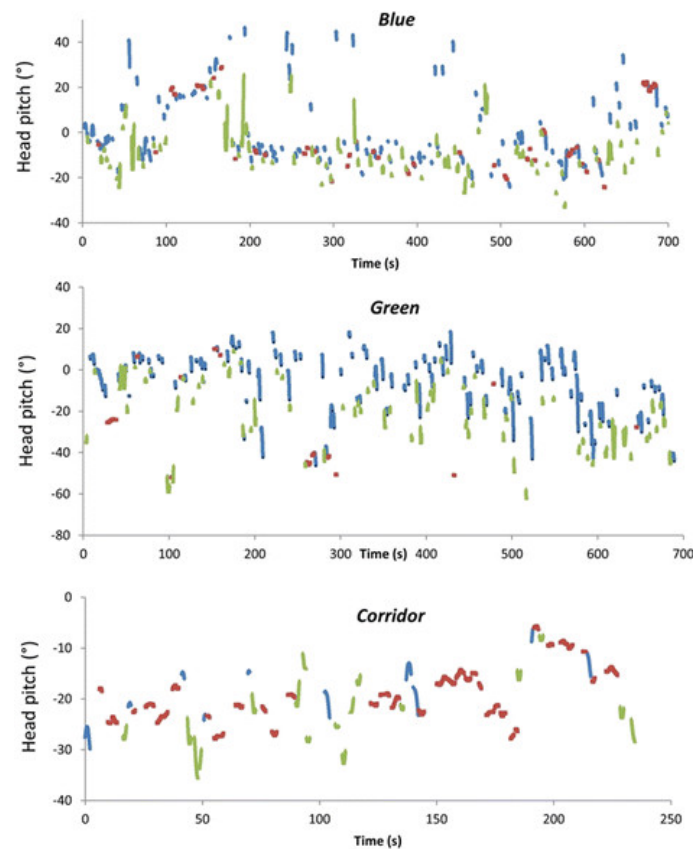


Figure 5.23: Incidence of stylised peaks (blue symbols), troughs (green symbols) and fixations (red symbols) in head pitch over time in three participants operating in three different environments: a green space (botanical garden), a blue space (snorkelling in an underwater environment) and finally a feature-less corridor. Sections of the graphs where there no data are shown correspond to periods where the changes in head pitch did not correspond to the identified stylised behaviour. The green space featured participants mostly orienting their head in a downward position as they investigated the surround environment and plants offered by the garden. The blue space features a large amplitude difference in head orientation (80 degrees), presumably as there were interesting features of the environment surrounding the individual. The corridor had the smallest amplitude difference. The corridor environment was extremely depauperate in visual (and other) stimuli while the blue and green environments were both rich in stimuli.

publications [WJL⁺15b]. Following the publication of this paper we are able to report 10 citations of the publication at the time of writing.

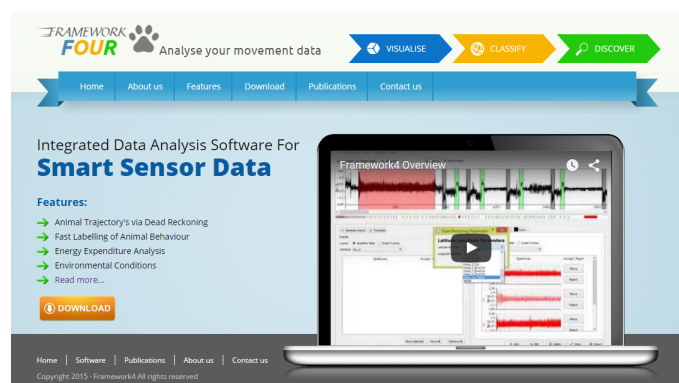


Figure 5.24: The home page of the Framework4 website. Users are presented with an textual summary of the four core attributes Framework4 allows users to derive. We also feature a video illustrating a one minute summary of the core features.

When the user downloads a copy of the software we request information about the user wishing to obtain the software. We capture the following fields; name, institution, geographical location, and email address. Using this data, we are able to report demographics on the usage of Framework4. We report over 60 users across 12 countries. Figure 5.25 illustrates the distribution of users across the world. We expect the usage of Framework4 to increase as people become more aware of its existence and its application becomes apparent across the literature.



Figure 5.25: A visualisation of the usage of Framework4. We report over 60 users across 12 countries.

5.6 Summary

This chapter introduces Framework4, a single program that explicitly links space use, movement, behaviour, and energy expenditure in free-living animals together with environmental conditions, doing so using an accessible column-separated format for ASCII-type data. The software makes available previously inaccessible methods for deriving these attributes through purpose built user interfaces and wizards, this allows the users of smart tags to calculate all four key attributes advocated by the daily diary and then to bind them together into one single output file. The result of this is a methodology which enhances the understanding of the processes that affect the way that animals and humans move within their environment.

We combine fine movement and gross movement at the same time, through a corrected dead reckoning algorithm which corrects cumulative errors from the estimates of derived heading and speed using conventional dead reckoning. We validate the application of this method through a field study, although simplistic, the study identifies the fine-scale inaccuracies with GPS data, which rings true of what researchers achieve in the field. Conversely, dead reckoning is powerful for recording fine-scale movement, but errors accumulate leading to large displacements over time. Combining dead reckoning with a large-scale positioning system (e.g. GPS) resets the cumulative errors allowing both fine-scale and gross movement to be derived in tandem. The corrected dead reckoning technologies we introduce does not consider any errors associated with the external position source used. For example, GPS consists of its own class of errors (accuracy to within appropriately 5 metres [BWJ⁺15]). It is therefore beneficial in future work to explore the effect of other logging techniques for position (e.g. video, differential GPS, and RFID) along with incorporating external sources of error into the resulting trajectory correction algorithm.

The new capacity this software brings opens new research opportunities by allowing the derivation of these attributes. The utility of Framework4 has advanced the research of human behaviour through the realisation of a new device for understanding how humans move and behave in their environment, and secondly the derivation of associated rules which identify how humans behave in specific environments. We record the demographical usage of Framework4 and report over 60 users, across 12 countries actively utilising the software to advance their research. The nature of Framework4 being a tool to enable biologists to determine the data channels they require for their analysis means it's essential for the software to keep up to date with the state-of-the-art methods used in the biology community. It is therefore necessary for Framework4 to develop an open API in future work from which

researchers can plug-in their research methods to make them available to the community within one unified data analysis platform.

6. VISUALISATION FOR LINKING BEHAVIOUR AND ENERGETICS FOR DIAGNOSTICS

“You can use all the quantitative data you can get, but you still have to distrust it and use your own intelligence and judgment.”

– Alvin Toffler, *American writer and futurist*

Contents

6.1	Introduction	116
6.2	Background	118
6.3	Online Methods	118
6.3.1	Data Dimensions	119
6.3.2	Raw Plot	120
6.3.3	Spherical Plot	120
6.3.4	Binning in Three-Dimensions	121
6.3.5	Spherical Histogram	123
6.3.6	G-Urchin	123
6.3.7	Differential G-Sphere	125
6.4	User interface	127
6.4.1	Data G-Sphere Construction	127
6.4.2	Operation G-Sphere Construction	130
6.5	Results	130
6.5.1	The Basic G-Sphere	130
6.5.2	The Spherical Histogram Plot	131
6.5.3	The G-Urchin	132
6.5.4	The Differential G-Urchin	133
6.5.5	Contrasting Patterns of Acceleration of Different Body Parts	133
6.6	Discussion	133
6.7	Summary	137

In the previous chapter, we introduced Framework4 a software tool which facilitates the derivation of behaviour and energy (among other attributes) into one unified file format. Through our feedback driven development it was suggested biologists would like to work on and compare channels recorded and derived from the smart sensor tags in tandem. In this chapter, we specifically explore the combination of power use and behaviour. We introduce a proof-of-concept visualisation to facilitate this which we seek to validate in further work. To date, aside from some studies that have worked to define general activity-specific metabolic rate using accelerometry [GHWF09], the community has not sought to bind energy use together with behaviour within one framework (because of previous technological constraints, namely, software). It is the hypothesis of this work that coupling energy-based metrics together with behavioural indices has the potential to enhance the identification of both processes and highlight the relevance of both with respect to the other.

Voluntary animal movement uses energy, so quantifying the allocation of chemical energy for mechanical output and how this relates to movement is important in understanding behaviour, lifestyle and exercise physiology. The energy-movement field is well studied e.g. Bleich et al. [BKW11], but it has been polarised into essentially two main branches

This chapter is based on the publication ‘A spherical-plot solution to linking acceleration metrics with animal performance, state, behaviour and lifestyle’ [WHW⁺16].

Note: The raw plot, spherical plot, spherical histogram and sphere tessellation methods were introduced in a previous publication by the author [WGJL12] for a previous degree. We report these methods for completeness, however, the main contribution of this chapter is the g-urchin and differential g-sphere methods, along with the associated results which tie together behaviour and energetics.

defined by differing methodologies. One branch examines power use [SMW⁺14], which typically requires measurement across extended periods [RJM⁺04, AHWS06, TLMP11] but is limited by the difficulties in attributing instantaneous power to performance [BS90]. The other seeks to quantify behaviour, relying variously on approaches such as high-speed cameras [MTR⁺13], point light displays [NWH⁺12] and force plates [GMS⁺13] for work on humans and, primarily, on observation-based methodologies for wild animals [WT13].

Animal movement, operates over wide space- and time-scales. It encompasses everything from a single limb motion describing a simple arc lasting less than a second, through co-ordination of repetitive limb motion in a whole animal during travel, which may last hours, to the diversity in the complex movement describing the various behaviours exhibited over the lifetime of an animal. Energy expenditure is a major element of optimisation studies that seek to define best strategies, which have a broad remit ranging from examining most enhanced performance by elite athletes [JSOF11] to animals adopting behaviours that maximise survival value [Ham70].

The advent of smart sensor technologies has enabled both the power use and the behaviour of both humans [YH10] and animals [BKW⁺14] to be acquired in tandem using accelerometers in animal/human-attached tags because both attitude and motion can be resolved using these transducers [SWQ⁺08]. In the energetics branch, workers have derived indices, such as those based on dynamic body acceleration (DBA) metrics [WWQ⁺06], that correlate tightly with oxygen consumption [HSW11] while behavioural studies have used various methods such as random forests, vector machines and artificial neural networks to identify behaviours [NSFR⁺12].

The movement and energy communities recognise the problem inherent in the complexity of acceleration data. These provide most value when recorded at high rates (typically more than 32 times a second) across each of the three axes defining orientation in space, producing effectively six channels of data, three relating to the gravity-based component of the acceleration and three relating to the animal-based movement [SWH⁺08]. It is this complexity that still represents an appreciable challenge for the animal (and human) behaviour community in binding energy use and behaviour within one framework [ELVK⁺12].

In this chapter, we introduce a proof-of-concept visualisation named the 'g-sphere' for animal/human-attached acceleration data that allows power use for movement over a temporal scale to be married with behavioural data into one visual for insight. The approach recognises that, because the earth's gravity is constant, a tri-axial plot of tri-axial, orthogonally placed, acceleration data fundamentally builds a sphere, a 'g-sphere' [GJL⁺09b]. This highlights patterns that are not intuitively obvious and inaccessible through traditional time-series visualisation. G-sphere visualisations highlight changes in movement patterns associated with, for example, human emotional state, injury and best practice in single sports manoeuvres but extend through to highlighting energy-based behavioural ecology in wild animals over time periods ranging from seconds to years.

This chapter consists of the following contributions:

- The binding together of energetics and behaviour which highlights patterns which can be used to inform our understanding and identification of processes.
- We show a work-in-progress visualisation of how both behaviour and power use can be linked together by visualising tri-axial acceleration data in 3-D plots (g-spheres) which capitalise on a suite of different acceleration-derived metrics and allow multiple dimensions to be visualised simultaneously. We implement this method in framework4, our publicly available software to make the methods we present accessible to the wider research community.
- We identify areas which we believe can benefit from the value of this approach; sports practice, disease and wild animal ecology. We outline the initial insights which can be made using the g-sphere visualisation for further validation in future work.

The rest of this chapter is organised as follows. In section 2 we discuss background to this research area. In section 3, we introduce the methods behind the g-sphere framework. In section 4, we provide an overview of the user interface which facilitates the g-sphere protocol. In section 5, we present results on real-world ecological data and the insights made using this new approach. Finally, in section 6 we summarise our findings.

6.2 Background

Traditional analysis of smart sensor data typically relies on the use of multiple time-series graphs. Often one is used for each component of the data recorded from a tri-axial accelerometer sensor along with other supporting attributes to derive and understand the way animals operate. So far in this thesis we have focused on enhancing the use of time-series graphs for data exploration, analysis, and presentation of animal movement data. However, for complex behaviours, which span across multiple dimensions it can be difficult to derive and interpret the underlying activity since the signal components need to be cognitively reconstructed in three-dimensions to understand the movement taking place [GJL⁺09a].

Spherical scatter plots are one solution which integrate the tri-axial accelerometer data channels into one three-dimensional visualisation, visually reconstructing each vector in the data series [GJL⁺09a]. To achieve this, the temporal component of the data series is discarded, instead, each acceleration vector is plotted as an offset from the origin to produce a three-dimensional scatter plot of the acceleration vectors. In effect, this off-loads the task of reconstructing the signal (and therefore movement or orientation) in three-dimensions explicitly from the user, by representing it implicitly in the visualisation to illustrate the geometric distribution of the data. However, in doing so any temporal information is lost. Colour can be used, although it is not easy to discern when the time sequence is long. Therefore, spherical scatter plots may only be used to complement data analysis, and not a replacement for time-series graphs.

Figure 6.1 illustrates the visualisation technique on animal posture data from a penguin. A penguin is plotted in the centre of a sphere for reference when playing back through the data series. The recording device was placed parallel to the spine of the penguin during deployment. The recorded vector therefore protrudes from the central spine of the animal directly upwards and onto the surface of the sphere. The resulting trajectory on the sphere therefore indicates the orientation of the animal throughout the time-span visualised (in this case a duration of 15 minutes). In this instance, it is possible to see that the penguin is rotating through a variety of orientations. Since the animal is lying horizontal it is likely that the penguin is swimming (which can be clarified by examining the associated dynamic acceleration signal). As the penguin ascends, descends, and adjusts trajectory the orientation changes, this is reflected in the mapping produced on the surface of the sphere.

The acceleration channel can be decomposed into static and dynamic components. The static acceleration is defined by the constant force of gravity, and therefore can be used to resolve orientation. Dynamic acceleration corresponds to the force of movement upon the sensor and therefore energy expenditure in three-dimensional space. The traditional spherical plots can only operate on one of these channels at once. In this chapter, we build on and extend the spherical scatter plot approach to combine movement data with power usage into one visual (therefore visualising six data attributes in tandem). We call the resulting visualisation the g-sphere, since its capacity for fully visualising the accelerometer (g) data. In the following section, we detail the methods for the spherical scatter plot technique, before identifying the g-urchin and differential g-sphere extensions to this approach.

6.3 Online Methods

The g-sphere visualisation technique has been incorporated into Framework4, our publicly available smart sensor analysis software, which operates on inter alia accelerometer and magnetometer data to determine key elements considered pivotal for movement analysis from smart sensor tags including trajectory, behaviour, energy expenditure, and environmental conditions (Chapter 5).

The basic g-sphere is derived from animal-attached, tag-based, tri-axial acceleration data, where the sensors have orthogonal placement, aligning with the major axes of the body. For the g-sphere, we build on the current state of the art method for visualising accelerometer data by Grundy et al. [GJL⁺09b], using spherical coordinate plots of the data to depict the distributions of the data, to incorporate power usage, and inferences about behaviour. To overcome the problems associated with large datasets (Section 1.3), we utilise frequency-based approaches which show an overview of the data. Firstly, a spherical histogram shows the number of data items in each facet of the spherical coordinate system. Secondly, a 'g-urchin' plot (name derived from its spiky appearance like a sea urchin), which allows the user to further inspect this information married with additional attributes for each

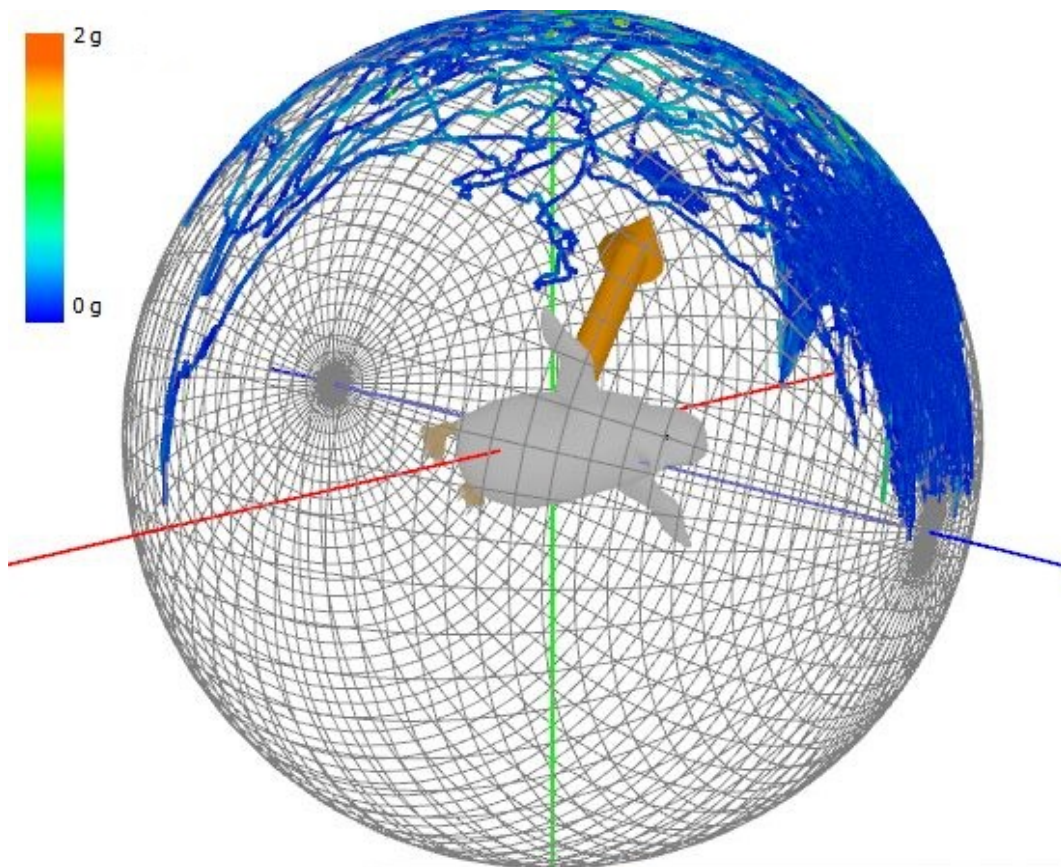


Figure 6.1: A spherical scatter plot visualisation of static acceleration of data acquired from a penguin spanning approximately 15 minutes of activity. The trajectory (mapped to the surface of the sphere) illustrates the orientation of the penguin over this duration. The trajectory is colour encoded to the y axis of acceleration. A model of a penguin is placed in the centre of the sphere, with the orientation vector pointing outward of the spine for reference. The visualisation shows the penguin rotating through a variety of orientations while swimming.

facet, such as power usage. Multiple g-urchins can be combined via addition and difference operations to analyse across instances, behaviours groupings, or data sets. The novelty of this work lies in the coupling of energetics with behaviour and the inclusion of a visualisation method for it based on the g-sphere and urchin plot. The methods by which each of these features operates are as follows;

6.3.1 Data Dimensions

The prerequisites of the g-sphere visualisation is to have static and dynamic acceleration, along with a power metric channel present in the data (derivable from raw acceleration data using Framework4) - seven dimensions total. Measured acceleration is the product of both a static component due to the constant force of gravity, which is manifest in accelerometers according to their orientation with respect to the Earth, and a dynamic component, which equates to the forces exerted on the sensor due to three-dimensional movement of the animal carrying the sensors. Separating these components from the raw accelerometer measurements allows us to isolate postural states and movement (Section 5.3.2.4). Dynamic acceleration-based metrics [WWQ⁺06] have been argued as a predictor of power [HSW11]. Two measures, Overall Dynamic Body Acceleration (ODBA) and Vectorial Dynamic Body Acceleration (VeDBA), have been used, which are essentially equivalent in terms of their power to predict VO₂ (the measure of the volume of oxygen used to convert energy from food) [QCW⁺12].

Framework4 facilitates the user in obtaining these data channels (Chapter 5) where the user can easily derive them using built-for-purpose dialogs and wizards. We have kept the g-sphere method open to using any method for obtaining these, including those in the future,

as additional channels can be computed outside the software and imported into Framework4 (such as, gyroscopic data which provides more accurate orientation data, but currently has a larger energy footprint).

6.3.2 Raw Plot

The first iteration of the g-sphere plots the smoothed data in a three dimensional scatter plot with the heave axis being allocated the y axis, the surge the x axis and the sway the z axis (Figure 6.2). Each vector is considered as an offset from the origin, directly plotted in three-dimensional space as a scatter plot. The colour and size of each data point can be linked to any associated attribute in the data set. A line may be added which connects each data point together to reintroduce the temporal aspect of the data into the visualisation. This representation shows short-lived behaviours well, providing a compelling visualisation of when forces exceed that exerted by gravity. Although when faced with larger data visualisation in three-dimensional makes it difficult to interpret and identify the underlying trends and patterns present.

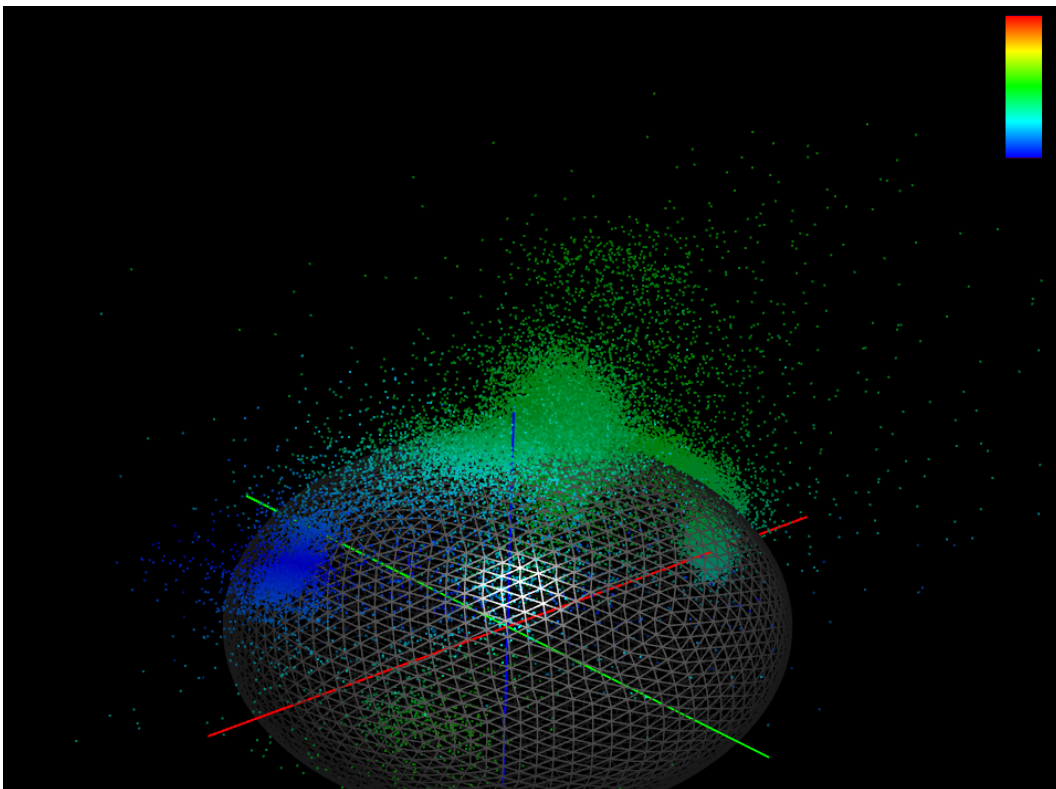


Figure 6.2: A 3-d scatter plot of static (orthogonal) tri-axial acceleration data recorded from an Imperial Cormorant. The time-span visualised contains diving, sitting on the water surface and flying behaviours. Visualising the raw values in three-dimensional makes it difficult to interpret and identify the underlying trends and patterns present.

