Paper:

Camplani, M., Paiement, A., Mirmehdi, M., Damen, D., Hannuuna, S., Burghardt, T. & Tao, L. (2016). Multiple Human Tracking in RGB-D Data: A Survey. *IET Computer Vision*
http://dx.doi.org/10.1049/iet-cvi.2016.0178

IET Computer Vision

Review Article

Multiple human tracking in RGB-depth data: a survey

ISSN 1751-9632
Received on 10th June 2016
Revised 19th November 2016
Accepted on 8th December 2016
E-First on 22nd March 2017
doi: 10.1049/iet-cvi.2016.0178
www.ietdl.org

Massimo Camplani[1], Adeline Paiement[1], Majid Mirmehdi[1] ✉, Dima Damen[1], Sion Hannuna[1], Tilo Burghardt[1], Lili Tao[1]

[1]Visual Information Laboratory, Faculty of Engineering, University of Bristol, Bristol BS8 1UB, UK

✉ *E-mail: m.mirmehdi@bristol.ac.uk*

Abstract: Multiple human tracking (MHT) is a fundamental task in many computer vision applications. Appearance-based approaches, primarily formulated on RGB data, are constrained and affected by problems arising from occlusions and/or illumination variations. In recent years, the arrival of cheap RGB-depth devices has led to many new approaches to MHT, and many of these integrate colour and depth cues to improve each and every stage of the process. In this survey, the authors present the common processing pipeline of these methods and review their methodology based (a) on how they implement this pipeline and (b) on what role depth plays within each stage of it. They identify and introduce existing, publicly available, benchmark datasets and software resources that fuse colour and depth data for MHT. Finally, they present a brief comparative evaluation of the performance of those works that have applied their methods to these datasets.

1 Introduction

Human tracking is a key component in many computer vision applications, including video surveillance [1], smart environments [2], assisted living [3, 4], advanced driver assistance systems (ADAS) [5], and sport analysis [6]. They are usually centred around RGB sensors and are characterised by a variety of