6.3.3 Spherical Plot

Normalising the vector encodes the directional component (removing magnitude), relevant when analysing posture, and movement direction. Given the x , y , and z channels of the vector, we compute the length of the vector L and normalize the components to x' , y' and z' which are then subsequently plotted in three-dimensional space:

$$L = \sqrt{x^2 + y^2 + z^2}$$

$$x' = \frac{x}{L} \quad y' = \frac{y}{L} \quad z' = \frac{z}{L}$$

Normalising the vectors projects them onto the surface of a sphere in 3-d scatter plots which gives an implicit conversion to spherical co-ordinates (r, θ, ϕ) [AMST11], where

theta (θ) corresponds to the angle of inclination, phi (ϕ) is the angle of rotation on a two-dimensional plane, and the radius is constant ($r = 1$) throughout. Each vector is plotted as a point in the display (Figure 6.3 (a)) and the size and radius of each point can be adjusted by a fixed amount, to link it to an attribute in the data set. Each point can be joined together in chronological order to show the temporal ordering of the vectors as a path in the three-dimensional space (Figure 6.3 (b)) so that the spherical scatter plot shows an intuitive summary of the geometric distribution of posture and direction. By linking the radius (r) of each coordinate to another attribute allows us to encode additional dimensions, such as depth, which, in this case, provides a compelling illustration of diving patterns along with the associated state (Figure 6.3 (c)).

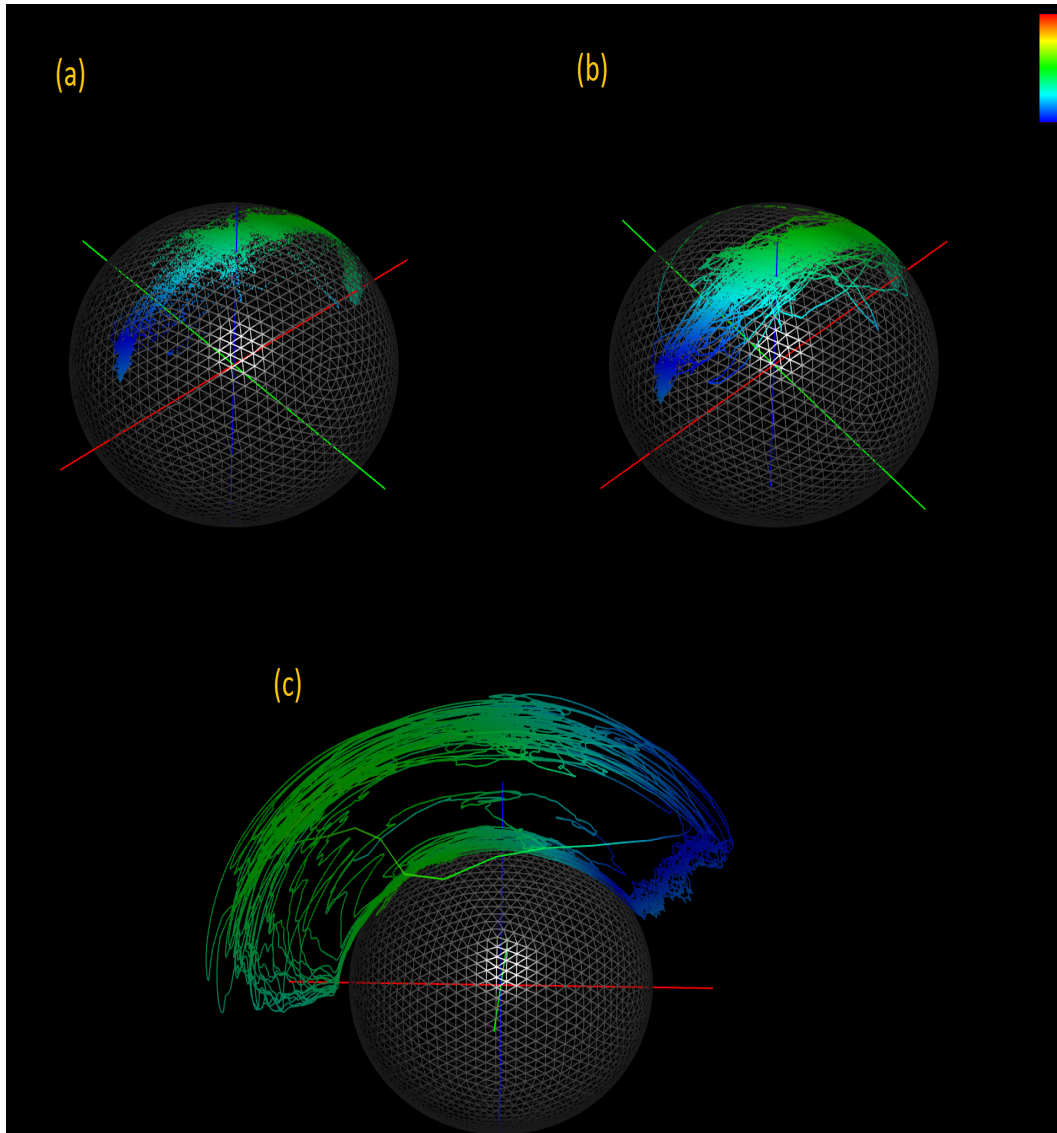


Figure 6.3: A spherical coordinate visualisation of (a) postural state plotted onto the surface of a sphere in three-dimensional space. (b) Each point is joined together in chronological order to show the temporal structure of the data. (c) The data is projected outwards from the sphere associated with depth, this shows a compelling visualisation of orientation associated with depth. This visualisation is useful for small sub-sets of data. However, when large time-spans are visualised the visualisation can become occluded.

6.3.4 Binning in Three-Dimensions

The challenges of dealing with large data sets include occlusion and overplotting problems where data values in a point cloud obscure other values, something that is exacerbated by increasing data (time). A well-established method in the visualisation community is to

employ an overview and focus approach [PCS95] which gives a contextual overview of the data, but provides potential to interact to obtain further details on demand by focusing down into the data. Applying this concept, we divide the surface of the sphere into facets (sphere tessellation) and treat the data within each facet to derive summary statistics (binning).

6.3.4.1 Sphere Tessellation

To represent the underlying data on which the chart is based accurately, we employ a histogram approach across the surface of the sphere using regular bin sizes to summarise the data appropriately [Tuf01]. Constructing a sphere from a series of uniform geometric primitives is a problem from the cartography domain, which allows projections which hold the shape, area, direction, bearing, distance and scale, normally considered unachievable in tandem [VW08].

The traditional method of constructing a sphere is generated by following the lines of latitude and longitude, although this results in variable sized facets misrepresenting the underlying data [Tea06]. We utilise a geodesic sphere, providing a close to uniform and regular sphere tessellation, using subdivision surfaces and spherical projection of an icosahedron platonic solid. The geodesic sphere starts with an icosahedron, each facet is repeatedly subdivided a pre-defined number of times with each of the acquired points projected onto the surface of a sphere (by normalisation). The result is a spherical object constructed of triangles, each of which is of regular shape and area. Teanby [Tea06] experiments with a geodesic sphere for binning to find an almost negligible effect in reconstructing the underlying data despite a slight variation in size and shape of each facet. Our user interface allows the variation of the facet size, and therefore the user can explore interactively the impact of the granularity of the frequency distribution. Figure 6.4 illustrates the icosahedron repeatedly subdivided.

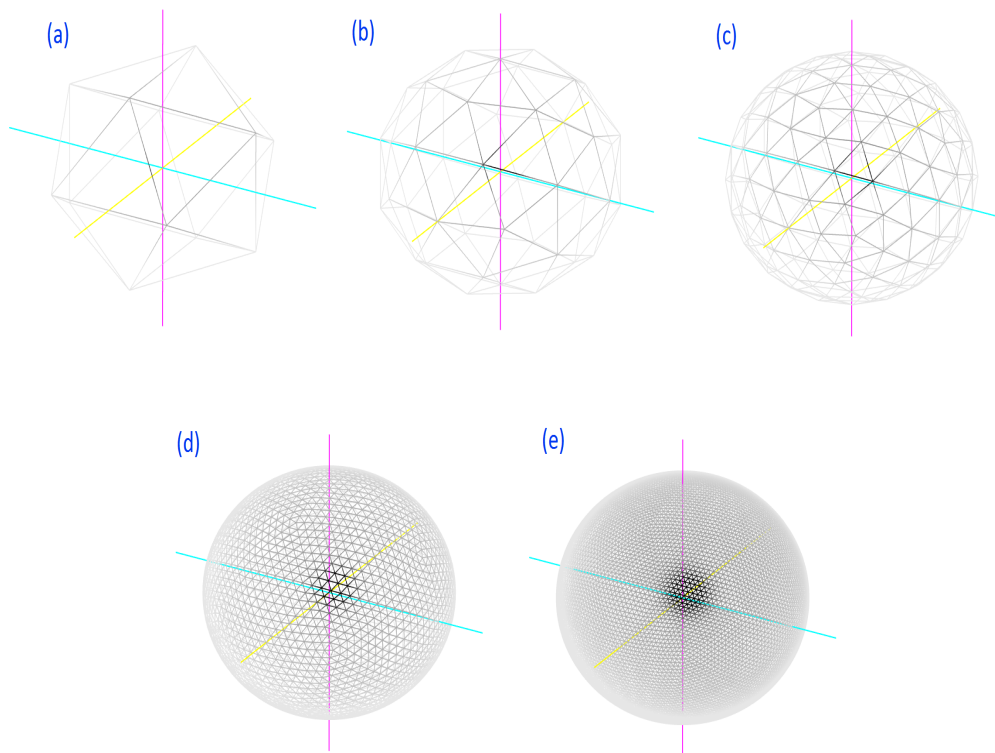


Figure 6.4: This image illustrates the creation of a close to uniform and regular sphere tessellation. (a) The initial icosahedron. (b - e) the icosahedron is repeatedly subdivided with each of the points projected onto a sphere on each iteration.

6.3.4.2 Binning Data

Each facet in the sphere tessellation represents a histogram bin in the resulting visualisation. Binning enables us to identify the facet with which a data item intersects on the geodesic sphere. Teanby [Tea06] propose a winding method which operates by linearly searching for an intersecting facet on the sphere which has a sum of angles with the test vector equating to 2π . Walker [WGJL12] proposes a more efficient method which leverage's the hierarchical structure of the geodesic sphere which operates in an equivalent manner to that of a search tree dividing and reducing the search space to a logarithmic complexity algorithm. The angle (θ) between the direction of the centre of each facet from the origin, (w) and the current vector (v) is computed using the dot product (below). The point is determined to be associated with the sphere facet with the smallest angle between them. This is recursively computed on the hierarchical structure until the lowest-level is reached which identifies the facet the item resides in.

$$\theta = \frac{v \cdot w}{|v| |w|}$$

For each facet, we compute the following statistics;

- (i) the number of data items intersecting each facet,
- (ii) the mean value of each data channel for the items in each facet, and,
- (iii) a frequency distribution of a user-defined data attribute consisting of a user-defined number of bins.

The data for attributes (i) and (ii) are normalised. Normalisation adjusts the values for each of the statistics to a notionally common scale (computed using equation 6.1, where x_{total} is the combined total of the values for the attribute present). This adjusts the values so that across the whole sphere the combination equates to 100%. The distributions for (iii) are normalised locally which allows us to create a histogram of each facet of the power usage occurring for a particular movement and postural state independent of the frequency of the underlying data in the facet (since the frequency equates to a percentage).

$$x_{new} = \frac{x}{x_{total}} \quad (6.1)$$

6.3.5 Spherical Histogram

The binned data for each facet can be displayed as a single histogram projecting perpendicularly from its respective facet. Each histogram length and colour defaults to being proportional to the normalised sample size for the sphere facet (Figure 6.5). The result is a visual which provides an overview of the data distribution over the sphere, illustrating the frequency of postures or movements in the data set. The user may adjust the length of the bins by providing a scale factor. Setting the scale factor to zero collapses each histogram to the surface of the sphere to provide an overlay which does not add additional clutter to the scene. The colour may be encoded as any other data attribute, in addition to the normalised frequency, along with the length.

6.3.6 G-Urchin

In a final evolution of this method, the smoothed, tri-axial acceleration axes (i.e. posture) can be encoded in addition to the frequency of items in each facet. A histogram for each facet of the sphere is computed for the items residing in the facet, which can be combined in a manner that represents the best fit from the combined values of power usage. In effect, we represent the global (i) and local (ii) frequency of data items across and within each facet of the sphere. We call this approach the 'g-urchin', as the technique produces a visualisation with the appearance of a sea urchin, with spines projecting from the sphere.

Each urchin spine is placed at a user-defined distance away from the sphere to avoid occlusion with any of the other layers of the visualisation, although a line is drawn to the centre of the facet it represents. The length and width of each spine can be ascribed to any data attribute. It is most effective when the spine characteristics are linked to histogram frequency or the number of items residing in the facet. Thus, each spine is not visible when the frequency is low. Each spine consists of a number of stacks, the width of which

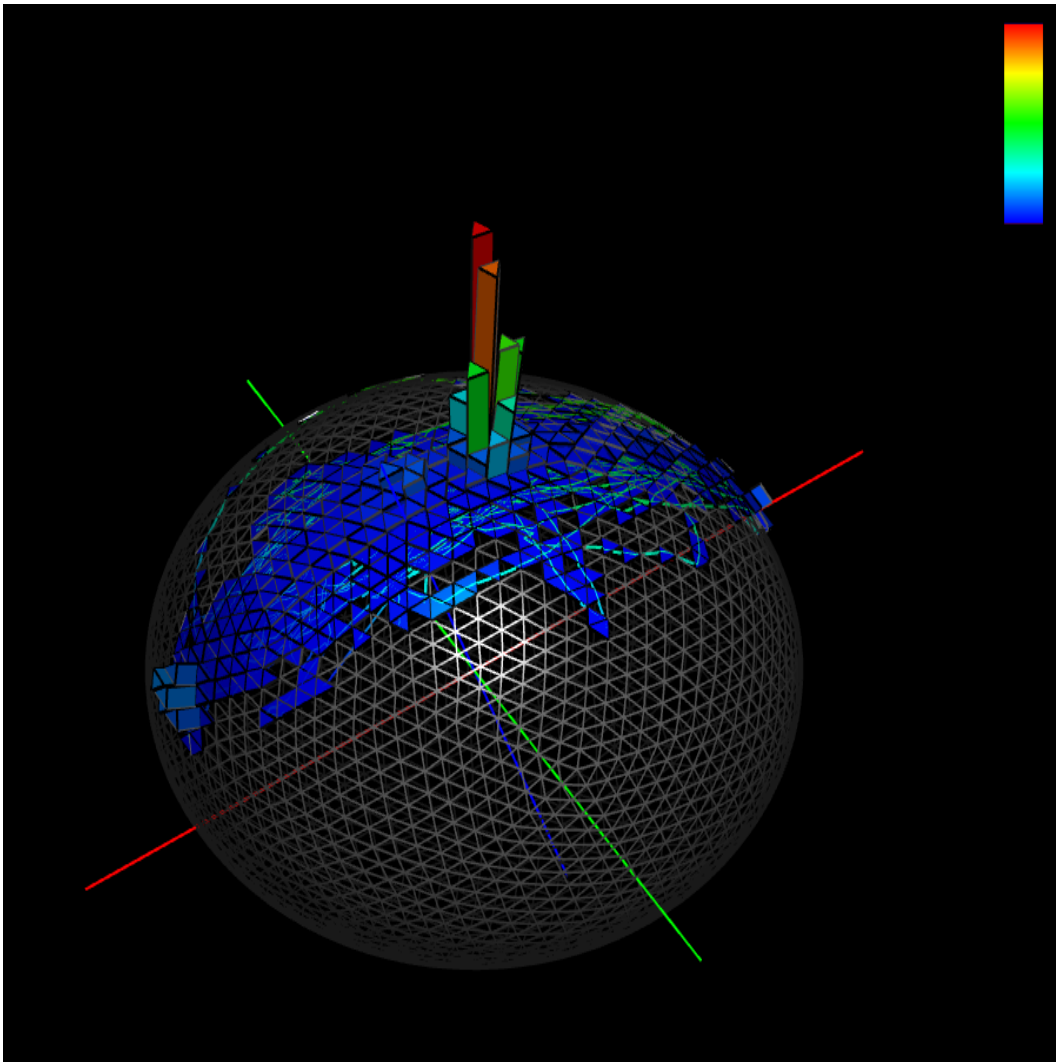


Figure 6.5: A spherical histogram visualisation. This visualisation depicts frequent orientation and postural states. Each column projecting from the sphere represents the sample size of data items contained in each corresponding facet. The longer each column, the more data items residing in the respective facet on the sphere, and therefore more time the animal spent in the specified posture (orientation). This visualisation is of a deployment from a cormorant. The animal spends the volume of the duration horizontal on the surface of the water (depicted by the tallest column originating directly through the z axis of the model). Variations of the orientation depict less frequent diving and flying patterns.

corresponds to the histogram bin width. Performing a double mouse click on any given urchin displays the histogram in a standard histogram view in a separate widget (Figure 6.6 left). We introduce two representations of urchins for representing these data values. Firstly, the fixed shape skittle (Figure 6.6 right), and secondly, the frequency based stack glyph (Figure 6.6 centre).

6.3.6.1 Fixed shape (Skittle)

A fixed shape spine is represented using a Hermite spline [Far14] to create a smooth curve. The start and end points of the spline are fixed at the respective top and bottom centre of the spine, with control points leading orthogonally outwards at a radius defined by the facet frequency. The resulting spline is revolved through 360 degrees around the spine axis to create a skittle like object. The skittle is split into segments defined by the number of bins, with each segment colour coded accordingly to the histogram size as per the corresponding bin. Figure 6.7 (top) illustrates the g-urchin using the skittle structure.

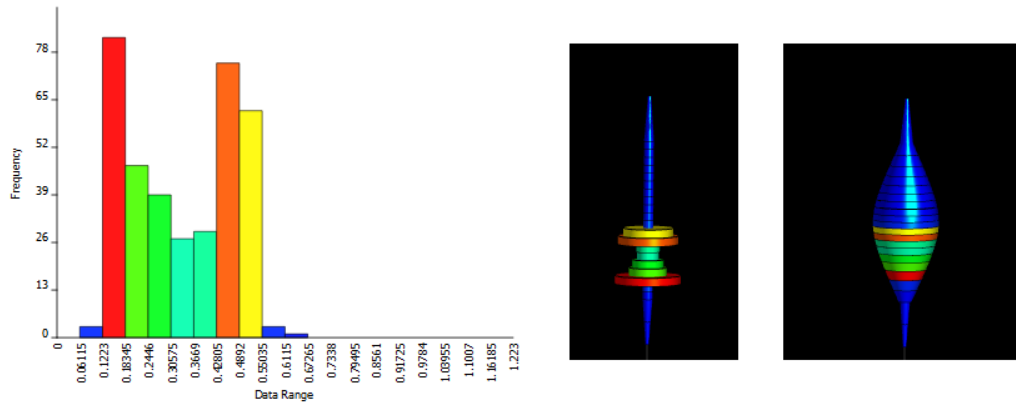


Figure 6.6: Left to right: Histogram, Frequency shape (stacked), fixed shape (skittle).

6.3.6.2 Frequency shape (Stacked)

The second option is to treat each bin in the histogram as a cylinder, with a fixed height corresponding to bin width, but for which the diameter corresponds to the number, or percentage, of elements inside the corresponding bin. Each cylinder is stacked and the resulting height and width scaled by item frequency which is then colour coded appropriately. Figure 6.7 (bottom) illustrates the g-urchin visualisation encoded using the stacked glyph. The resulting visualisation allows us to incorporate power usage and behavioural state together in one visualisation. A highly energetic state will be presented as a large cylinder or red highlight near the top of the urchin.

6.3.7 Differential G-Sphere

Standardisation of the data for time using binning allows a sphere from one situation (species, individual, time period) to be applied with another, providing the g-spheres are of the same sphere tessellation and bin size. We introduce two operations for this purpose; firstly subtraction, which is used for highlighting differences, and summation, which combines g-spheres together. This gives the notion of two sphere types; a data g-sphere generated from raw data, and an operation g-sphere generated by applying an operation to two existing g-spheres. It is important to note that due to the standardisation process, operations can be applied to any combination of the two g-sphere representations (data or operational). The user builds up g-spheres of particular interest and then applies a series of operations to build a g-sphere representation for their analysis needs using the following operations.

Difference is used to subtract two g-spheres; GA , and GB from each other. The absolute difference between the two spheres, for each facet in the sphere (f), and each corresponding bin (b) in the frequency distribution is computed. The result is a new operation g-sphere (G') which highlights the difference between GA and GB .

$$G' = \sum_{i=0}^f \sum_{j=0}^b |GA_{ij} - GB_{ij}|$$

Summation is used to combine two g-spheres together. The items in each bin are added together. The result is a new operation g-sphere (G') which combines the spheres GA and GB together.

$$G' = \sum_{i=0}^f \sum_{j=0}^b |GA_{ij} + GB_{ij}|$$

Each frequency distribution is normalised to eradicate any bias towards data sets containing different number of data points (or durations). The effect in the frequency distribution is a percentage where each bin contributes towards a subset of the distribution. As such, the entire frequency distribution totals 100%. When combining the distributions together by addition or subtraction, the result is the difference in percentage between the two histograms. The operation therefore does not directly compute the difference of the frequency of data

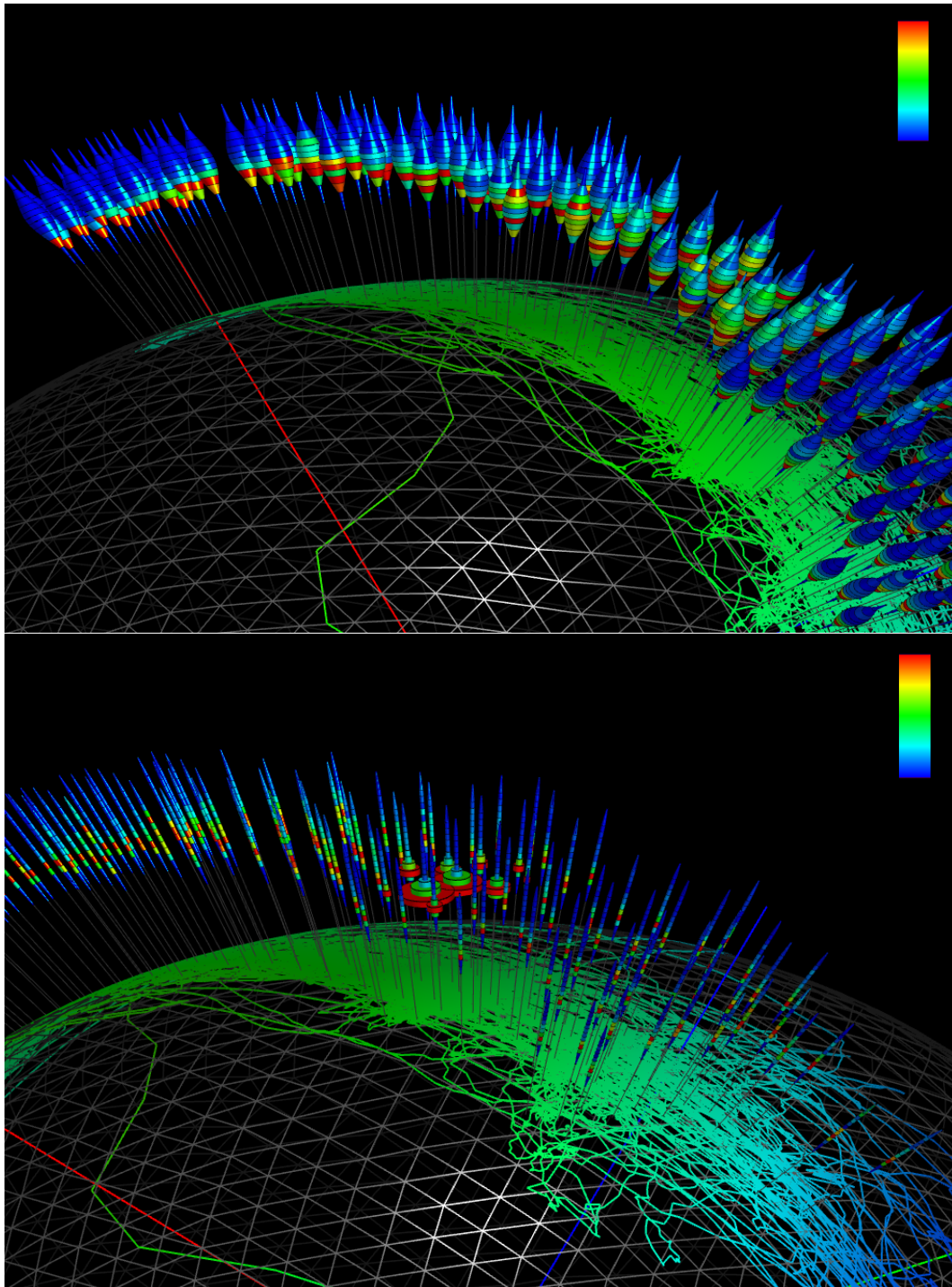


Figure 6.7: G-Urchin of skittle shape (top), and stacked frequency (bottom) urchins emitted from the centre of each facet of the sphere. Each urchin represents the underlying data within the corresponding facet. The size of each histogram bin is colour coded accordingly.

items residing in each bin but the percentage in difference which ignores the size of the underlying data. The newly computed distribution is then normalised again to equate to 100%.

For the operation procedure to be effective, it is vital the g-spheres be in alignment with each other with regards to tag orientation and placement. Prior to an operation taking place, the user is presented with a dialog which shows both g-spheres for alignment. The user can overlay the spheres or keep them adjacent to each other, and specify a number of rotations on each sphere in the x , y , and z , axes until the data are aligned. In future work, it is necessary to investigate an algorithm or protocols in the biological domain to automatically align the two data series.

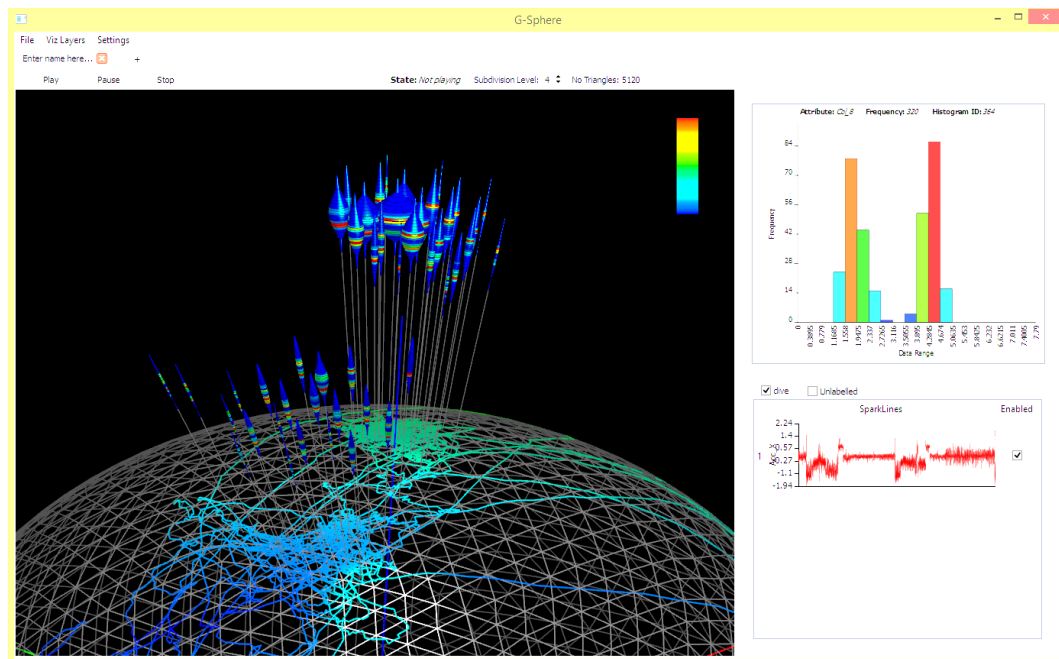


Figure 6.8: Overview of the user interface for a data g-sphere. Top: toolbar for constructing a new g-sphere. Left: visualisation panel of the g-sphere. Top right: histogram view of a selected g-urchin in the visualisation panel. Bottom right: interface to select specific behaviours and data subsets to utilise in the visualisation.

6.4 User interface

A built-for-purpose user interface facilitates the user in building interactive g-sphere and urchin visualisations. The g-sphere user interface is built into Framework4 and made accessible via the ‘tool’ menu. We modify Framework4 to use a multi-tab interface, which allows multiple data sets to be loaded and labelled in the software simultaneously. The g-sphere operates on the data sets and associated classified behaviours. The g-sphere also uses a multi-tab interface for the same purpose of creating multiple g-spheres from different data, attributes, and operations. Upon creation of a g-sphere, a new tab is created which is named by the user for later reference throughout the software. Each tab represents a separate interface, specific for interacting with the corresponding g-sphere. The interface is split into three components (Figure 6.8 illustrates an overview). At the centre is the g-sphere visualisation panel, with which the user can interact by rotating, moving, and scaling to zoom in and out. In the top right, the histogram panel shows an individual instance of a histogram in two-dimensions when a histogram or urchin is double clicked. In the bottom right of the display, the details of the g-sphere are shown. For a data g-sphere, the behaviour classification grouping for the data are shown. Here, the user can select which behaviours to visualise in the g-sphere, which updates the visualisation instantly. Upon creating a new g-sphere the user is queried for the type of g-sphere to create. Figure 6.9 illustrates the work-flow graphically. The user selects to create a data g-sphere or operational g-sphere, which then stems off into the associated processes to create the selected g-sphere. We now describe the creation of data, and operational g-spheres along with their associated interaction techniques.

6.4.1 Data G-Sphere Construction

The data g-sphere is directly created from an underlying data source pre-loaded in Framework4. Upon creating a new data g-sphere the user is presented with a dialog (Figure 6.10 (left)). Here, the user is asked to specify a name for the g-sphere, which is used as a reference for it throughout the application, select the data source to use, and three channels in the data, which correspond to the x , y , and z axes in the visualisation. Typically, this is used to select either the static or dynamic components of tri-axial acceleration, such that, posture or movement can be analysed. The user can select to import settings from another g-sphere, which is useful when the g-sphere is to be applied with another g-sphere to create

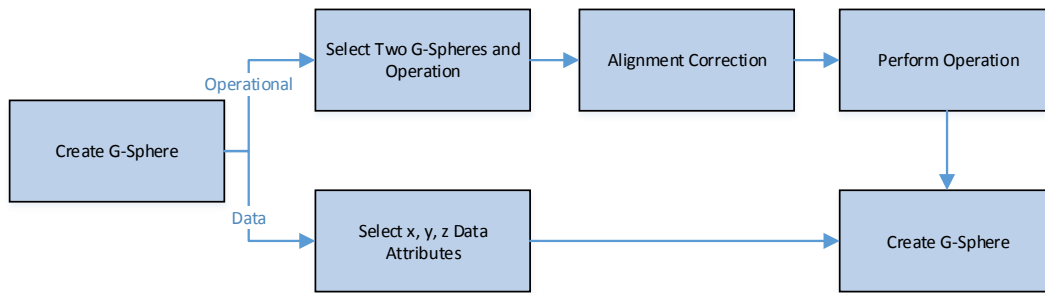


Figure 6.9: This image illustrates the work-flow when creating a new g-sphere. The user selects whether to create a data or operational g-sphere. The data g-sphere is computed directly from the underlying x , y , and z data channels selected by the user. The operational g-sphere requires the user to specify two existing g-spheres (operational or data) and an operation to perform between them. The g-spheres are aligned and then the operation is performed before creating the new resulting g-sphere.

an operational sphere (where the histogram sizes must be equal). Once complete, the g-sphere is created in a new tab.

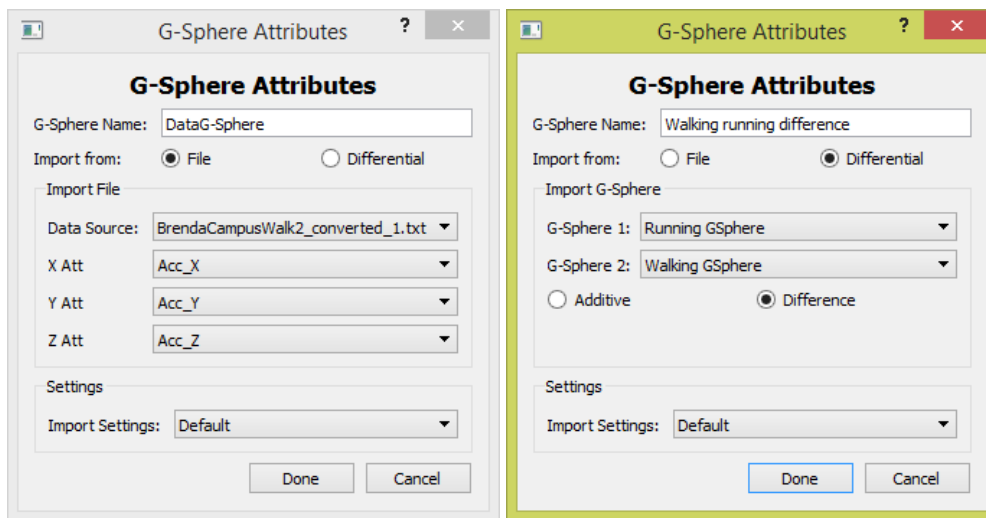


Figure 6.10: The dialog for creating a new g-sphere. Left: the user has selected to create a data g-sphere. The x , y , and z axes must be selected from the data source. Right: the user has selected to create an operational g-sphere. Two g-spheres must be selected, along with the operation to perform between them.

6.4.1.1 Interaction

The data g-sphere tab features three components, (i) the visualisation (ii) a histogram view and (iii) a behaviour selection widget.

Visualisation The g-sphere visualisation is depicted in the central viewpoint of the tab, since it is the focus of the software. The g-sphere is rendered in three-dimensional space. We use an orthographic projection which means not all sides of the sphere can be depicted simultaneously. To allow the user to fully inspect the visualisation, we allow free rotation and displacement of the g-sphere in three-dimensional space, using the cursor and arrow keys.

Each visualisation (raw plot, spherical plot, spherical histogram, and g-urchin) is treated as a separate layer in the scene. This allows the user to view multiple visualisation simultaneously, by enabling or disabling a visualisation layer using the toolbar menu. A settings panel allows the user to adjust parameters for each visualisation (as defined previously), such as, changing the size of each facet on the sphere, or the urchin shape utilised [AMM⁺07].

The g-sphere removes the temporal component of the data series, opting to encode the geometric distribution of the underlying data. Time is typically represented through static

or dynamic presentations in the visualisation community [AMM⁺07]. Static representations visualise time in still images (e.g. temporal axes or storyboards), while dynamic representations utilise the physical dimension of time to highlight the temporal structure (e.g. animation). Dynamic representations are well suited to convey the development of a visualisation over time. However, when longer data series are applied, across multiple-dimensions it can be increasingly difficult for the user to follow changes and cognitively make connections between different temporal points in time [TMB02]. Static representations show all the information explicitly in the visualisation, this allows the user to fully concentrate on the data, although this relies on utilising the screen real estate [GCML06]. Effectively representing time is an active area of research in the visualisation community [AMST11].

To re-introduce the temporal component of the data into the g-sphere visualisation we choose to utilise animation, specifically through allowing the user to playback through the data set to see how the visualisation evolves. This involves iterating a counter at a set time interval, or manually (through a slider control). For a given time index (t), the data set is rendered up to the current time index, such that, the range $0 - t$ is rendered. This is a compelling feature, which allows the user to understand what the animal was doing and when, although it falls short when visualising larger durations of time. In future work we seek to explore methods for effectively representing the temporal structure of the data in the g-sphere visualisation and compare our g-sphere technique to conventional methods performed on traditional time-series graphs.

Histogram view The spherical histogram and urchin plots represent an overview of the distribution of the frequency of the number of data items in each facet of the sphere, and the distribution of items residing in each facet. Upon clicking on a histogram bin, or urchin the user is presented with a detailed view of the corresponding histogram in the histogram view (Figure 6.6 (left)). This allows the user to see the data values each bin represents and obtain details-on-demand of the underlying data.

Behaviour widget In Framework4, we facilitate the user to identify and label behaviours. In the g-sphere visualisation we marry together behaviours with power use. To further aid the user in analysing power use for specific behaviours, we allow the user to enable / disable whole behaviours and individual instances of a classified behaviour. A behaviour widget (Figure 6.11) facilitates this. Each tab represents a behaviour group, with each assigned behaviour illustrated through a stacked time-series graph in a list view. The user can use the corresponding checkbox next to each tab to enable / disable a behaviour group, or select individual instances in the list to build the g-sphere from.

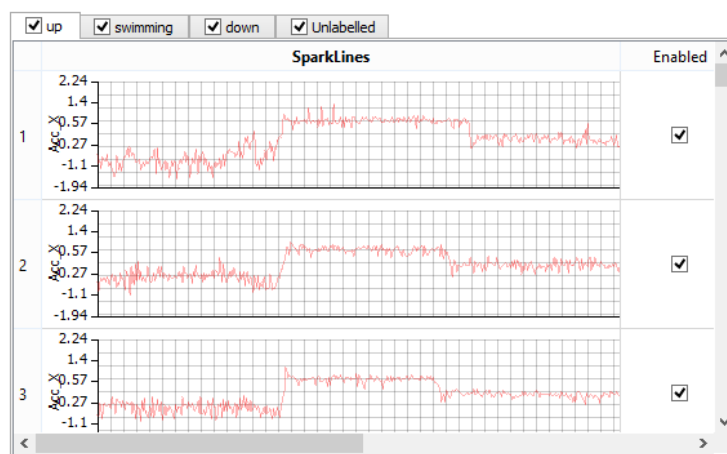


Figure 6.11: A tabular widget of behaviour classifications in the data is shown in the bottom right of a data g-sphere window (Figure 6.8). The user can enable / disable whole behaviour groups, and individual behaviour instances to visualise.

6.4.2 Operation G-Sphere Construction

An operation g-sphere is constructed by applying an operation on two existing g-spheres (data or operational). The user specifies two existing g-spheres in the construction dialog (Figure 6.10 (right)), along with the operation to apply between them (difference or summation). Once complete, the user is asked to align the two spheres together, since it is vital the g-spheres be in alignment with each other with regards to tag orientation and placement. The g-sphere alignment dialog (figure 6.12) allows the user to align two g-spheres together. A preview of the g-spheres is shown using the histogram, or raw plot visualisations of the underlying data to assist the user in aligning them. The user rotates each sphere using spin boxes for rotations on the x , y , and z axes, and using the cursor to rotate each g-sphere. Once complete, the coordinate systems of the spheres are aligned by the user-defined transformations and the operation selected is computed.

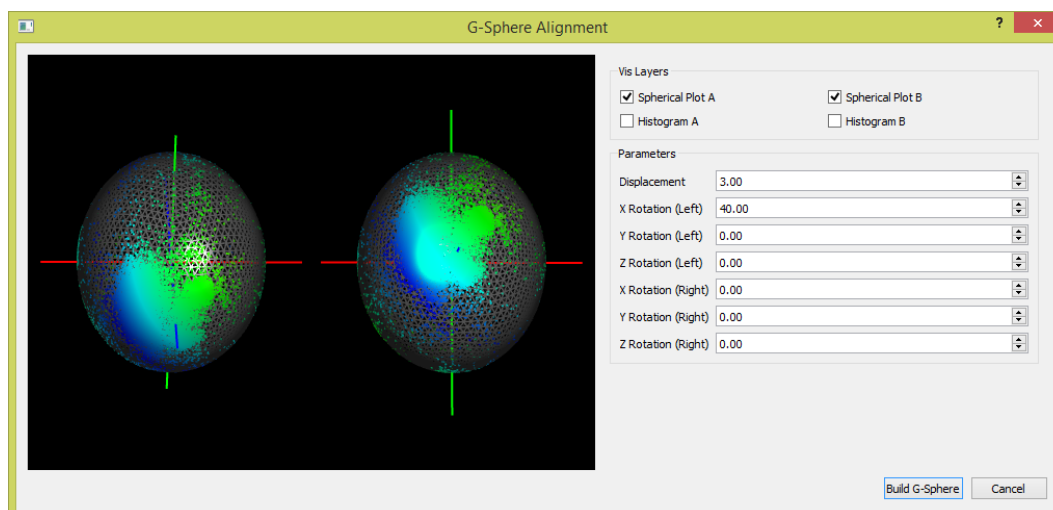


Figure 6.12: The g-sphere alignment dialog which allow the user to align two g-spheres together. The user selects the amount to rotate the left, and right g-spheres by using (a) the spin boxes on the right, or (b) the cursor to rotate the g-spheres. The user can select the visualise the histogram, or raw underlying data to assist in the alignment.

6.4.2.1 Interaction

The operational g-sphere does not represent any underlying data, but aggregated data summaries. As a result of this, the raw plot, and spherical scatter plot views are disabled. Only the aggregation visualisations (spherical histogram and g-urchin) are relevant in this view. We therefore only allow the user to enable / disable these visual layers. Similarly, there is no temporal component (as the derived data is frequency based), or behaviours to select between. The behaviour view and temporal playback features are disabled. The visualisation window remains central, with the user able to rotate the sphere and displace it. To the right of the visualisation display is the histogram view (discussed prior) where the users can obtain details-on-demand through clicking on the sphere facets in the histogram and urchin views.

6.5 Results

6.5.1 The Basic G-Sphere

An animal-attached tag mounted in the centre of an animal's back with orthogonal, tri-axial accelerometers (aligned with the major axes of the body) produces a 'static' acceleration signal with a vectorial sum of 1.0g when the animal is stationary due to gravity. Plots of such tri-axial data in a 3-d graph therefore tend to populate the outer surface of a sphere (a 'g-sphere') which becomes most apparent as the animal adopts body orientations with multiple combinations of body pitch and roll (Figure 6.13 (a)). When animals move, points on the g-sphere may leave the sphere surface as individual acceleration values reflect g-forces

derived from the animal acceleration. This has been termed 'dynamic acceleration' and can be dealt with in a number of different ways, one of which is to remove it by selective smoothing [SWH⁺08] and normalising (see methods) to leave the postural data. Thus, body attitude, which is a major step in elucidating behaviour [WWQ⁺06], is defined by the position of the data points on the sphere and different animals have different distributions of point distributions depending on their lifestyles.

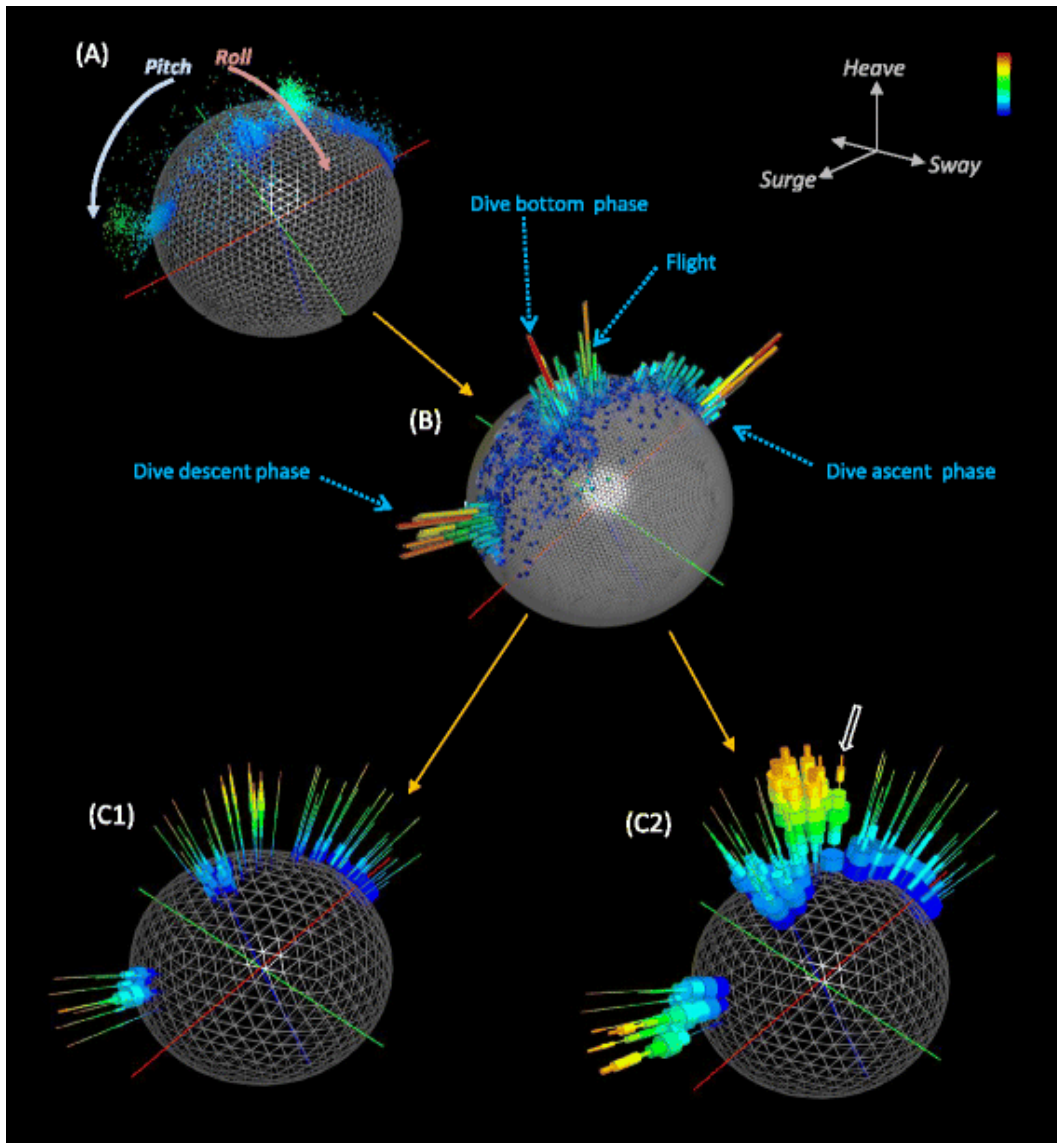


Figure 6.13: Example behavioural data from a cormorant. Six dives and a short period of flight are visualised by (a) a point-based g-sphere (with point colour equating with dynamic body acceleration). (b) shows the same data as (a) but as a histogram plot. Both (c) images depict urchin plots of (b); (C1) shows percentages of dynamic body acceleration allocation taken across the whole g-sphere while (C2) shows percentages amounting to 100% per facet. Note the higher values of dynamic body acceleration attributed to flight and descent of the water column, particularly emphasised by the 100% facet percentage. Note also how certain spines show multi-modes (e.g. white arrow) which can be indicative of different behaviours at one body attitude.

6.5.2 The Spherical Histogram Plot

Increasing time periods viewed within the basic g-sphere tend to result in increasing occlusion and over-plotting of the data, making visualisation more confusing and less useful as the number of data points increases (Figure 6.13 (a)). A representation of the time allocated to various postures can, however, be obtained by tessellating the surface of the g-sphere

into facets, summing the data points within each facet, and presenting the number of points within each facet by a projection into space away from the g-sphere, producing a spherical histogram (Figure 6.13 (b)). Such plots typically show modes representing behaviours with smaller peaks indicating less time spent engaged in that behaviour (Figure 6.13 (b), Figure 6.14 (a)).

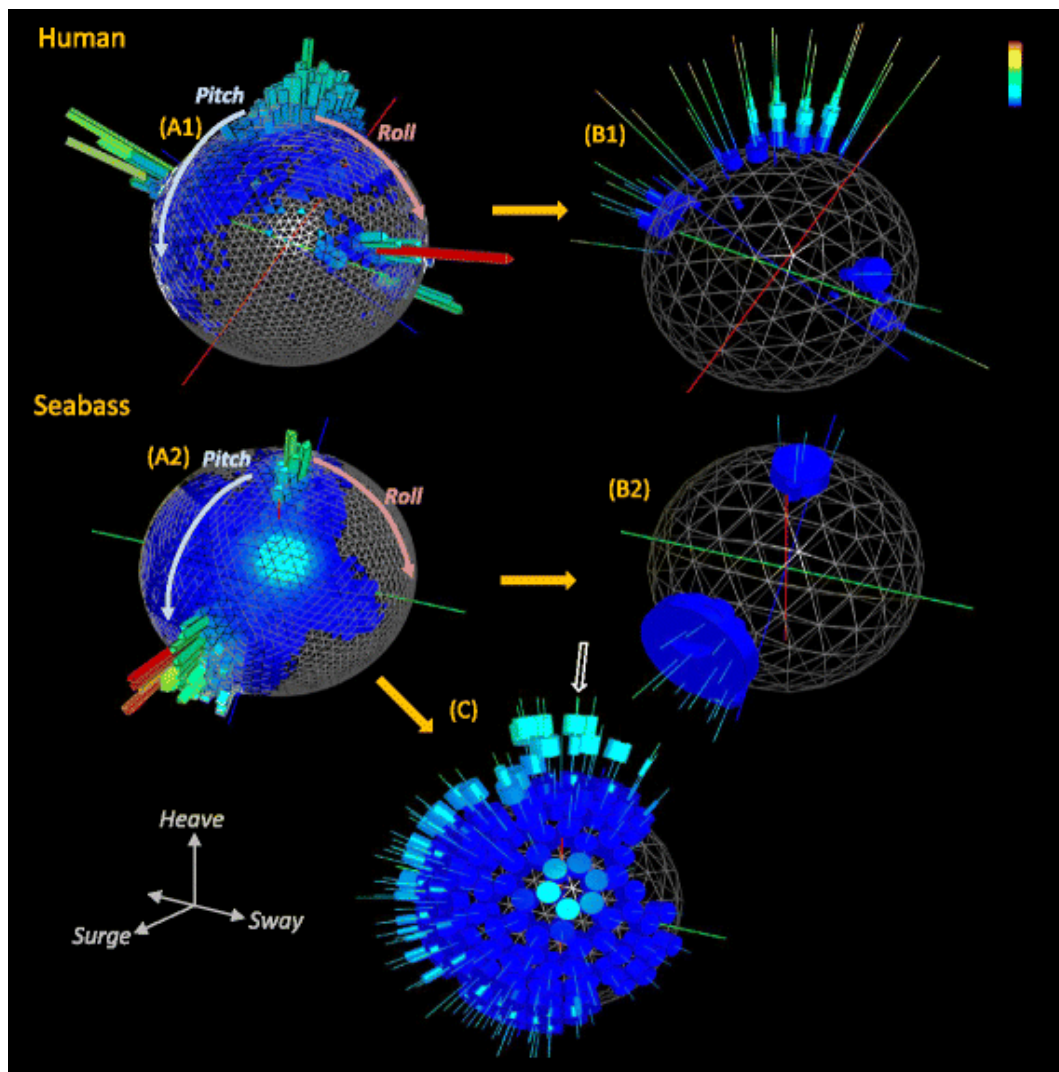


Figure 6.14: Examples of posture and energy-linked posture visualised for two contrasting species (a human and a fish) over 24 hours. The human data are taken from a person on a walking/camping tour while the fish data are from a hole-dwelling reef species that often rests by wedging itself at unusual angles. The left-hand figures (a) show spherical histogram plots, indicating how time is allocated to different body postures (the 'North pole' position shows the species in the 'normal' upright position). The first right-hand figure for each species (b) shows how each posture is linked to varying putative power levels. Note how the human has higher power-proxy levels associated with the vertical posture due to walking. Both the human and the fish have low power-proxy levels at low 'latitude' angles acquired during resting/sleep, exemplified by the large diameter blue discs. Data normalised to give a global percentage for all angles may hide infrequent, but higher-energy, activities. Normalising the data to 100% per facet (c) highlights these though. In this case, the low-energy life style of the fish is still apparent (B2), with higher energies occurring fleetingly and only when the fish is vertical (white arrow). The colour coding has blue as low, and red as high, values.

6.5.3 The G-Urchin

While basic g-spheres and spherical histogram plots quantify the time allocated to different postural states, they impart no information on power use. This information can be incorporated into the g-sphere by calculating the dynamic body acceleration (see methods), which

correlates linearly with power [HSW11], for each of the postural data points within each facet on the sphere. In this, we note that although one study has shown that a strong relationship between dynamic body acceleration and energy expenditure holds for a (seabird) species operating in three media and multiple different body angles [ELVK⁺12], confirmation that this is also the case for more species will need further work (see [HSW11]). To visualise this, the sphere facets can be populated with thin spines, one spine per facet, radiating into space, like a sea urchin (facets without data have no spine). Spines acquire stacked rings representing the frequency distribution of the dynamic body acceleration values associated with that posture/facet. The position of each ring on the spine indicates the dynamic body acceleration value (lower values are closer to the g-sphere surface), the depth of the ring indicates the width of the dynamic body acceleration bin, and the diameter of the ring is proportional to the number of data points within that bin (Figure 6.13 (c) and Figure 6.14 (b)). This 'g-urchin' can be represented so that it is normalised for all data across the sphere, which highlights the processes that dominate in terms of both the time and proxy for energy across the whole time period considered. Alternatively, data can be normalised within each facet to highlight the energetic proxies of particular postures irrespective of their time contribution (Figure 6.13 (c2) and Figure 6.14 (c)). Urchin plots thus show differences between behaviours within species (Figure 6.13 (c1)), differences in lifestyles between species (Figure 6.14 (b), and (c)), and differences in behaviour of any individual through time (Figure 6.15).

6.5.4 The Differential G-Urchin

The process of comparing individuals or the same individual over different times can be enhanced by subtracting one spherical histogram plot or one g-urchin from another. These differential plots can be colour-coded, for example, according to which dynamic body acceleration bin from which urchin has the higher value (Figure 6.16). This highlights differences in assumed power use associated with posture and therefore behaviour, with notable changes even associated even with state [WGM⁺14].

6.5.5 Contrasting Patterns of Acceleration of Different Body Parts

The approach described above is nominally conceived for accelerometers mounted on the trunk of the body because its movement most properly accords with dynamic body acceleration proxies for power [WWQ⁺06]. Nonetheless, the approach appears to be useful for comparing the consistency of movement between body parts and their interaction, for instance, deriving g-spheres from accelerometers attached to both the trunk and the hand of the same individual (Figure 6.17).

6.6 Discussion

Application of g-spheres and their derivatives to raw tri-axial acceleration data adds another potential avenue to visualise and identify behaviour [NSFR⁺12]. This approach concatenates 6 complex lines of acceleration data into one plot binding animal attitude and proxy for power use into one visualisation that shows modes of behaviour (Figure 6.13). The indicated value lies in its potential for use as a template match approach for specific activity pattern identification across data [Bar06]. Thus, behavioural description and identification (Figures 6.13, 6.14, and 6.15) may not require matched observed behaviours with example data but stem from a visually apparent clustering within the plot. In particular, differences between various g-sphere derivatives, especially spherical histogram and urchin plots (Figure 6.16), may have use in identifying specific variation in posture and power-use proxies between behaviours. For example, the spherical histogram plots in Figure 6.16 (a) provide an example of how the posture of a subject changed according to whether they had watched a happy or sad film clip, with the allocation of time to facet position changing. Similarly, the posture and allocation of dynamic body acceleration to different body postures during walking changed after a fall (Figure 6.16 (b)). The g-spheres therefore employ fundamentally different principles to other methods in the manner of data visualisation and interpretation.

In a first iteration, the most common behaviours are identified because of the way they dominate the basic g-sphere visualisation (Figure 6.13), which is an important feature of understanding time management in animals. However, even behaviour that is only a small

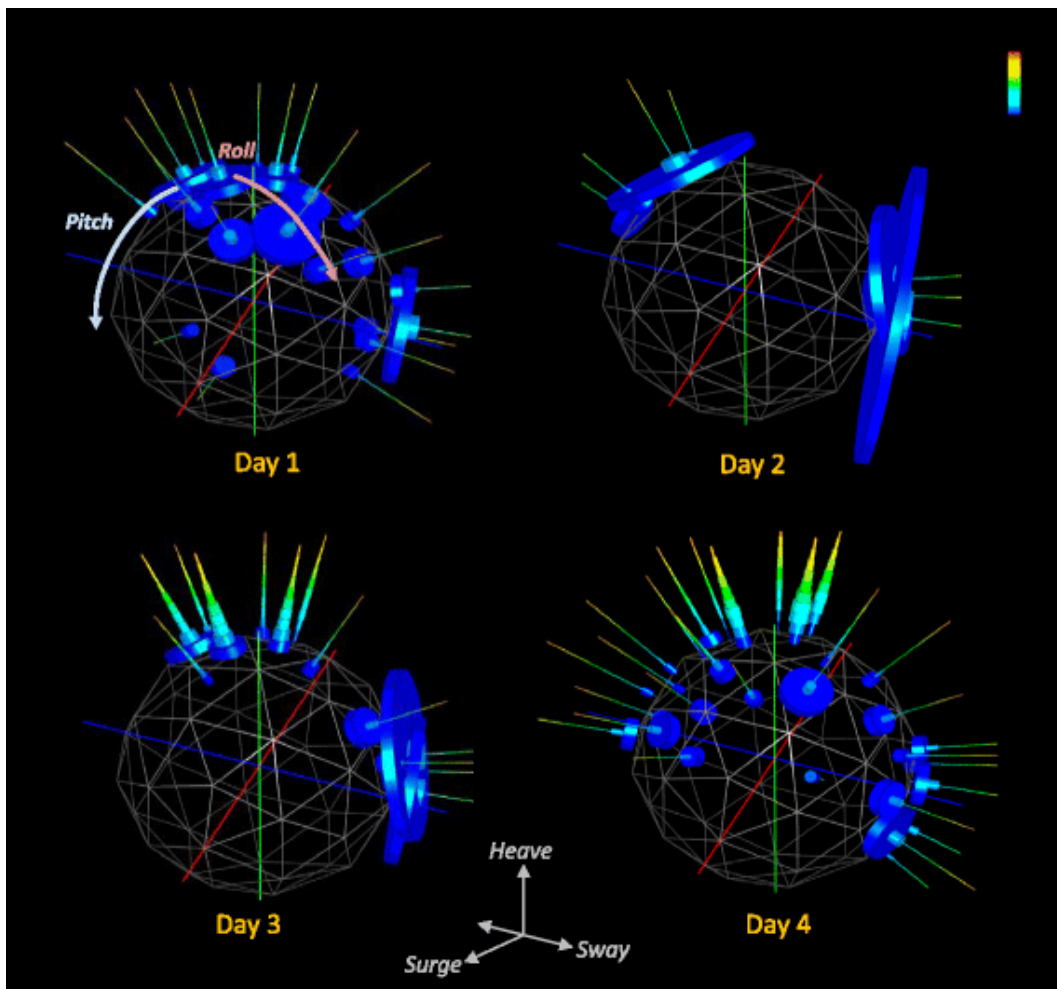


Figure 6.15: Example urchin plots for four consecutive 24 hour periods after the release of a European badger (wearing a collar-mounted accelerometer) following anaesthesia. The 'North pole' facets show when the animal was properly horizontal (i.e. in standing or walking posture). Note how the first two days show no high energy activity because the animal was either resting or asleep. The second day shows only four changes in position. By day three, higher energy, normal posture activities such as walking are apparent at the North pole. This process is further enhanced in day 4, with North pole spine dynamic body acceleration distributions having modes that have moved up the length of the spines to indicate higher power use. Dynamic body acceleration values are colour-coded with maximum values (in red) of 1 g

fraction of the time budget, but is energetically distinct and therefore likely to be apparent in the dynamic body acceleration distributions on urchin spines, may be identified by moving from the globally normalised g-urchin to one that is normalised to facet (Figure 6.14 (b), and (c)).

Importantly, mono-, bi-, or even tri-modality in the frequency distributions of dynamic body acceleration allocated to particular facets or groups of adjacent facets, point to multiple behaviours occurring at similar animal postural attitudes. This is illustrated, for example, in the cormorant behaviour where the white arrow in Figure 6.13 (c2) shows multi-modality in dynamic body acceleration due to both dive ascent behaviour and flight behaviour being apparent in the same body attitude facet. It is also exemplified in the stationary and swimming behaviours in the seabass, shown in the bi-modality of the dynamic body acceleration distributions along urchin spines at the North Pole (see Figure 6.14 (c)). The time-based adoption of behaviours may also be studied with this, for example, in the badger data presented (Figure 6.15). Here, 'normal' walking behaviour is only manifest during day 4 post-sedation, when the urchin spines at the North Pole acquire a dynamic body acceleration mode that is greater than 1.0 g (Figure 6.15). Such observations could be incorporated into statistical classifiers and classification algorithms, an area we seek to investigate its

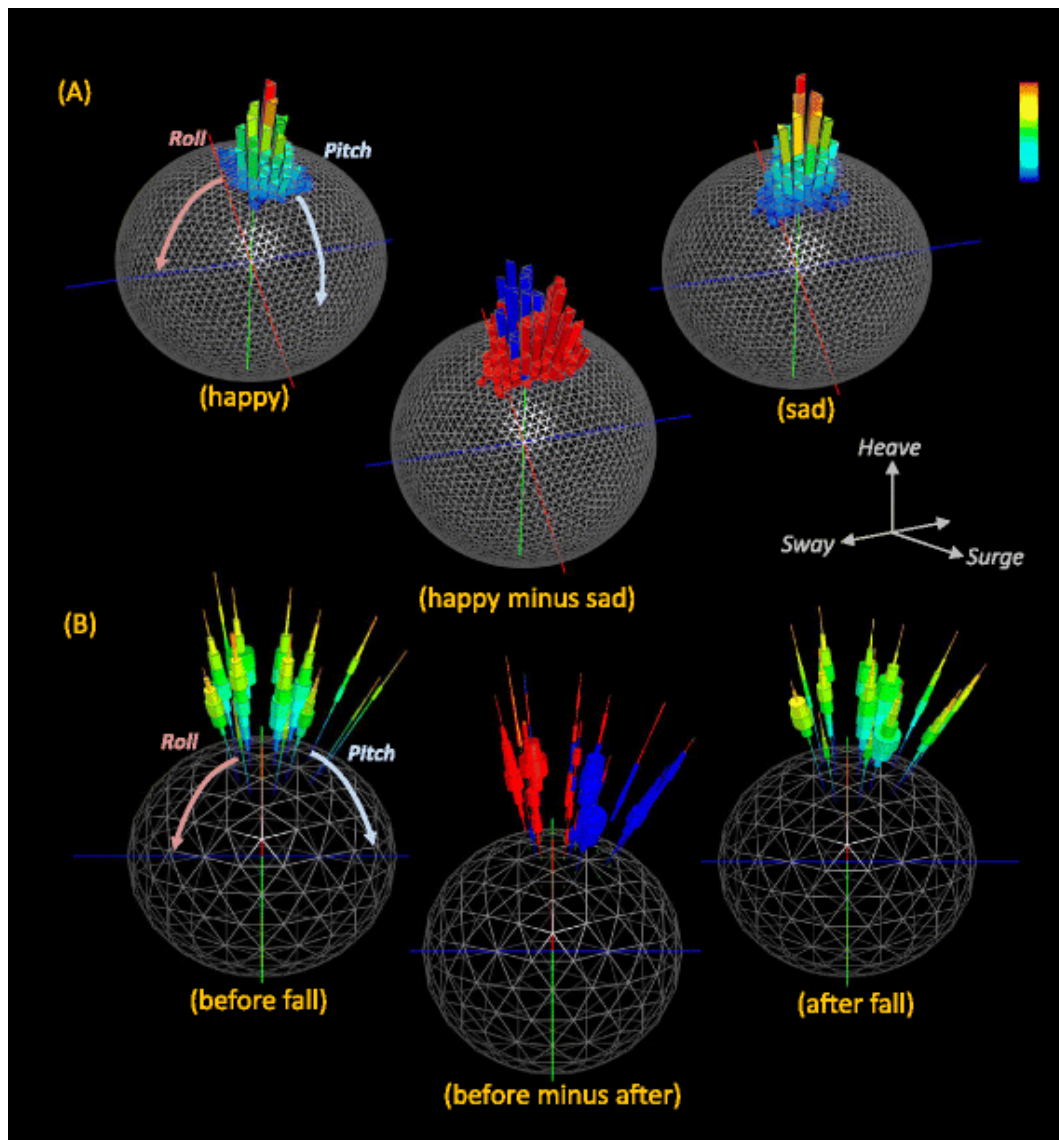


Figure 6.16: Example posture and dynamic body acceleration values associated with 'state' in humans. (a) shows two spherical histogram plots for a person walking after seeing 'happy' and 'sad' film clips. A third differential spherical histogram plot highlights the difference between the two situations (blue = a higher relative frequency of 'happy' points per facet while red = a higher relative frequency of 'sad' points per facet). Note how the two conditions are reflected in the postural changes (b) shows urchin plots for someone trekking across snow pulling a sledge one minute before a fall and one minute after recovering from the fall. The differential urchin shows both differences in postures adopted between the two situations as well as the dynamism of the walking (red shows a higher relative dynamic body acceleration frequency 'before the fall' while blue shows the reverse).

application for in future work.

Generation of frequency distributions of dynamic body acceleration, as a proxy for power, may also enhance the process of separating behaviours. It also visualises the overall allocation of power proxies, either to specific behaviours over short periods such as seconds or to collections of behaviour over longer periods (see Figures 6.13, 6.14, and 6.15) extending to months or even years. Depending on the timescales, collections of particular behaviours could provide a representation of different lifestyles, as well as their considered associated energetic outlay, allowing powerful comparisons to be made between systems or scenarios. Examples include comparisons between species with contrasting lifestyles (Figure 6.14) or within species lifestyle comparisons. The future may also benefit from the use of g-sphere approaches based on multiple accelerometers used on different parts of the body or even having accelerometers on hand-held objects. The expectation

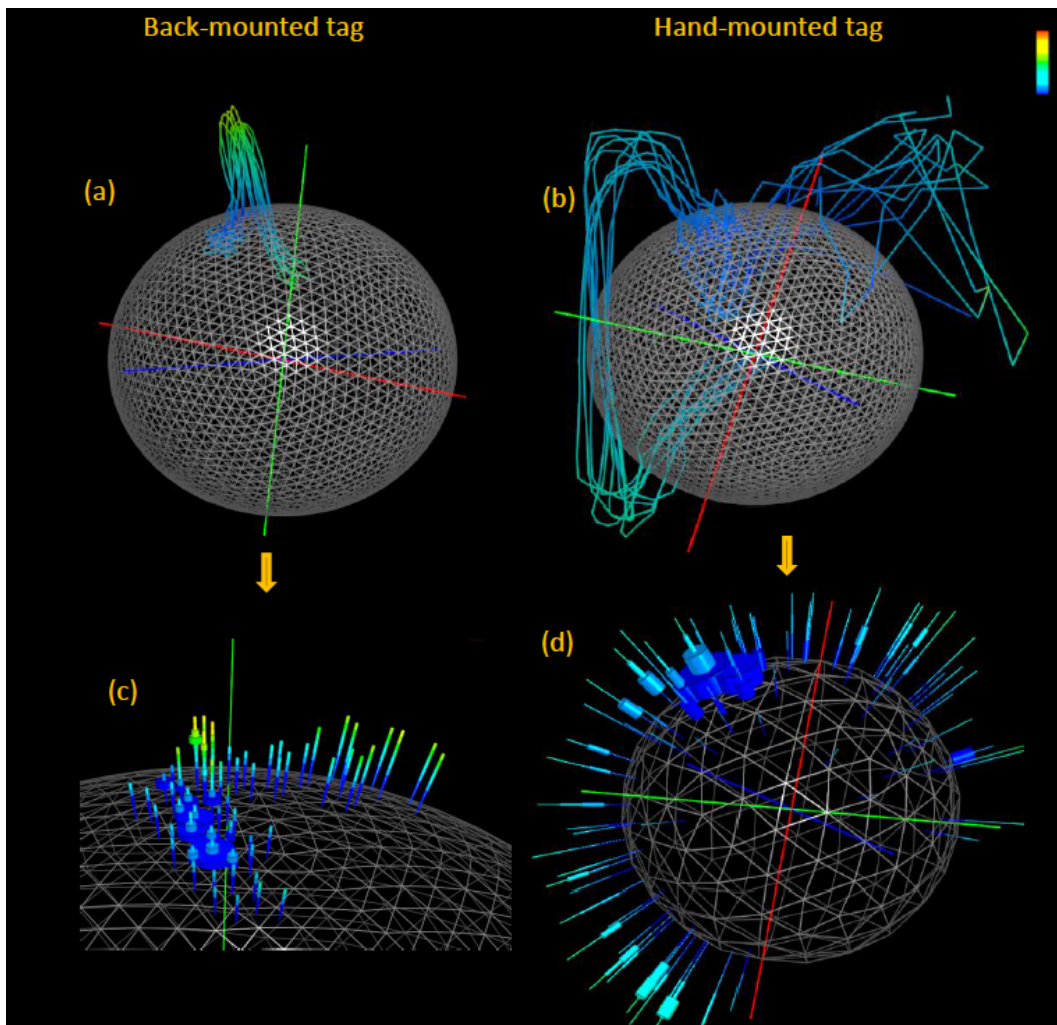


Figure 6.17: Acceleration signal patterns according to tag placement on the body. The figure shows (two upper visualisations) raw data from (a) a back-mounted tag synchronized with (b) a hand-mounted tag during the last 1.5 seconds of 10 basketball penalty shots. The two lower visualisations (c) and (d), show the corresponding urchin plots. The colour coding has blue as low, and red as high, values.

is that this may be particularly useful in sport applications (Figure 6.17) where effective movement must be stylised for maximum performance because limb-, or sports equipment-mounted sensors will represent local forces and perhaps local power-usage proxies better than trunk-mounted systems which produce a body-integrated signal. Importantly, such power-proxy comparisons, from trunk- or limb-mounted sensors, may help identify efficient solutions to activities where performance, such as running speed over a given distance or animal breeding success over months, should be equatable with the putative energetic cost. This sort of consideration thus may have advantages for elite athletes as well as for conservation bodies examining the costs of the lifestyle of their animals. Equally, changes in behaviour that occur with disease or illness, such as constrained activity stemming from rheumatoid arthritis [SSC⁺10], could be rapidly identifiable using this approach.

From our initial results we hypothesise that g-spheres and their derivatives may form the basis for summary statistics which highlight particular aspects of performance, behaviour and lifestyle, which may function to be powerful descriptors of e.g. animal lifestyle, linked, among other things, to physical limitations based on taxonomic, allometric or environmental (e.g. water versus terrestrial) constraints [WBE97]. In addition, such visualisations may help both children and adults to understand how the physical activity levels in their lifestyles compare to those recommended [Act11]. Further work is required to investigate these hypotheses and substantiate the g-sphere approach through a formal evaluation of the methodologies along with domain level case studies in the biological, sports science, and medical fields.

6.7 Summary

Studies on animals and people are increasingly using tri-axial accelerometers sampling at higher rate frequencies to quantify behaviour or energy expenditure. Such studies produce high data volumes and typically use analytical methods that concentrate on analysing either behaviour or power. The combination of studying power and movement in tandem has not been explored prior, even though, because behaviour is manifest by movement which requires energy, both are inextricably linked. In chapter 5 we introduced Framework4 a software tool for deriving both power and movement and combining them into one unified file format. In this chapter, we utilise the derived attributes and demonstrate how both behaviour and power use can be linked by visualising tri-axial acceleration data in 3-D plots (g-spheres) which capitalise on a suite of different acceleration-derived metrics and allow multiple dimensions to be visualised simultaneously.

The g-sphere approach discards the temporal nature of the data, instead favouring the visual encoding of the geometric distribution of the data. We re-introduce the temporal component of the data using animation, although, when longer data series are applied, across multiple-dimensions it can be increasingly difficult for the user to follow changes and cognitively make connections between different temporal points in time [TMB02]. Furthermore, the three-dimensional nature of the visualisation can introduce overplotting and occlusion, something we have investigated in the earlier chapters. Frequency based approaches are applied (histogram and g-urchin techniques) to assist in providing an overview of the data, but these themselves can occlude each other in the three-dimensional environment (since one primitive may be legitimately positioned behind the other). Future work is required to investigate how to overcome these issues. Despite these limitations, our initial results indicate that the g-sphere approach can highlight patterns that are not readily accessible using traditional time-series visualisation and may be used to inform our understanding and identification of processes in areas as disparate as sports practice, disease and wild animal ecology. Further work is required to investigate this hypothesis and substantiate the g-sphere approach through a formal evaluation of the methodologies with users along with domain level case studies in the respective fields.

7. CONCLUSION AND FUTURE WORK

*"If I have seen further than others, it is by standing upon the shoulders of giants."
– Isaac Newton, English physicist and mathematician*

Contents

7.1 Conclusion	138
7.2 Future Work - Smart Sensor Data Visualisation	141
7.3 Future Work - Time-Series Visualisation	142

TEMPORAL data is ubiquitous. Virtually every field of science and business is linked to some form of measurements over time, which all have strong connections to temporal phenomena. Data is rapidly becoming larger than ever before as our ability to collect data continuously increases due to technological advancements. Understanding temporal patterns is key to gaining knowledge and insight. However, traditional analysis techniques are often not adequate to deal with the complexity and size of the data we are surrounded by today. There is a growing need for advanced knowledge discovery techniques to make sense and use of an ever-increasing information space.

Information visualisation is one way of realising this, utilising human perceptual capabilities to gain insights into large and abstract data sets to make large data coherent. Visual analytics is a recent outgrowth from the field given the complexity of a growing information space, combining the processing power of the computer, with the analytical reasoning capabilities of humans through interactive visual interfaces. The application of visual analytics to abstract temporal data for the knowledge discovery process presents many difficulties. Temporal data is challenging in its representation, size and scalability, high-dimensionality, and its unique structure. In chapter 2, we introduced the field of data mining, an overview of state-of-the-art information visualisation methods, and an evaluation of the current state of the visual analytics techniques applied to temporal data. We classify techniques by the tasks they support the user in undertaking on temporal data, these are typically composed as; classification, clustering, indexing and retrieval, pattern discovery, prediction, and segmentation. The review demonstrates the application of visual analytics for solving difficult problems arising from time-oriented data. Visual analytics is a young and emerging field, despite this our classification table highlights an abundance of unsolved problems and missing links in the visual analytic literature.

One challenge is that screen resolution is small in comparison to data storage capacity. When more data items are rendered than the available pixels an over-plotting problem occurs where more than one data item is assigned to each pixel, which leads to a loss of information. In chapter 2, we provided a graphical overview of these approaches, which can be decomposed into, multi-scale representations, frequency, and lens based interaction techniques. In chapter 3, we evaluated these visualisations and extensions to these existing approaches. Based on our evaluation we propose TimeNotes, a visualisation technique utilising built-for-purpose interaction techniques that supports analysis, interaction and presentation through a new navigation technique for visualising and interacting with time-series data. We undertake and report an empirical study and a field study. We test both static and interactive features of our environment confirming validity of state of the art techniques and their integration into a novel approach. Our findings illustrate that TimeNotes provides a more effective working environment for the exploration, analysis, and presentation of time-series data.

Our user study and longitudinal study with domain experts uncovered that the reality of researchers faced with a large number of dimensions recorded at high frequency (i.e. resulting in a large number of data samples) is that it is time consuming and error-prone for the domain expert to manually label and identify specific phenomena in the data. Machine learning techniques have been considered, but have been difficult to introduce because of the large numbers of training sets required and their discriminating precision in practise. In chapter 4, we introduced TimeClassifier, a visual analytics system for the semi-automatic

classification of animal behaviour, combining the search of large time-series data with the semi-automatic classification of events using user-defined templates. TimeClassifier focuses on algorithmic machine learning with visual analytics to assist in accelerating the labelling of animal behaviour. It therefore operates through a separate study and does not build on the TimeNotes platform, although it is incorporated into the same software suite. In future work, we seek to validate the combination of TimeNotes with TimeClassifier. Our system brings together the expert knowledge of the user with those of the pattern matching capabilities of the matched filtering algorithm applied to classification. TimeClassifier requires one instance of behaviour for the matching process to take place. Visualisation is utilised to search for matching behaviour classifications and interaction to facilitate a user in the loop approach for the checking, rejecting, and accepting of results to maintain a high accuracy. We report a number of real-world case studies which show that involving the user-in-the-loop overcomes the issues faced with machine learning techniques which leads to more trust in results, along with increased efficiency and accuracy.

We extend TimeClassifier in chapter 5 to encapsulate the three remaining elements considered pivotal for movement ecologists advocated by the daily diary device [WSL08]. These are, dead reckoning movements, body acceleration derived energetics, and positional linked environmental data. In addition to the template matched behaviour capabilities of TimeClassifier and a new way of combining high resolution motion sensor data with GPS data to derive animal movement at a fine scale of granularity. We expose these methods through the introduction of a user-friendly software solution, named, Framework4 which allows the user to derive meaningful data rapidly from the complex smart sensor data. The result of the derived attributes is output into one single time-aligned unified file for further analysis outside of the software. We report two impact case studies of its use. Firstly, a new head mounted logging device for deriving head orientation supported by Framework4. Secondly, the insights Framework4 have made possible to determine rules behind the way in which humans frame the environment they are in. All the work we present in this thesis has been included in the Framework4 software package, which we make publicly available on the world-wide web.

While the ability to derive and export the four key attributes advocated by the daily diary into one time aligned unified format is empowering, we find that traditional methods for analysis do not allow biologists to make use of the data channels bound together, but instead operate on the individual attributes separately. In chapter 6, we introduced a work-in-progress visualisation technique called the g-sphere, which marries together behavioural data with power use for movement over time. We capitalise on different acceleration derived metrics which allow multiple dimensions to be visualised simultaneously. Since, behaviour is manifest by movement which requires energy, which are both inextricably linked, its usage has the potential for biologists to understand and identify processes in areas, such as, sports practice, disease and wild animal ecology.

The goal of this thesis identified in the introduction was to explore visualisation and visual analytic techniques which enhance the understanding of high resolution, multiple channel (and therefore multi-dimensional) time-series data with an application towards assisting movement ecologists in analysing data acquired by smart sensor tags [JMF03]. We began by studying biologists work-flows to decompose the research challenges they face into specific domain level problem statements we sought to investigate. It was the hypothesis of this work that visualisation coupled with machine learning (where necessary) can effectively and efficiently enable biologists to undertake these tasks. We now reflect on these statements to report our findings throughout the project.

1. **There is no protocol used for mounting devices during deployment.** In this work we have not been able to establish methods which normalise the acceleration data to remove the inconsistencies arising from different mounting locations used and devices moving during deployment. The data recorded from the daily diary device alone is not sufficient to determine the initial mounting location to adjust against, additionally it is not feasible in the field to obtain such data which would require specific measuring devices. We believe that the desirable approach to solve this issue is in the biological domain where an effective mounting protocol is necessary which reduces or removes the risk of such events occurring.
2. **Ecologists manually explore time-series graphs of several attributes to gain an understanding of the mapping from signal to behaviour.** TimeNotes introduces

built for purpose interaction techniques for exploring time-series data. We allow the user to annotate the graph, and bookmark interesting regions identified. As the user gains an understanding of the data they can construct a mental map of the data landscape illustrating the mapping from signal to behaviour by annotating and bookmarking relevant regions, which can be later revisited during analysis. Our user study illustrates that TimeNotes provides a more effective environment for biologists operating on their smart sensor data in comparison to the current state-of-the-art approaches.

3. **Once the biologist has determined the signal to behaviour mapping they will manually decode the data set and label the entire data series with the behaviours of interest.** TimeClassifier leverage's visual analytics for the semi-automatic classification of animal behaviour. A user selects an initial behaviour in the data series, which is then used to search for repeated instances of throughout the series. The user can adjust the similarity to search with, as well as, rejecting and accepting results. This semi-automatic visual analytic approach allows the user to apply their domain knowledge of animal behaviour to identify and label the instances relevant to their research requirements.
4. **Once the biologist has explored and analysed the data it is typical they will seek to present their findings to their research group or to the larger research community through academic journals and conferences.** TimeNotes is built specifically for presentation intents to enable biologists to present and share their findings with other researchers in the community. The TimeNotes work-space used for exploration and analysis can be exported to save the visualisation state. This export can be shared with other researchers who can restore the original work-space created. Fused interaction allows researchers to trace through the annotations and bookmarked regions identified as relevant by the researcher. We also include an option to export to the scalable vector graphics format which produces high-quality scalable images that can be embedded in research publications. While we do not include the presentation aspect in our formal user evaluation, we perform a task based evaluation of the support for presentation intents. In future work, we seek to formally validate these claims through a formal user based study.
5. **There is no software available which binds the four concepts and data attributes proposed by Wilson et al [WLD⁺07] together.** Framework4 enables users to derive the four concepts proposed by Wilson et al. [WLD⁺07]. These are, (i) The derivation of animal behaviour (supported via TimeClassifier) (ii) Animal trajectory to understand animal movement. (iii) Energy expenditure to understand how much energy the animal is using. (iv) Environmental conditions by accessing the data together in tandem to understand the context in which animals operate. We introduce purpose built user interfaces to assist users in selecting appropriate parameters and obtaining direct feedback on the results. We also introduce a new way of combining high resolution motion sensor data with GPS data to create a view of the animal at both the fine scale motor movement and large-scale movement. We present two case studies which show the utility of Framework4 for deriving new insights into human activity and report its usage by over 60 biologists in the field across 12 countries.
6. **Since the data attributes required for analysis are typically computed as separate entities, there is poor support for higher-level exploration and analysis tasks which require data attributes to be bound together.** In section 1.2.1 we outline the typical tasks undertaken by biologists for higher-level exploration and analysis. We focus specifically on one higher-level task, the binding of energetics with movement data to explore behaviour stylisation, comparing between different species performing the same behaviour to optimise or understand the way movement occurs. We introduce a proof-of-concept visualisation called the 'g-sphere'. This concept binds together energetics with movement, allowing multiple dimensions to be visualised simultaneously. We present initial insights which indicate the approach may hold value in obtaining new insights from areas such as sports practice, disease and wild animal ecology. Further work is required to formally validate this approach in the field.

7.2 Future Work - Smart Sensor Data Visualisation

While we validate our hypotheses, and illustrate the application of visualisation coupled with machine learning for efficient and effective smart sensor analysis there are still many challenges faced by biologists when operating on smart sensor data. In this section, we summarise the research questions and challenges identified and arising from the work presented in this thesis.

- **More expressive search for behaviours.** In TimeClassifier we utilise search-by-example for classifying animal behaviour. Normalised cross-correlation is used as the underlying search algorithm, however, this approach does not allow search to be performed for behaviours occurring at different frequencies (e.g. walking). Other techniques such as dynamic time warping are able to extract signals occurring at varying durations, although they impose a higher computational complexity ($O(N^2)$). Further research is required into efficient distance measures to perform search flexibly across different temporal durations as a real-time operation. The implementation of the TimeClassifier framework, would allow such a distance measure to be directly utilised by replacing the normalised cross-correlation implementation. Furthermore, more complex search parameters are required for detecting advanced behaviours which are independent of the shape of the signal produced. In such a scenario, the user needs to search via other means, for example, threshold analysis, or rule based search could be effective where behaviour is determined by amplitude (not shape) or dependant on utilising already identified behaviours (e.g. hunting is always followed by eating). Integrating these search parameters with a decision tree approach (e.g. Wijk et al. [vdEvW11]) could be an interesting research direction which would allow biologists to combine search techniques to formulate advanced queries for the detection of otherwise difficult to extract behaviours.
- **Incorporation of advanced machine learning techniques.** In this thesis we specifically research visual analytics for the machine learning task of classification by applying search to identify behaviour in smart sensor data (TimeClassifier). It is likely that other machine learning methods could assist in this process to make the labelling process more efficient in the hands of end users. The application of machine learning coupled with interactive data visualisation is an active area of research [TKC17]. Applying relevance feedback [KP99] to learn from the rejected / accepted results could assist biologists in obtaining 100% recall and precision more efficiently, as the search space could be reduced with every accept or reject applied to appropriately move other instances exhibiting similar patterns. Clustering could be used to identify an initial starting set of behaviours for input into classification. Similarly, pattern discovery could be applied to assist in drawing conclusions from the data before the biologist undertakes their exploration and analysis tasks. Deep learning could be harnessed to assist in the labelling process, where it has already been used in the computer vision, and audio recognition fields, to in some cases produce superior results to human experts [KSH12]. Furthermore, we anticipate that as the machine learning field evolves, new techniques will become available which learn from few labelled instances.
- **Scalability of visualisation approaches.** As smart sensor data evolves, the duration of deployments and number of sensors which can be utilised in tandem will increase. This advancement will lead to the acquired data getting larger and more complex. In this thesis, we have created methods which are applicable to the current size and complexity of the data available (roughly 10 million data items). As data duration evolves this will increase beyond any data available in the field today. Further research needs to be undertaken to future-proof the methods we present in this thesis. The TimeNotes and g-sphere techniques are inherently constrained to the screen resolution available although operate effectively for the data complexity currently faced. The TimeClassifier algorithm needs to operate in real-time, although it is currently constrained by memory capacity. In such a scenario, out-of-core algorithms would be effective in minimising the effect of large data on available memory capacity, along with investigating applying the algorithms to execute on parallel environments (e.g. GPUs) which could see a reduction in execution time.
- **Integration of TimeNotes and TimeClassifier.** TimeClassifier and TimeNotes operate through separate implementations (TimeNotes associated with visualisation

and interaction, while TimeClassifier introduces a framework for the semi-automatic classification of behaviour through visual analytics). We do not integrate the two methods together, although they are incorporated into the same software suite. In future work, we seek to validate the combination of TimeNotes where we believe it would be of benefit to incorporate our accelerated analytic methods for labelling behaviour from TimeClassifier into the exploration, analysis, and presentation visualisation and interaction techniques which TimeNotes offers.

- **Support for higher-level exploration and analysis.** In section 1.2.1 we outline the typical tasks undertaken by biologists for higher-level exploration and analysis (i.e. on the newly acquired labelled data sequences coupled with the underlying data along with those derived from Framework4 and TimeClassifier). We list some of the higher-order exploration and analysis tasks we observed in the field. We select one of the higher-order tasks to investigate and introduce a proof-of-concept visualisation called the ‘g-sphere’ to explore the combination of energy use coupled with movement over time. This is one facet of the tasks biologists wish to undertake but are limited in their ability to do so without the appropriate visualisations and analytical methods available. In future work, it is important that there is support available for the higher-level tasks (Section 1.2.1), along with further discussions to understand the advanced tasks biologists wish to undertake on their data.
- **Evaluation of G-Sphere.** The g-sphere visualisation technique we introduce is a work-in-progress visualisation which has been created as a proof of concept for supporting a higher-level task biologists sought to undertake. We explore some initial insights, although do not perform a formal user evaluation. We believe these insights have the potential to add value to biological research studies. Therefore, a key area of future research is to evaluate the application of the g-sphere methods with users. Particularly in regard to effectively representing the temporal nature of the data, along with the trade-off of the three-dimensional structure of the visualisation versus the traditional two-dimensional approach using multiple stacked time-series graphs. We expect such an evaluation to give insight and potentially identify fruitful areas of research for evolving the g-sphere method.
- **Dead reckoning accuracy.** The new dead reckoning approach we introduce combines high resolution motion sensor data with GPS data to create a view of the animal at both the fine scale motor movement and large-scale movement. This approach does not include the errors arising from the GPS data utilised, which has its own class of errors (accuracy within approximately 5 metres [BWJ⁺15]). Further work is necessary to include these errors in the algorithm and visualise uncertainty on the trajectory which would be of benefit to the biologists to understand how trustworthy the resulting data is. Furthermore, it would be interesting to research other methods of tracking beyond that of integrating GPS data into the dead reckoning tracks. RFID, and differential GPS are both new methods which could provide a higher level of accuracy. Therefore, further research is necessary to investigate these technologies, understand the trade-off of each, and the accuracy of the underlying trajectories obtained.
- **Open API into Framework4.** Framework4 was introduced to enable biologists to derive the necessary attributes they require to meet their analysis needs. We introduce capabilities to derive the four attributes from Wilson et al. [WLD⁺07]. As biological research evolves, and new sensors are introduced, these methods will likely become out-dated and perhaps no longer relevant, although the presence of a software tool such as Framework4 would still hold value. We therefore see it of benefit to the biological research community that framework4 features an open API for researchers to plug-in the latest research evolving in the field.

7.3 Future Work - Time-Series Visualisation

The study of time-series data is a well-established discipline, with Playfair first introducing the notion of a time-series and the time-series graph in 1786 [Pla86]. Despite this long standing, the data we face today is several orders of magnitude larger in size and complexity which presents challenges requiring new methods for effective exploration, analysis, and

presentation. The literature we explore, and the work presented in this thesis illustrate that the problem of effectively handling time-series data is still an active area of research and generally needs to be explored further. In this section, we outline some of the future research directions in general for the temporal research community:

- **Big data.** The scale of data being generated and collected today is unprecedented. Screen resolution is small in comparison to data storage capacity. When more data items are rendered than the available pixels an over-plotting and occlusion problem occurs where more than one data item is assigned to each pixel, which leads to a loss of information. A purely visual approach of the raw underlying data is often no longer appropriate for this reason. To deal with this data it is essential new approaches are introduced. The application of visual methods linked to automated algorithms is one avenue for future research. Methods of zooming and filtering to abstract away from the underlying data is also of interest for this purpose. Overall, much work needs to be undertaken in this area to prepare and build foundations to future-proof the visualisations and algorithms we utilise. Dimensionality reduction is one option, which can be used to simplify the data-series through an appropriate approximation and therefore reduce the complexity of the data. Alternatively, algorithms which target massively parallel architectures (e.g. GPUs) can provide further assistance in achieving real-time execution.
- **Combination with analytical methods.** Our task based evaluation of visual analytic methods in chapter 2 uncovers gaps in the literature for supporting the knowledge discovery tasks for temporal data. In chapter 4 we introduce TimeClassifier for the classification of temporal data. Further research is necessary in this area to fully support the other knowledge discovery tasks, specifically, classification and clustering, indexing for event data, prediction approaches for multi-variate temporal and event data, and the task of segmentation. Furthermore, in the visual analytic field the global trend has been on applying existing algorithms from the data mining community, we perceive an interesting direction of research for the two fields will be building user-driven algorithms specifically geared for a visual analytics approach. Aside from temporal data, the application of visual analytics across other data domains would be of value to other disciplines. Furthermore, making these methods usable by researchers without computer programming background is an important area. In Framework4, we exposed a user friendly framework for deriving biological data channels from smart sensor data. This could be taken further with an equation builder like interface with standard statistical and mathematical operations exposed to non-programmers.
- **Collaboration with experts.** All of the work we present in this thesis has been conceived with the collaboration we hold with our research partners in the movement ecology community. Without their support, we would not have understood the domain level problems they faced, and therefore have come up with innovative and effective solutions for their needs. Certainly, from a movement ecology viewpoint there are a number of problems which close communication from the visualisation and biology communities can solve. This includes, segmenting movement tracks, applying vertical movement to dead-reckoning to generate three-dimensional movement traces, exposing patterns and trends in movement and behaviour, and discovering interesting patterns and trends between data dimensions. We would also like to see collaboration with domain experts in other areas (e.g. physical sciences and engineering) where we believe the methods we present in this thesis can be applied to assist them in their data analysis.
- **Multi-dimensional visualisation.** Multi-dimensional data provides many challenges [GPL⁺11]. Users typically wish to be able to perceive and identify correlations between dimensions. Effectively mapping data containing a high number of dimensions in a spatially low dimension environment (i.e. the two-dimensions of a computer monitor) to a visual metaphor from which correlations can be identified is challenging. Investigating the combination of data attributes logged by smart sensor devices, in addition to other derived channels (e.g. spatial position) is of great value to biologists (as we demonstrate in chapter 6, where we link together behaviour and energetics). There is a distinct lack of effective techniques available to the movement ecology community to assist them in uncovering insight into animal behaviour. We believe that purpose built

visualisation techniques for this purpose will allow them to extract new knowledge and understanding into animal behaviour.

Appendices

A. USER STUDY TRAINING MATERIAL

User Study

Visual representations for exploring large time-series data

Introduction

- You should try to answer each question for **accuracy** and then **speed**.
- There is no time limit in answering each question, though the time used by you is recorded.

Introduction (continued)

- If possible please try to keep breaks to a minimum during a task.
- You can rest for as long as you like between tasks.
- The software will show how many questions are remaining for the current task in the **top left corner**.

Introduction (continued)

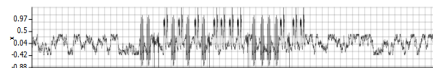
- Please do not discuss your answers with others during the experiment.
- Please **DO NOT** switch off the machine or log off at the end of the session.

HOW TO READ EACH VISUALISATION

Hierarchical Zooming

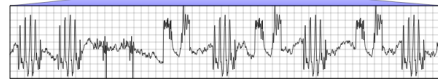
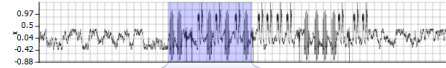
- Method of navigating through large time-series graphs.
- Build a hierarchy of different zoom levels of regions in the data.
- Two visualisation techniques in the study:
 - Stack zoom
 - Bookmark zoom

How to read a bookmark visualization



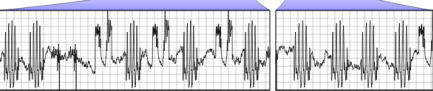
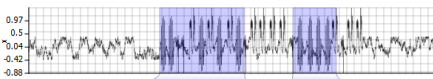
- Start off with the whole time-series graph plotted
- User can select regions of interest which are zoomed in...

How to read a bookmark visualization



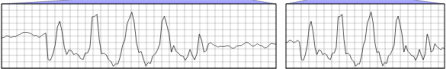
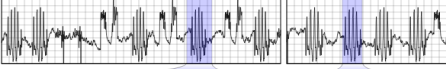
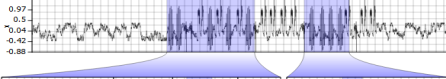
- Here the region of interest is focused in the lower view

How to read a bookmark visualization



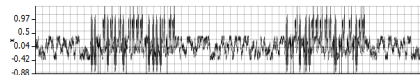
- A hierarchy of zoom levels is constructed by further selection.

How to read a bookmark visualization



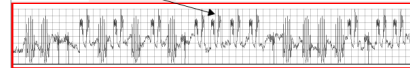
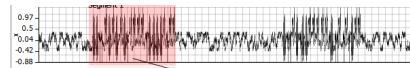
- Navigate through the hierarchy using the connections

How to read a stack zoom visualization



- Again, the whole time-series graph is plotted
- User can select regions of interest which are zoomed in...

How to read a stack zoom visualization



- A region of interest is added which is shown below the root. Arrows show the connection between the parent location.

How to read a stack zoom visualization

How to read a stack zoom visualization

- Navigate through the hierarchy using the arrows.

TERMINOLOGY

Signal Amplitude

- Amplitude is a measure of the change between the highest value and lowest value.

Signal frequency

- Frequency is a measure to determine how quickly a signal goes through a cycle.

Leaf Node

- A leaf node is a level with no further child zoom levels attached.

TASKS

Task overview

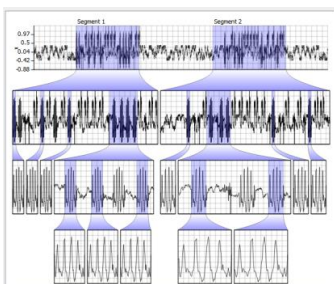
- The user study comprises of five tasks:
 - **Task One:** Count the number of leaf nodes.
 - **Task Two:** Identify the signal with the largest amplitude.
 - **Task Three:** Identify the signal with the highest frequency.
 - **Task Four:** Label a series of signals in the data set.
 - **Task Five:** Count the number of labelled signals.

Task One: Count the number of leaf nodes

Task One

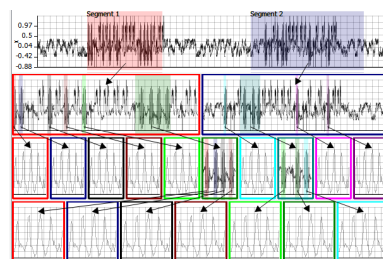
- Your screen will show you a box containing either a stack zoom or bookmark plot visualisation.
- Your task here is to identify the number of leaf nodes each highlighted segment contains for each visualisation. Once you have done this please enter your answer into the text box and then click **NEXT**.

Bookmark Plot

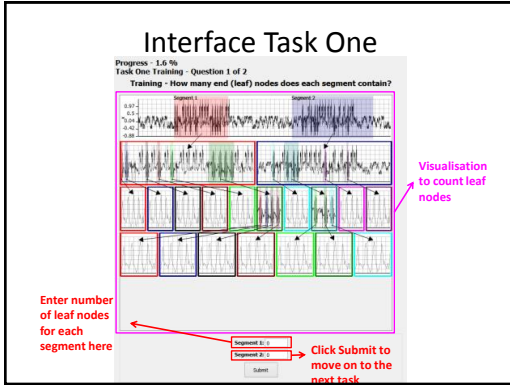


- How many leaf nodes for segment 1? Segment 2?

Stack Zoom

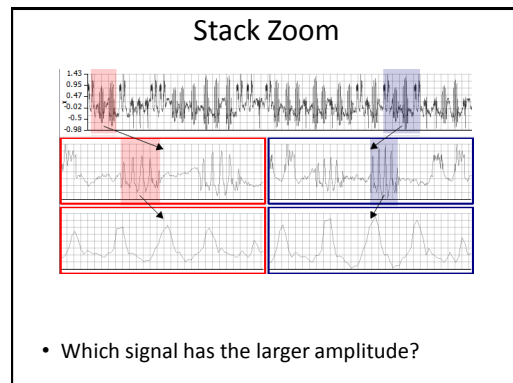
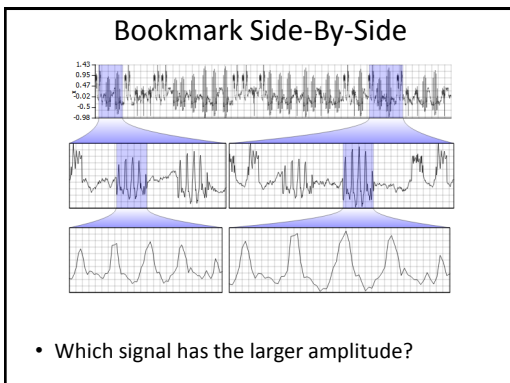
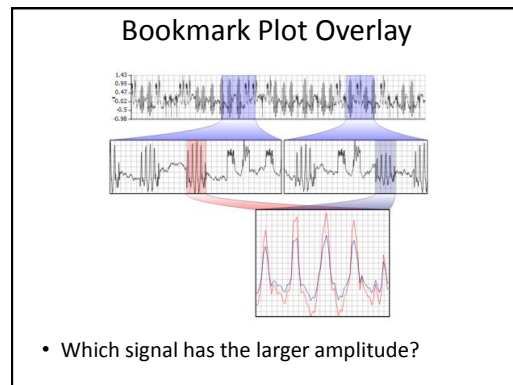


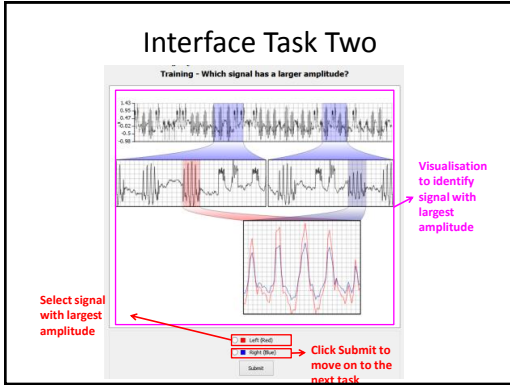
- How many leaf nodes for segment 1? Segment 2?



Task Two: Identify the signal with the largest amplitude

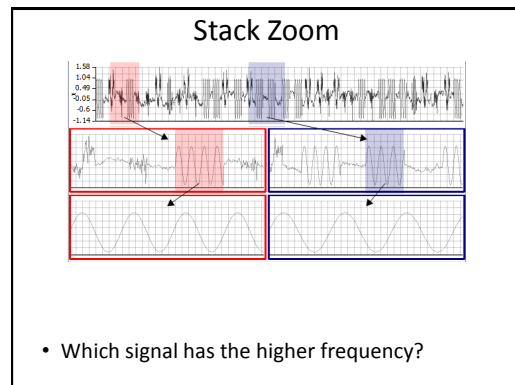
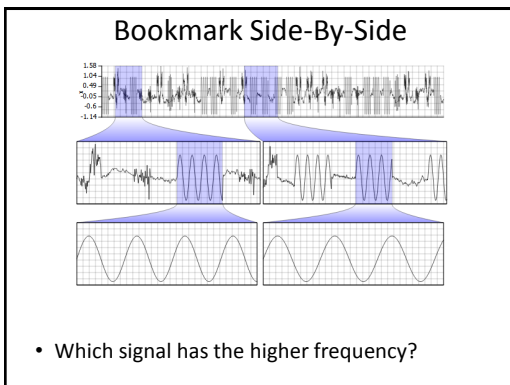
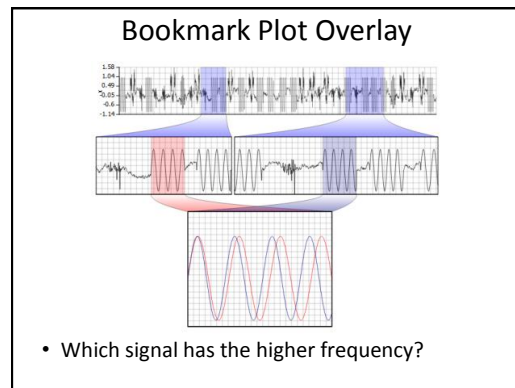
- ### Task Two
- Your screen will show you a box containing either a stack zoom, bookmark plot or overlay visualisation.
 - Your task here is to identify which of two signals has the larger amplitude. Once you have done this please select your answer and then click **NEXT**.

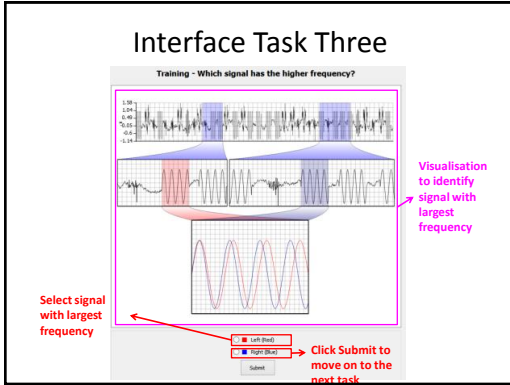




Task Three: Identify the signal with the highest frequency

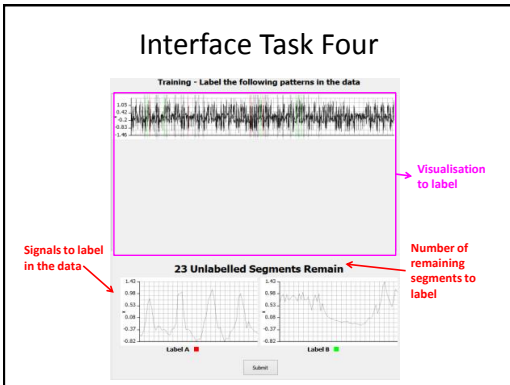
- ### Task Three
- Your screen will show you a box containing either a stack zoom, bookmark plot or overlay visualisation.
 - Your task here is to identify which of two signals has the larger frequency. Once you have done this please select your answer and then click NEXT.





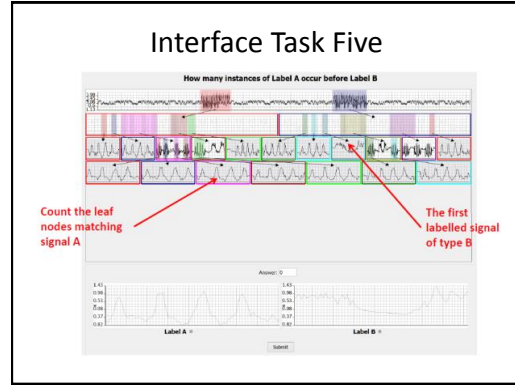
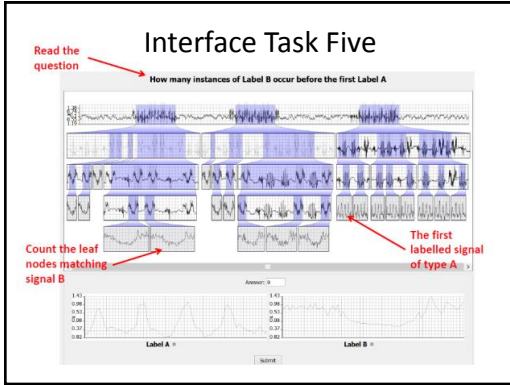
Task Four: Label a series of signals in the data set

- ### Task Four
- Your screen will show you a box containing either a stack zoom or bookmark plot visualisation containing a number of overlaid grey segments.
 - Your task here is to find and assign a label to as many grey segments as possible. Once you have done this **click NEXT**.



Task Five: Count the number of labelled signals

- ### Task Five
- Your screen will show you a box containing either a stack zoom or bookmark plot visualisation containing a number of leaf nodes that match label A or label B.
 - Your task here is to read the question and count how many labels of one type are before the first occurrence of the second type. Once you have done this **click NEXT**.



BIBLIOGRAPHY

- [AA05] Natalia Andrienko and Gennady Andrienko. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [AAD⁺10] Gennady Andrienko, Natalia Andrienko, Urska Demsar, Doris Dransch, Jason Dykes, Sara Irina Fabrikant, Mikael Jern, Menno-Jan Kraak, Heidrun Schumann, and Christian Tominski. Space, time and visual analytics. *International Journal of Geographical Information Science*, 24(10):1577–1600, 2010.
- [AAMHK12] A. Abdulla-Al-Maruf, Hung-Hsuan Huang, and K. Kawagoe. Time series classification method based on longest common subsequence and textual approximation. In *Digital Information Management (ICDIM), 2012 Seventh International Conference on*, pages 130–137, Aug 2012.
- [ABG⁺14] Bilal Alsallakh, Markus Bögl, Theresia Gschwandtner, S. Miksch, Bilal Esmal, Arghad Arnaout, Gerhard Thonhauser, and Philipp Zöllner. A visual analytics approach to segmenting and labeling multivariate time series data. In Margit Pohl and Jonathan C. Roberts, editors, *EuroVis Workshop on Visual Analytics (EuroVA)*, pages 31–35. Eurographics, Eurographics, 06/2014 2014.
- [ACMR00] Catherine M Arrington, Thomas H Carr, Andrew R Mayer, and Stephen M Rao. Neural mechanisms of visual attention: object-based selection of a region in space. *Cognitive Neuroscience, Journal of*, 12(Supplement 2):106–117, 2000.
- [Act11] Start Active Stay Active. A report on physical activity for health from the four home countries chief medical officers. *The Department of Health*, 2011.
- [AEEK99] Mihael Ankerst, Christian Elsen, Martin Ester, and Hans-Peter Kriegel. Visual classification: an interactive approach to decision tree construction. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 392–396. ACM, 1999.
- [AHWS06] JRS Arch, D Hislop, SJY Wang, and JR Speakman. Some mathematical and technical issues in the measurement and interpretation of open-circuit indirect calorimetry in small animals. *International journal of obesity*, 30(9):1322–1331, 2006.
- [Ale05] R McNeill Alexander. Models and the scaling of energy costs for locomotion. *Journal of Experimental Biology*, 208(9):1645–1652, 2005.
- [AMA07] D. Archambault, T. Munzner, and D. Auber. Topolayout: Multilevel graph layout by topological features. *Visualization and Computer Graphics, IEEE Transactions on*, 13(2):305–317, March 2007.
- [AMKP04] Chadia Abras, Diane Maloney-Krichmar, and Jenny Preece. User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*, 37(4):445–456, 2004.
- [AMM⁺07] Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. Visualizing time-oriented dataa systematic view. *Computers & Graphics*, 31(3):401–409, 2007.
- [AMST11] Wolfgang Aigner, Silvia Miksch, Heidrun Schuman, and C. Tominski. *Visualization of Time-Oriented Data*. Human-Computer Interaction. Springer Verlag, 1st edition, 2011.
- [AO01] Cláudia M Antunes and Arlindo L Oliveira. Temporal data mining: An overview. In *KDD Workshop on Temporal Data Mining*, pages 1–13, 2001.

-
- [ASJM⁺08] Natacha Aguilar Soto, Mark P Johnson, Peter T Madsen, Francisca Díaz, Iván Domínguez, Alberto Brito, and Peter Tyack. Cheetahs of the deep sea: deep foraging sprints in short-finned pilot whales off tenerife (canary islands). *Journal of Animal Ecology*, 77(5):936–947, 2008.
- [BAA⁺14] OR Bidder, O Arandjelović, F Almutairi, ELC Shepard, SA Lambertucci, LA Qasem, and RP Wilson. A risky business or a safe bet? a fuzzy set event tree for estimating hazard in biotelemetry studies. *Animal Behaviour*, 93:143–150, 2014.
- [BAF⁺13] Markus Bogl, Wolfgang Aigner, Peter Filzmoser, Tim Lammarsch, Silvia Miksch, and Alexander Rind. Visual analytics for model selection in time series analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2237–2246, 2013.
- [BAF⁺14] Markus Bögl, Wolfgang Aigner, Peter Filzmoser, Theresia Gschwandtner, Tim Lammarsch, S. Miksch, and Alexander Rind. Visual analytics methods to guide diagnostics for time series model predictions. In Adam Perer, Enrico Bertini, Ross Maciejewski, and Jimeng Sun, editors, *Proceedings of the 2014 IEEE VIS Workshop on Visualization for Predictive Analytics*, 2014.
- [BAF⁺15] Markus Bögl, Wolfgang Aigner, Peter Filzmoser, Theresia Gschwandtner, Tim Lammarsch, Silvia Miksch, and Alexander Rind. Integrating predictions in time series model selection. In Enrico Bertini and Jonathan C. Roberts, editors, *Proceedings of the EuroVis Workshop on Visual Analytic, EuroVA*, pages 73–77. EuroGraphics, EuroGraphics, 2015.
- [BAG⁺15] Peter Bodesinsky, Bilal Alsallakh, Theresia Gschwandtner, Silvia Miksch, Enrico Bertini, and Jonathan C Roberts. Exploration and assessment of event data. In *Proceedings of the Sixth International EuroVis Workshop on Visual Analytics (EuroVA) 2015*. Eurographics Association, 2015.
- [BAP⁺05] Paolo Buono, Aleks Aris, Catherine Plaisant, Amir Khella, and Ben Shneiderman. Interactive pattern search in time series. In *Electronic Imaging 2005*, pages 175–186. International Society for Optics and Photonics, 2005.
- [Bar06] Roger Bartlett. Artificial intelligence in sports biomechanics: New dawn or false hope? *Journal of sports science & medicine*, 5(4):474, 2006.
- [Bat14] Brian Battaile. TrackReconstruction: Reconstruct animal tracks from magnetometer, accelerometer, depth and optional speed data. <https://cran.r-project.org/web/packages/TrackReconstruction/index.html>, 2014. [Online; accessed 14-November-2015].
- [BBG⁺09] Jorik Blaas, Charl P. Botha, Edward Grundy, Mark W. Jones, Robert S. Laramee, and Frits H. Post. Smooth graphs for visual exploration of higher-order state transitions. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):969–976, 2009.
- [BBTLB13] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Al Le Borgne. Machine learning strategies for time series forecasting. In Marie-Aude Aufaure and Esteban Zimnyi, editors, *Business Intelligence*, volume 138 of *Lecture Notes in Business Information Processing*, pages 62–77. Springer Berlin Heidelberg, 2013.
- [BC87] Richard A Becker and William S Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.
- [BCGL⁺14] Owen R Bidder, Hamish A Campbell, Agustina Gómez-Laich, Patricia Urgé, James Walker, Yuzhi Cai, Lianli Gao, Flavio Quintana, and Rory P Wilson. Love thy neighbour: automatic animal behavioural classification of acceleration data using the k-nearest neighbour algorithm. *PloS one*, 9(2):88609, 2014.
- [BD13] Peter J Brockwell and Richard A Davis. *Time series: theory and methods*. Springer Science & Business Media, 2013.

-
- [BDF08] Luca Börger, Benjamin D Dalziel, and John M Fryxell. Are there general mechanisms of animal home range behaviour? a review and prospects for future research. *Ecology letters*, 11(6):637–650, 2008.
- [Ber67] J Bertin. 1983 semiology of graphics: Diagrams, networks, maps, 1967.
- [Ber06] Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.
- [BGD94] Frederick J Bremner, Stephen J Gotts, and Dina L Denham. Hinton diagrams: Viewing connection strengths in neural networks. *Behavior Research Methods, Instruments, & Computers*, 26(2):215–218, 1994.
- [BHSM04] Corey JA Bradshaw, Mark A Hindell, Michael D Sumner, and Kelvin J Michael. Loyalty pays: potential life history consequences of fidelity to marine foraging regions by southern elephant seals. *Animal Behaviour*, 68(6):1349–1360, 2004.
- [BKW11] SN Bleich, R Ku, and YC Wang. Relative contribution of energy intake and energy expenditure to childhood obesity: a review of the literature and directions for future research. *International journal of obesity*, 35(1):1–15, 2011.
- [BKW⁺14] Danielle D Brown, Roland Kays, Martin Wikelski, Rory P Wilson, and A Peter Klimley. *Observing the unwatchable through acceleration logging of animal behavior*. Bibliothek der Universität Konstanz, 2014.
- [BM04] L. Berry and T. Munzner. Binx: Dynamic exploration of time series datasets across aggregation levels. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages p2–p2, Oct 2004.
- [BPS⁺07] P. Buono, C. Plaisant, A. Simeone, A. Aris, B. Shneiderman, G. Shmueli, and W. Jank. Similarity-based forecasting with simultaneous previews: A river plot interface for time series forecasting. In *Information Visualization, 2007. IV '07. 11th International Conference*, pages 191–196, July 2007.
- [BQW12] Owen R Bidder, Lama A Qasem, and Rory P Wilson. On higher ground: How well can dynamic body acceleration determine speed in variable terrain? *PloS one*, 7(11):e50556, 2012.
- [Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [BRG⁺12] Jrgen Bernard, Tobias Ruppert, Oliver Goroll, Thorsten May, and Jrn Kohlhammer. Visual-interactive preprocessing of time series data. In Andreas Kerren and Stefan Seipel, editors, *SIGRAD*, volume 81 of *Linkping Electronic Conference Proceedings*, pages 39–48. Linkping University Electronic Press, 2012.
- [BS90] EJ Bassegy and AH Short. A new method for measuring power output in a single leg extension: feasibility, reliability and validity. *European Journal of Applied Physiology and Occupational Physiology*, 60(5):385–390, 1990.
- [BS06] Jasmin Blanchette and Mark Summerfield. *C++ GUI programming with Qt 4*. Prentice Hall Professional, 2006.
- [BSS⁺12] Owen R Bidder, Marion Soresina, Emily LC Shepard, Lewis G Halsey, Flavio Quintana, Agustina Gómez-Laich, and Rory P Wilson. The need for speed: testing acceleration for estimating animal travel rates in terrestrial dead-reckoning systems. *Zoology*, 115(1):58–64, 2012.
- [BWJ⁺15] O. R. Bidder, J. S. Walker, M. W. Jones, M. D. Holton, P. Urge, D. M. Scantlebury, N. J. Marks, E. A. Magowan, I. E. Maguire, and R. P. Wilson. Step by step: reconstruction of terrestrial animal movement paths by dead-reckoning. *Movement Ecology*, 3(1):1–16, September 2015.
- [CAM01] Junping Cai, Niels L Andersen, and Cristian Malureanu. In-field practical calibration of three-axis magnetometers. In *Proceedings of the 2010 International Technical Meeting of The Institute of Navigation*, pages 67–73, 2001.

-
- [Car00] Michael J Caruso. Applications of magnetic sensors for low cost compass systems. In *Position Location and Navigation Symposium, IEEE 2000*, pages 177–184. IEEE, 2000.
- [CC86] Guido Chelazzi and Roberto Calzolari. Thermal benefits from familiarity with the environment in a reptile. *Oecologia*, 68(4):557–558, 1986.
- [cF11] Tak chung Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164 – 181, 2011.
- [CGK⁺07] Remco Chang, Mohammad Ghoniem, Robert Kosara, William Ribarsky, Jing Yang, Evan Suma, Caroline Ziemkiewicz, Daniel Kern, and Agus Sudjianto. Wirevis: Visualization of categorical, time-varying data from financial transactions. In *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*, pages 155–162. IEEE, 2007.
- [CISSW06] Chaomei Chen, Fidelia Ibekwe-SanJuan, Eric SanJuan, and Chris Weaver. Visual analysis of conflicting opinions. In *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, pages 59–66. IEEE, 2006.
- [Coo05] Lynne Cooke. Eye tracking: How it works and how it relates to usability. *Technical Communication*, 52(4):456–463, 2005.
- [Cor12] John Corbett. Charles joseph minard: Mapping napoleons march, 1861, 2012.
- [CPSK10] K.J. Cios, W. Pedrycz, R.W. Swiniarski, and L. Kurgan. *Data Mining: A Knowledge Discovery Approach*. Springer US, 2010.
- [CSP⁺14] Alessandro Camerra, Jin Shieh, Themis Palpanas, Thanawin Rakthanmanon, and Eamonn Keogh. Beyond one billion time series: indexing and mining very large time series collections with isax2+. *Knowledge and information systems*, 39(1):123–151, 2014.
- [CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [DAW13] Tuan Nhon Dang, A. Anand, and L. Wilkinson. Timeseer: Scagnostics for high-dimensional time series. *Visualization and Computer Graphics, IEEE Transactions on*, 19(3):470–483, March 2013.
- [Dix09] Alan Dix. *Human-computer interaction*. Springer, 2009.
- [DNKS10] Nicholas Diakopoulos, Mor Naaman, and Funda Kivran-Swaine. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 115–122. IEEE, 2010.
- [DOL03] Maria Cristina Ferreira De Oliveira and Haim Levkowitz. From visual data exploration to visual data mining: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 9(3):378–394, 2003.
- [EA12] Philippe Esling and Carlos Agon. Time-series data mining. *ACM Comput. Surv.*, 45(1):12:1–12:34, December 2012.
- [ELVK⁺12] Kyle H. Elliott, Maryline Le Vaillant, Akiko Kato, John R. Speakman, and Yan Ropert-Coudert. Accelerometry predicts daily energy expenditure in a bird with high activity levels. *Biology Letters*, 9(1), 2012.
- [Eve06] Brian S Everitt. *The Cambridge dictionary of statistics*. Cambridge University Press, 2006.
- [Far14] Gerald Farin. *Curves and surfaces for computer-aided geometric design: a practical guide*. Elsevier, 2014.
- [FKSS06] Jerry Alan Fails, Amy Karlson, Layla Shahamat, and Ben Shneiderman. A visual interface for multivariate temporal data: Finding patterns of events across multiple histories. In *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, pages 167–174. IEEE, 2006.

-
- [FSA99] Yoav Freund, Robert Schapire, and N Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.
- [GAA04] Peter Gatala, Natalia Andrienko, and Gennady Andrienko. Interactive analysis of event data using space-time cube. In *Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on*, pages 145–152. IEEE, 2004.
- [GAM⁺14] Theresia Gschwandtner, Wolfgang Aigner, S. Miksch, Johannes Gärtner, Simone Kriglstein, Margit Pohl, and Nikolaus Suchy. Timecleanser: A visual analytics approach for data cleansing of time-oriented data. In Stefanie Lindstaedt, Michael Granitzer, and Harald Sack, editors, *14th International Conference on Knowledge Technologies and Data-driven Business (i-KNOW 2014)*, pages 1–8. ACM Press, ACM Press, 2014.
- [GCML06] Diansheng Guo, Jin Chen, Alan M MacEachren, and Ke Liao. A visualization system for space-time and multivariate patterns (vis-stamp). *IEEE transactions on visualization and computer graphics*, 12(6):1461–1474, 2006.
- [GGW13] Michael Glueck, Tovi Grossman, and Daniel Wigdor. A model of navigation for very large data views. In *Proceedings of Graphics Interface 2013*, GI '13, pages 9–16, Toronto, Ont., Canada, Canada, 2013. Canadian Information Processing Society.
- [GHW09] JA Green, Lewis G Halsey, RP Wilson, and PB Frappell. Estimating energy expenditure of animals using the accelerometry technique: activity, inactivity and comparison with the heart-rate technique. *Journal of Experimental Biology*, 212(4):471–482, 2009.
- [GJG⁺15] Samah Gad, Waqas Javed, Sohaib Ghani, Niklas Elmqvist, Tom Ewing, Keith N Hampton, and Naren Ramakrishnan. Themedelta: Dynamic segmentations over temporal topic models. *Visualization and Computer Graphics, IEEE Transactions on*, 21(5):672–685, 2015.
- [GJL⁺09a] Ed Grundy, Mark W. Jones, Robert S. Laramee, Rory P. Wilson, and Emily F. Shepard. Visualization of sensor data from animal movement. *Eurographics/IEEE-VGTC Symposium on Visualization (Eurovis) 2009, Computer Graphics Forum*, 28(2):815–822, June 2009.
- [GJL⁺09b] Edward Grundy, Mark W. Jones, Robert S. Laramee, Rory P. Wilson, and Emily L. C. Shepard. Visualisation of sensor data from animal movement. *Comput. Graph. Forum*, 28(3):815–822, 2009.
- [GMS⁺13] O Girard, GP Millet, J Slawinski, S Racinais, and JP Micallef. Changes in running mechanics and spring-mass behaviour during a 5-km time trial. *Int J Sports Med*, 34(09):832–840, 2013.
- [GPL⁺11] Zhao Geng, ZhenMin Peng, Robert S Laramee, Jonathan C Roberts, and Rick Walker. Angular histograms: Frequency-based visualizations for large, high dimensional data. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2572–2580, 2011.
- [GS12] Machon Gregory and Ben Shneiderman. Shape identification in temporal data sets. In *Expanding the Frontiers of Visual Analytics and Visualization*, pages 305–321. Springer, 2012.
- [GSB⁺15] Theresia Gschwandtner, Heidrun Schuman, Jürgen Bernard, Thorsten May, Markus Bögl, Silvia Miksch, Jörn Kohlhammer, Martin Röhlig, and Bilal Alsalkh. Enhancing time series segmentation and labeling through the knowledge generation model. In Ross Maciejewski and Fabio Marton, editors, *Poster Proceedings of the Eurographics Conference on Visualization (EuroVis 2015)*, page 3. The Eurographics Association, The Eurographics Association, 2015.
- [GWS11] Adrian C Gleiss, Rory P Wilson, and Emily LC Shepard. Making overall dynamic body acceleration work: on the theory of acceleration as a proxy for energy expenditure. *Methods in Ecology and Evolution*, 2(1):23–33, 2011.

-
- [Häg70] Torsten Hägerstrand. What about people in regional science? *Papers in regional science*, 24(1):7–24, 1970.
- [Ham70] William D Hamilton. Selfish and spiteful behaviour in an evolutionary model. 1970.
- [HBAO⁺01] Sascha K Hooker, Robin W Baird, S Al-Omari, S Gowans, and Hal Whitehead. Behavioral reactions of northern bottlenose whales (*hyperoodon ampullatus*) to biopsy darting and tag attachment procedures. *Fishery Bulletin*, 99(2):303–308, 2001.
- [HBB⁺11] Lucy A Hawkes, Sivananinthaperumal Balachandran, Nyambayar Batbayar, Patrick J Butler, Peter B Frappell, William K Milsom, Natsagdorj Tseveenmyadag, Scott H Newman, Graham R Scott, Ponnusamy Sathiyaselvam, et al. The trans-himalayan flights of bar-headed geese (*anser indicus*). *Proceedings of the National Academy of Sciences*, 108(23):9516–9519, 2011.
- [HBH07] LG Halsey, C-A Bost, and Yves Handrich. A thorough and quantified method for classifying seabird diving behaviour. *Polar Biology*, 30(8):991–1004, 2007.
- [HDK⁺07] Ming C Hao, Umeshwar Dayal, Daniel Keim, Dominik Morent, Joern Schneidewind, et al. Intelligent visual analytics queries. In *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*, pages 91–98. IEEE, 2007.
- [HDKS07] Ming Hao, Umeshwar Dayal, Daniel Keim, and Tobias Schreck. Multi-Resolution Techniques for Visual Exploration of Large Time-Series Data. In K. Museth, T. Moeller, and A. Ynnerman, editors, *Eurographics/ IEEE-VGTC Symposium on Visualization*. The Eurographics Association, 2007.
- [HF09] Christian Holz and Steven Feiner. Relaxed selection techniques for querying time-series graphs. In *UIST '09: Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 213–222, New York, NY, USA, 2009. ACM.
- [HJM⁺11] Ming C Hao, Halldor Janetzko, Sebastian Mittelstädt, Water Hill, Umeshwar Dayal, Daniel A Keim, Manish Marwah, and Ratnesh K Sharma. A visual analytics approach for peak-preserving prediction of large seasonal time series. In *Computer Graphics Forum*, volume 30, pages 691–700. Wiley Online Library, 2011.
- [HMJ⁺12] Ming C. Hao, Manish Marwah, Halldór Janetzko, Umeshwar Dayal, Daniel A. Keim, Debprakash Patnaik, Naren Ramakrishnan, and Ratnesh K. Sharma. Visual exploration of frequent patterns in multivariate time series. *Information Visualization*, 11(1):71–83, 2012.
- [HS04] Harry Hochheiser and Ben Shneiderman. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, March 2004.
- [HSCW13] Steffen Hadlak, Heidrun Schumann, Clemens H Cap, and Till Wollenberg. Supporting the visual analysis of dynamic networks by clustering associated temporal attributes. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2267–2276, 2013.
- [HSQ⁺09] Lewis G Halsey, ELC Shepard, F Quintana, A Gomez Laich, JA Green, and RP Wilson. The relationship between oxygen consumption and body acceleration in a range of species. *Comparative Biochemistry and Physiology Part A: Molecular & Integrative Physiology*, 152(2):197–202, 2009.
- [Hsu07] Wynne Hsu. *Temporal and spatio-temporal data mining*. IGI Global, 2007.
- [HSW11] Lewis G Halsey, Emily LC Shepard, and Rory P Wilson. Assessing the development and application of the accelerometry technique for estimating energy expenditure. *Comparative Biochemistry and Physiology Part A: Molecular & Integrative Physiology*, 158(3):305–314, 2011.

-
- [IBDF11] Petra Isenberg, Anastasia Bezerianos, Pierre Dragicevic, and Jean-Daniel Fekete. A study on dual-scale data charts. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2469–2478, 2011.
- [IDDB94] P loalè, P Dall’Antonia, L Dall’Antonia, and S Benvenuti. Flight paths of homing pigeons studied by means of a direction recorder. *Ethology ecology & evolution*, 6(4):519–527, 1994.
- [ISS04] Fidelia Ibekwe-Sanjuan and Eric Sanjuan. Mining textual data through term variant clustering: the termwatch system. In *Recherche d’Information et ses Applications (RIAO 2004)*, pages 487–503. FID, 2004.
- [JBMS09] Heike Janicke, Michael Bottinger, Uwe Mikolajewicz, and Gerik Scheuermann. Visual exploration of climate variability changes using wavelet analysis. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1375–1382, November 2009.
- [JE10] W. Javed and N. Elmqvist. Stack zooming for multi-focus interaction in time-series data visualization. In *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, pages 33–40, March 2010.
- [JE13] Waqas Javed and Niklas Elmqvist. Stack zooming for multifocus interaction in skewed-aspect visual spaces. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1362–1374, August 2013.
- [JLG⁺12] Hans-Christian Jetter, Svenja Leifert, Jens Gerken, Sören Schubert, and Harald Reiterer. Does (multi-)touch aid users’ spatial memory and navigation in ‘panning’ and in ‘zooming & panning’ UIs? In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 83–90, 2012.
- [JME10] Waqas Javed, Bryan McDonnell, and Niklas Elmqvist. Graphical perception of multiple time series. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):927–934, 2010.
- [JMF03] Ian D Jonsen, Ransom A Myers, and Joanna Mills Flemming. Meta-analysis of animal movement using state-space models. *Ecology*, 84(11):3055–3063, 2003.
- [JS09] Jing Jin and Pedro Szekely. Querymarvel: a visual query language for temporal patterns using comic strips. In *Visual Languages and Human-Centric Computing, 2009. VL/HCC 2009. IEEE Symposium on*, pages 207–214. IEEE, 2009.
- [JS10] Jing Jin and Pedro Szekely. Interactive querying of temporal data using a comic strip metaphor. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 163–170. IEEE, 2010.
- [JSOF11] Mario Jovanovic, Goran Sporis, Darija Omrcen, and Fredi Fiorentini. Effects of speed, agility, quickness training method on power performance in elite soccer players. *The Journal of Strength & Conditioning Research*, 25(5):1285–1292, 2011.
- [JT03] Mark P Johnson and Peter L Tyack. A digital acoustic recording tag for measuring the response of wild marine mammals to sound. *Oceanic Engineering, IEEE Journal of*, 28(1):3–12, 2003.
- [KBH04] Robert Kosara, Fabian Bendix, and Helwig Hauser. Time histograms for large, time-dependent data. In *Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization*, pages 45–54. Eurographics Association, 2004.
- [KBK⁺11] Miloš Krstajić, Enrico Bertini, Daniel Keim, et al. Cloudlines: Compact display of event episodes in multiple time-series. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2432–2439, 2011.

-
- [KCHP04] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. Segmenting time series: A survey and novel approach. *Data mining in time series databases*, 57:1–22, 2004.
- [Kei02] D.A. Keim. Information visualization and visual data mining. *Visualization and Computer Graphics, IEEE Transactions on*, 8(1):1–8, Jan 2002.
- [Keo06] Eamonn Keogh. A decade of progress in indexing and mining large time series databases. In *Proceedings of the 32nd international conference on Very large data bases*, pages 1268–1268. VLDB Endowment, 2006.
- [Kin10] R. Kincaid. Signallens: Focus+context applied to electronic time series. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):900–907, Nov 2010.
- [KK03] Eamonn Keogh and Shruti Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Mining and Knowledge Discovery*, 7(4):349–371, October 2003.
- [KKA95] D. A. Keim, H. P. Kriegel, and M. Ankerst. Recursive pattern: a technique for visualizing very large amounts of data. In *Visualization, 1995. Visualization '95. Proceedings., IEEE Conference on*, pages 279–286, 463, Oct 1995.
- [KKEM10] Daniel A. Keim, Joern Kohlhammer, Geoffrey Ellis, and Florian Mansmann, editors. *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics, November 2010.
- [KL06] Robert Kincaid and Heidi Lam. Line graph explorer: scalable display of line graphs using focus+ context. In *Proceedings of the working conference on Advanced visual interfaces*, pages 404–411. ACM, 2006.
- [KMJE12] Sungahn Ko, Ross Maciejewski, Yun Jang, and David S Ebert. Marketanalyzer: an interactive visual analytics system for analyzing competitive advantage using point of sale data. In *Computer Graphics Forum*, volume 31, pages 1245–1254. Wiley Online Library, 2012.
- [KMS⁺08] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. *Visual analytics: Scope and challenges*. Springer, 2008.
- [KP99] Eamonn J. Keogh and Michael J. Pazzani. Relevance feedback retrieval of time series data. In *Proceedings of the 22Th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 183–190, 1999.
- [KR05a] Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and information systems*, 7(3):358–386, 2005.
- [KR05b] Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowl. Inf. Syst.*, 7(3):358–386, March 2005.
- [Kra03] Menno-Jan Kraak. The space-time cube revisited from a geovisualization perspective. In *Proc. 21st International Cartographic Conference*, pages 1988–1996, 2003.
- [KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [KSS04] Daniel A Keim, Jörn Schneidewind, and Mike Sips. Circleview: a new approach for visualizing time-related multidimensional data sets. In *Proceedings of the working conference on Advanced visual interfaces*, pages 179–182. ACM, 2004.
- [KW05] Thomas Kapler and William Wright. Geotime information visualization. *Information Visualization*, 4(2):136–146, 2005.

-
- [KWS⁺15] Patrick Köthür, Carl Witt, Mike Sips, Norbert Marwan, Stefan Schinkel, and Doris Dransch. Visual analytics for correlation-based comparison of time series ensembles. In *Computer Graphics Forum*, volume 34, pages 411–420. Wiley Online Library, 2015.
- [LAB⁺14] Tim Lammarsch, Wolfgang Aigner, Alessio Bertone, Silvia Miksch, and Alexander Rind. Mind the time: Unleashing temporal aspects in pattern discovery. *Computers & Graphics*, 38:38–50, 2014.
- [LCP⁺13] Philip Legg, David HS Chung, Matthew L Parry, Rhodri Bown, Mark W Jones, Iwan W Griffiths, Min Chen, et al. Transformation of an uncertain video search pipeline to a sketch-based visual analytics loop. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2109–2118, 2013.
- [Lew95] J. P. Lewis. Fast template matching. *Vision Interface*, 95:120–123, 1995.
- [LKL05] Jessica Lin, Eamonn Keogh, and Stefano Lonardi. Visualizing and discovering non-trivial patterns in large time series databases. *Information Visualization*, 4(2):61–82, July 2005.
- [LKLC03] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11. ACM, 2003.
- [LLO12] Yuan Li, Jessica Lin, and Tim Oates. Visualizing variable-length time series motifs. In *SDM*, pages 895–906. SIAM / Ompress, 2012.
- [LMT⁺12] Michael J. Lanzone, Tricia A. Miller, Philip Turk, David Brandes, Casey Halverson, Charles Maisonneuve, Junior Tremblay, Jeff Cooper, Kieran O’Malley, Robert P. Brooks, and Todd Katzner. Flight responses by a migratory soaring raptor to changing meteorological conditions. *Biology Letters*, 8(5):710–713, 2012.
- [LP02] Jessica Lin Eamonn Keogh Stefano Lonardi and Pranav Patel. Finding motifs in time series. In *Proc. of the 2nd Workshop on Temporal Data Mining*, pages 53–68, 2002.
- [LS06] Srivatsan Laxman and P.S. Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, 2006.
- [LSS09] Zhicheng Liu, John Stasko, and Timothy Sullivan. Selltrend: Inter-attribute visual analysis of temporal transaction data. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1025–1032, 2009.
- [Mac95] AM MacEachren. *How maps work: representation, visualization, and design*. Guilford Press, 1995.
- [Mar72] Alexander Marshack. *The roots of civilization: A study in prehistoric cognition; the origins of art, symbol and notation*, 1972.
- [MDB87] Bruce Howard McCormick, Thomas A DeFanti, and Maxine D Brown. *Visualization in scientific computing*, 1987.
- [Mit10] Theophano Mitsa. *Temporal data mining*. CRC Press, 2010.
- [MJS⁺09] Paula Martiskainen, Mikko Järvinen, Jukka-Pekka Skön, Jarkko Tiirikainen, Mikko Kolehmainen, and Jaakko Mononen. Cow behaviour pattern recognition using a three-dimensional accelerometer and support vector machines. *Applied Animal Behaviour Science*, 119(1):32–38, 2009.
- [MJT04] Patrick JO Miller, Mark P Johnson, and Peter L Tyack. Sperm whale behaviour indicates the use of echolocation click buzzes creaks in prey capture. *Proceedings of the Royal Society of London B: Biological Sciences*, 271(1554):2239–2247, 2004.

-
- [MMKN08] Peter McLachlan, Tamara Munzner, Eleftherios Koutsofios, and Stephen North. LiveRAC: Interactive visual exploration of system management time-series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1483–1492, 2008.
- [MSI+03] Yoko Mitani, Katsufumi Sato, Shinichiro Ito, Michael F Cameron, Donald B Siniff, and Yasuhiko Naito. A method for reconstructing three-dimensional dive profiles of marine mammals using geomagnetic intensity data: results from two lactating weddell seals. *Polar Biology*, 26(5):311–317, 2003.
- [MTJ+07] Ross Maciejewski, Benjamin Tyner, Yun Jang, Cheng Zheng, Rimma V Nehme, David S Ebert, William S Cleveland, Mourad Ouzzani, Shaun J Grannis, and Lawrence T Glickman. Lahva: Linked animal-human health visual analytics. In *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*, pages 27–34. IEEE, 2007.
- [MTR+13] YL Meur, B Thierry, G Rabita, S Dorel, G Honnorat, J Brisswalter, and C Hauswirth. Spring-mass behaviour during the run of an international triathlon competition. *Int J Sports Med*, 34:1–8, 2013.
- [Mun14] Tamara Munzner. *Visualization Analysis and Design*. CRC Press, 2014.
- [NSFR+12] Ran Nathan, Orr Spiegel, Scott Fortmann-Roe, Roi Harel, Martin Wikelski, and Wayne M Getz. Using tri-axial acceleration data to identify behavioral modes of free-ranging animals: general concepts and tools illustrated for griffon vultures. *The Journal of experimental biology*, 215(6):986–996, 2012.
- [NWH+12] Evelien Nackaerts, Johan Wagemans, Werner Helsen, Stephan P Swinnen, Nicole Wenderoth, and Kaat Alaerts. Recognizing biological motion and emotions from point-light displays in autism spectrum disorders. *PLoS one*, 7(9):e44473, 2012.
- [PCS95] Catherine Plaisant, David Carr, and Ben Shneiderman. Image-browser taxonomy and guidelines for designers. *IEEE Softw.*, 12(2):21–32, March 1995.
- [Pen01] CJ Pennycuik. Speeds and wingbeat frequencies of migrating birds compared with calculated benchmarks. *Journal of Experimental Biology*, 204(19):3283–3294, 2001.
- [Phy09] Thair Nu Phyu. Survey of classification techniques in data mining. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, pages 18–20, 2009.
- [Pla86] William Playfair. Commercial and political atlas: Representing, by copper-plate charts, the progress of the commerce, revenues, expenditure, and debts of england, during the whole of the eighteenth century. *Corry, London. Republished in The Commercial and Political Atlas and Statistical Breviary*, 1786.
- [Pow07] David M. W. Powers. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. Technical Report SIE-07-001, School of Informatics and Engineering, Flinders University, Adelaide, Australia, 2007.
- [PTWS14] Nicholas L Payne, Matthew D Taylor, Yuuki Y Watanabe, and Jayson M Semmens. From physiology to physics: are we recognizing the flexibility of biological tools? *J Exp Biol*, 217(Pt 3):317–22, 2014.
- [QCW+12] Lama Qasem, Antonia Cardew, Alexis Wilson, Iwan Griffiths, Lewis G Halsey, Emily LC Shepard, Adrian C Gleiss, and Rory Wilson. Tri-axial dynamic acceleration as a proxy for animal energy expenditure; should we be summing values or calculating the vector. *PLoS One*, 7(2):e31187–e31187, 2012.
- [RA07] Jodi JL Rowley and Ross A Alford. Techniques for tracking amphibians: the effects of tag attachment, and harmonic direction finding versus radio telemetry. *Amphibia-Reptilia*, 28(3):367–376, 2007.

-
- [RAL10] Valérie Renaudin, Muhammad Haris Afzal, and Gérard Lachapelle. Complete triaxis magnetometer calibration in the magnetic domain. *Journal of sensors*, 2010, 2010.
- [RCM⁺12] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 262–270. ACM, 2012.
- [RCW05] Yan Ropert-Coudert and Rory P Wilson. Trends and perspectives in animal-attached remote sensing. *Frontiers in Ecology and the Environment*, 3(8):437–444, 2005.
- [RG⁺05] Chotirat (Ann) Ratanamahatana, Jessica Lin 0001, Dimitrios Gunopulos, Eamonn J. Keogh, Michail Vlachos, and Gautam Das. Mining time series data. In Oded Maimon and Lior Rokach, editors, *The Data Mining and Knowledge Discovery Handbook*, pages 1069–1103. Springer, 2005.
- [RJM⁺04] John J Reilly, DM Jackson, Colette Montgomery, LA Kelly, Christine Slater, Stan Grant, and JY Paton. Total energy expenditure and physical activity in young scottish children: mixed longitudinal study. *The Lancet*, 363(9404):211–212, 2004.
- [RLG⁺05] Chotirat Ann Ralanamahatana, Jessica Lin, Dimitrios Gunopulos, Eamonn Keogh, Michail Vlachos, and Gautam Das. Mining time series data. In *Data Mining and Knowledge Discovery Handbook*, pages 1069–1103. Springer, 2005.
- [RLK⁺15] M. Rhligh, M. Luboschik, F. Krger, T. Kirste, H. Schumann, M. Bgl, B. Alsallakh, and S. Miksch. Supporting activity recognition by visual analytics. In *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*, pages 41–48, Oct 2015.
- [RLL⁺05a] Kathy Ryall, Neal Lesh, Tom Lanning, Darren Leigh, Hiroaki Miyashita, and Shigeru Makino. Querylines: Approximate query for visual browsing. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1765–1768, New York, NY, USA, 2005. ACM.
- [RLL⁺05b] Kathy Ryall, Neal Lesh, Tom Lanning, Darren Leigh, Hiroaki Miyashita, and Shigeru Makino. Querylines: Approximate query for visual browsing. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1765–1768, New York, NY, USA, 2005. ACM.
- [RLRS15] Christian Richter, Martin Luboschik, Martin Röhlig, and Heidrun Schumann. Sequencing of Categorical Time Series. Poster at the IEEE Conference on Visual Analytics Science and Technology (VAST), October 2015.
- [RLS⁺14] Martin Rohlig, Martin Luboschik, Heidrun Schumann, Markus Bogl, Bilal Alsallakh, and Silvia Miksch. Analyzing parameter influence on time-series segmentation and labeling. In *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*, pages 269–270. IEEE, 2014.
- [Rob07] Jonathan C Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV'07. Fifth International Conference on*, pages 61–71. IEEE, 2007.
- [SA14] Joan Serrà and Josep LI Arcos. An empirical evaluation of similarity measures for time series classification. *Knowledge-Based Systems*, 67:305–314, 2014.
- [SAS⁺11] Mai Sakai, Kagari Aoki, Katsufumi Sato, Masao Amano, Robin W Baird, Daniel L Webster, Gregory S Schorr, and Nobuyuki Miyazaki. Swim speed and acceleration measurements of short-finned pilot whales (*Globicephala macrorhynchus*) in Hawai'i. *Mammal study*, 36(1):55–59, 2011.

-
- [SCBN04] Katsufumi Sato, Jean-Benoît Charrassin, Charles-André Bost, and Yasuhiko Naito. Why do macaroni penguins choose shallow body angles that result in longer descent and ascent durations? *Journal of Experimental Biology*, 207(23):4057–4065, 2004.
- [Sen09] Pavel V Senin. Literature review on time series indexing. 2009.
- [SH08] Emily L C Shepard and Lewis G Halsey. Identification of animal movement patterns using tri-axial accelerometry. *Endangered Species Research*, 10:47–60, 2008.
- [Shn96] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.
- [SKU⁺12] Mike Sips, Patrick Köthür, Andrea Unger, Hans-Christian Hege, and Doris Dransch. A visual analytics approach to multiscale exploration of environmental time series. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2899–2907, 2012.
- [SLW⁺14] Pavel Senin, Jessica Lin, Xing Wang, Tim Oates, Sunil Gandhi, Arnold P Boedihardjo, Crystal Chen, Susan Frankenstein, and Manfred Lerner. Grammarviz 2.0: a tool for grammar-based pattern discovery in time series. In *Machine Learning and Knowledge Discovery in Databases*, pages 468–472. Springer, 2014.
- [Smi97] Steven W. Smith. *The scientist and engineer's guide to digital signal processing*. California Technical Publishing, San Diego, CA, USA, 1997.
- [SMW⁺14] David M Scantlebury, Michael GL Mills, Rory P Wilson, John W Wilson, Margaret EJ Mills, Sarah M Durant, Nigel C Bennett, Peter Bradford, Nikki J Marks, and John R Speakman. Flexible energetics of cheetah hunting strategies provide resistance against kleptoparasitism. *Science*, 346(6205):79–81, 2014.
- [SNS⁺10] Kozue Shiomi, Tomoko Narazaki, Katsufumi Sato, Kenichiro Shimatani, Nobuaki Arai, Paul J Ponganis, and Nobuyuki Miyazaki. Data-processing artefacts in three-dimensional dive path reconstruction from geomagnetic and acceleration data. *Aquat Biol*, 8:299–304, 2010.
- [Spe07] R Spense. Information visualization: Design for interaction, 2007.
- [SSC⁺10] Pamela Semanik, Jing Song, Rowland W Chang, Larry Manheim, Barbara Ainsworth, and Dorothy Dunlop. Assessing physical activity in persons with rheumatoid arthritis using accelerometry. *Medicine and science in sports and exercise*, 42(8):1493, 2010.
- [SSI⁺09a] Kentaro Q. Sakamoto, Katsufumi Sato, Mayumi Ishizuka, Yutaka Watanuki, Akinori Takahashi, Francis Daunt, and Sarah Wanless. Can ethograms be automatically generated using body acceleration data from free-ranging birds? *PLoS ONE*, 4(4):e5379, 04 2009.
- [SSI⁺09b] Kentaro Q Sakamoto, Katsufumi Sato, Mayumi Ishizuka, Yutaka Watanuki, Akinori Takahashi, Francis Daunt, and Sarah Wanless. Can ethograms be automatically generated using body acceleration data from free-ranging birds. *PLoS one*, 4(4):e5379, 2009.
- [Sta14] J. T. Stasko. The value of visualization...and why interaction matters, capstone speech, eurovis 2014, 2014.
- [SVM14] Carlos Oscar Sánchez Sorzano, J Vargas, and A Pascual Montano. A survey of dimensionality reduction techniques. *arXiv preprint arXiv:1403.2877*, 2014.
- [SWH⁺08] Emily LC Shepard, Rory P Wilson, Lewis G Halsey, Flavio Quintana, Agustina Gómez Laich, Adrian C Gleiss, Nikolai Liebsch, Andrew E Myers, and Brad Norman. Derivation of body motion via appropriate smoothing of acceleration data. *Aquat. Biol.*, 4:235–241, 2008.

-
- [SWLL13] Guo-Dao Sun, Ying-Cai Wu, Rong-Hua Liang, and Shi-Xia Liu. A survey of visual analytics techniques and applications: State-of-the-art research and future challenges. *Journal of Computer Science and Technology*, 28(5):852–867, 2013.
- [SWQ⁺08] Emily LC Shepard, Rory P Wilson, Flavio Quintana, Agustina Gómez Laich, Nikolai Liebsch, Diego A Albareda, Lewis G Halsey, Adrian Gleiss, David T Morgan, Andrew E Myers, et al. Identification of animal movement patterns using tri-axial accelerometry. *Endangered Species Research*, 10(2.1), 2008.
- [SWR⁺12] Allen R Sanderson, Brad Whitlock, Oliver Rübél, Hank Childs, and Gunther Weber. A system for query based analysis and visualization. 2012.
- [Sys] Jensen Software Systems. Multitrace. <http://www.jensen-software.com/>. Accessed: 25-05-2016.
- [SZWK12] Andreas Stoffel, Leishi Zhang, Stefan Hagen Weber, and Daniel A Keim. Amplio vqa—a web based visual query analysis system for micro grid energy mix planning. In *Proceedings of the Eurovis workshop on visual analytics (EuroVA)*, Vienna, 2012.
- [TAS04] Christian Tominski, James Abello, and Heidrun Schumann. Axes-based visualizations with radial layouts. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 1242–1247. ACM, 2004.
- [TC06] James J Thomas and Kristin A Cook. A visual analytics agenda. *Computer Graphics and Applications, IEEE*, 26(1):10–13, 2006.
- [TDH⁺16] Jason D Thiem, Jeff W Dawson, Daniel Hatin, Andy J Danylchuk, Pierre Dumont, Adrian C Gleiss, Rory P Wilson, and Steven J Cooke. Swimming activity and energetic costs of adult lake sturgeon during fishway passage. *Journal of Experimental Biology*, 219(16):2534–2544, 2016.
- [Tea06] NA Teanby. An icosahedron-based method for even binning of globally distributed remote sensing data. *Computers & Geosciences*, 32(9):1442–1450, 2006.
- [TK07] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *Int J Data Warehousing and Mining*, 2007:1–13, 2007.
- [TKC17] Gary KL Tam, Vivek Kothari, and Min Chen. An analysis of machine-and human-analytics in classification. *IEEE transactions on visualization and computer graphics*, 23(1):71–80, 2017.
- [TLMP11] Stewart G Trost, Paul D Loprinzi, Rebecca Moore, and Karin A Pfeiffer. Comparison of accelerometer cut points for predicting activity intensity in youth. *Med Sci Sports Exerc*, 43(7):1360–1368, 2011.
- [TMB02] Barbara Tversky, Julie Bauer Morrison, and Mireille Betrancourt. Animation: can it facilitate? *International journal of human-computer studies*, 57(4):247–262, 2002.
- [TPRH11] Cagatay Turkay, Július Parulek, Nathalie Reuter, and Helwig Hauser. Interactive visual analysis of temporal cluster structures. In *Computer Graphics Forum*, volume 30, pages 711–720. Wiley Online Library, 2011.
- [Tre82] Anne Treisman. Perceptual grouping and attention in visual search for features and objects. *Journal of Experimental Psychology: Human Perception and Performance*, pages 194–214, 1982.
- [TSAA12] Christian Tominski, Heidrun Schumann, Gennady Andrienko, and Natalia Andrienko. Stacking-based visualization of trajectory attribute data. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2565–2574, 2012.
- [TSWS05] Christian Tominski, Petra Schulze-Wollgast, and Heidrun Schumann. 3d information visualization for time dependent data on maps. In *Information Visualisation, 2005. Proceedings. Ninth International Conference on*, pages 175–181. IEEE, 2005.

-
- [Tuf01] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, second edition, 2001.
- [vdEvW11] S. van den Elzen and J.J. van Wijk. Baobabview: Interactive construction and analysis of decision trees. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 151–160, Oct 2011.
- [vLAA⁺13] Tatiana von Landesberger, Gennady Andrienko, Natalia Andrienko, Sebastian Bremm, Matthias Kirschner, Stefan Wesarg, and Arjan Kuijper. Opening up the black box of medical image segmentation with statistical shape models. *The Visual Computer*, 29(9):893–905, 2013.
- [VSGW12] Sylvie P Vandenabeele, Emily L Shepard, Adam Grogan, and Rory P Wilson. When three per cent may not be three per cent; device-equipped seabirds experience variable flight constraints. *Marine Biology*, 159(1):1–14, 2012.
- [VW08] Jarke J Van Wijk. Unfolding the earth: Myriahedral projections. *The Cartographic Journal*, 45(1):32–42, 2008.
- [VWVdW99] Jarke J Van Wijk and Huub Van de Wetering. Cushion treemaps: Visualization of hierarchical information. In *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*, pages 73–78. IEEE, 1999.
- [VWVS99] Jarke J Van Wijk and Edward R Van Selow. Cluster and calendar based visualization of time series data. In *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*, pages 4–9. IEEE, 1999.
- [VWW13] Sylvie P Vandenabeele, Rory P Wilson, and Martin Wikelski. New tracking philosophy for birds. *Frontiers in Ecology and the Environment*, 11(1):10–12, 2013.
- [WaCPR11] Rick Walker, Llyr ap Cenydd, Serban Pop, and Jonathan C Roberts. Visual analysis of user behaviour in pay-per-bid auctions. 2011.
- [WAG05] L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 157–164, Oct 2005.
- [WAM01] Marc Weber, Marc Alexa, and Wolfgang Müller. Visualizing time-series on spirals. In *Infovis*, volume 1, pages 7–14, 2001.
- [WAPW06] Colin Ware, Roland Arsenault, Matthew Plumlee, and David Wiley. Visualizing the underwater behavior of humpback whales. *IEEE Computer Graphics and Applications*, 26(4):14–18, 2006.
- [War04] Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [War10] Colin Ware. *Visual thinking: For design*. Morgan Kaufmann, 2010.
- [Wat01] Martin Wattenberg. Sketching a graph to query a time-series database. In *CHI'01 Extended Abstracts on Human factors in Computing Systems*, pages 381–382. ACM, 2001.
- [WB98] Ming-Chuan Wu and Alejandro P Buchmann. Encoded bitmap indexing for data warehouses. In *Data Engineering, 1998. Proceedings., 14th International Conference on*, pages 220–230. IEEE, 1998.
- [WBE97] Geoffrey B West, James H Brown, and Brian J Enquist. A general model for the origin of allometric scaling laws in biology. *Science*, 276(5309):122–126, 1997.
- [WBJ16] James Walker, Rita Borgo, and Mark W. Jones. Timenotes: A study on effective chart visualization and interaction techniques for time-series data. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of Information Visualization 2015)*, 23(1):549–558, January 2016.

-
- [WCBL94] RP Wilson, BM Culik, R Bannasch, and J Lage. Monitoring antarctic environmental variables using penguins. *Marine ecology progress series. Oldendorf*, 106(1):199–202, 1994.
- [WCDA91] Rory P Wilson, B Culik, R Danfeld, and D Adelung. People in antarctica how much do adélie penguins *pygoscelis adeliae* care? *Polar Biology*, 11(6):363–370, 1991.
- [WFHM99] Sarah Wanless, Suzanne K Finney, Michael P Harris, and Dominic J McCafferty. Effect of the diel light cycle on the diving behaviour of two bottom feeding marine birds: the blue-eyed shag *phalacrocorax atriceps* and the european shag *p. aristotelis*. *Marine Ecology Progress Series*, 188:219–224, 1999.
- [WGJL12] James Walker, Zhao Geng, Mark Jones, and Robert S Laramee. Visualization of Large, Time-Dependent, Abstract Data with Integrated Spherical and Parallel Coordinates. pages 43–47, Vienna, Austria, 2012. Eurographics Association.
- [WGK10] Matthew O Ward, Georges Grinstein, and Daniel Keim. *Interactive data visualization: foundations, techniques, and applications*. CRC Press, 2010.
- [WGM⁺14] Rory P Wilson, Ed Grundy, Richard Massy, Joseph Soltis, Brenda Tysse, Mark Holton, Yuzhi Cai, Andy Parrott, Luke A Downey, Lama Qasem, et al. Wild state secrets: ultra-sensitive measurement of micro-movement can reveal internal processes in animals. *Frontiers in Ecology and the Environment*, 12(10):582–587, 2014.
- [WGS⁺02] Rory P Wilson, David Grémillet, Jon Syder, Mandy AM Kierspel, Stefan Garthe, Henri Weimerskirch, Christian Schäfer-Neth, J Alejandro Scolaro, C Bost, Joachim Plötz, et al. Remote-sensing systems and seabirds: their use, abuse and potential for measuring marine environmental variables. *Marine Ecology Progress Series*, 228:241–261, 2002.
- [WHR⁺92] Rory P Wilson, Kit Hustler, Peter G Ryan, Alan E Burger, and E Christian Noldeke. Diving birds in cold water: do archimedes and boyle determine energetic costs? *American Naturalist*, pages 179–200, 1992.
- [WHW⁺15] Gwendoline Ixia Wilson, Mark D. Holton, James Walker, Mark W. Jones, Ed Grundy, Ian M. Davies, David Clarke, Adrian Luckman, Nick Russill, Vianney Wilson, Rosie Plummer, and Rory P. Wilson. A new perspective on how humans assess their surroundings; derivation of head orientation and its role in 'framing' the environment. *PeerJ*, 3:e908, 2015.
- [WHW⁺16] Rory P Wilson, Mark D Holton, James S Walker, Emily LC Shepard, D Mike Scantlebury, Vianney L Wilson, Gwendoline I Wilson, Brenda Tysse, Mike Gravenor, Javier Ciancio, et al. A spherical-plot solution to linking acceleration metrics with animal performance, state, behaviour and lifestyle. *Movement ecology*, 4(1):22, 2016.
- [WIK⁺05] Shinichi Watanabe, Masako Izawa, Akiko Kato, Yan Ropert-Coudert, and Yasuhiko Naito. A new technique for monitoring the detailed behaviour of terrestrial animals: a case study with the domestic cat. *Applied Animal Behaviour Science*, 94(1):117–131, 2005.
- [WJL⁺15a] James S. Walker, Mark W. Jones, Robert S. Laramee, Owen R. Bidder, Hannah J. Williams, Rebecca Scott, Emily L. C. Shepard, and Rory P. Wilson. Time-classifier: a visual analytic system for the classification of multi-dimensional time series data. *The Visual Computer*, 31(6-8):1067–1078, 2015.
- [WJL⁺15b] James S. Walker, Mark W. Jones, Robert S. Laramee, Mark D. Holton, Emily L. C. Shepard, Hannah J. Williams, D. Michael Scantlebury, J. Marks Nikki, Elizabeth A. Magowan, Iain E. Maguire, Owen R. Bidder, Agustina Di Virgilio, and Rory P. Wilson. Prying into the intimate secrets of animal lives; software beyond hardware for comprehensive annotation in daily diary tags. *Movement Ecology*, 3(29), September 2015.

-
- [WKK⁺07] Martin Wikelski, Roland W Kays, N Jeremy Kasdin, Kasper Thorup, James A Smith, and George W Swenson. Going wild: what a global small-animal tracking system could do for experimental biologists. *Journal of Experimental Biology*, 210(2):181–186, 2007.
- [WLD⁺07] Rory P Wilson, Nikolai Liebsch, Ian M Davies, Flavio Quintana, Henri Weimerskirch, Sandra Storch, Klaus Lucke, Ursula Siebert, Solvin Zankl, Gabriele Müller, et al. All at sea with animal tracks; methodological and analytical solutions for the resolution of movement. *Deep Sea Research Part II: Topical Studies in Oceanography*, 54(3):193–210, 2007.
- [WLY⁺14] Yingcai Wu, Shixia Liu, Kai Yan, Mengchen Liu, and Fangzhao Wu. Opinion-flow: Visual analysis of opinion diffusion on social media. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1763–1772, 2014.
- [WNW⁺15] Gwendoline Ixia Wilson, Brad Norman, James S. Walker, Hannah J. Williams, Mark D. Holton, D. Clarke, and Rory P. Wilson. In search of rules behind environmental framing; the case of head pitch. *Movement ecology*, 3(1):1–9, 2015.
- [WPP⁺97] Rory P Wilson, Klemens Pütz, Gerrit Peters, Boris Culik, J Alejandro Scolaro, Jean-Benoît Charrassin, and Yan Ropert-Coudert. Long-term attachment of transmitting and recording devices to penguins and other seabirds. *Wildlife Society Bulletin*, pages 101–106, 1997.
- [WSL08] Rory P Wilson, ELC Shepard, and N Liebsch. Prying into the intimate details of animal lives: use of a daily diary on animals. *Endangered Species Research*, 4(1-2):123–137, 2008.
- [WSSM12] Jishang Wei, Zeqian Shen, Neel Sundaresan, and Kwan-Liu Ma. Visual cluster exploration of web clickstream data. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 3–12. IEEE, 2012.
- [WT13] Yuuki Y Watanabe and Akinori Takahashi. Linking animal-borne video to accelerometers reveals prey capture variability. *Proceedings of the National Academy of Sciences*, 110(6):2199–2204, 2013.
- [WTD⁺05] Yutaka Watanuki, Akinori Takahashi, Francis Daunt, Sarah Wanless, Mike Harris, Katsufumi Sato, and Yasuhiko Naito. Regulation of stroke and glide in a foot-propelled avian diver. *Journal of Experimental biology*, 208(12):2207–2216, 2005.
- [WW88] Rory P Wilson and Marie-Pierre T Wilson. Foraging behaviour in four sympatric cormorants. *The Journal of Animal Ecology*, pages 943–955, 1988.
- [WWQ⁺06] Rory P Wilson, Craig R White, Flavio Quintana, Lewis G Halsey, Nikolai Liebsch, Graham R Martin, and Patrick J Butler. Moving towards acceleration for estimates of activity-specific metabolic rate in free-living animals: the case of the cormorant. *Journal of Animal Ecology*, 75(5):1081–1090, 2006.
- [XCH⁺14] Cong Xie, Wei Chen, Xinxin Huang, Yueqi Hu, Scott Barlowe, and Jing Yang. Vaet: A visual analytics approach for e-transactions time-series. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1743–1752, 2014.
- [YH10] Che-Chang Yang and Yeh-Liang Hsu. A review of accelerometry-based wearable motion detectors for physical activity monitoring. *Sensors*, 10(8):7772–7788, 2010.
- [ZCPB11] Jian Zhao, Fanny Chevalier, Emmanuel Pietriga, and Ravin Balakrishnan. Exploratory Analysis of Time-series with ChronoLenses. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2422–2431, October 2011.
- [ZCW⁺14] Jian Zhao, Nan Cao, Zhen Wen, Yale Song, Yu-Ru Lin, and Christopher M Collins. Fluxflow: Visual analysis of anomalous information spreading on social media. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1773–1782, 2014.

-
- [ZJG⁺10] Hartmut Ziegler, Marco Jenny, Tino Gruse, Daniel Keim, et al. Visual market sector analysis for financial time series data. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 83–90. IEEE, 2010.
- [ZSB⁺12] Leishi Zhang, Andreas Stoffel, Michael Behrisch, Sebastian Mittelstadt, Tobias Schreck, René Pompl, Simon Weber, Holger Last, and Daniel Keim. Visual analytics for the big data eraa comparative review of state-of-the-art commercial systems. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 173–182. IEEE, 2012.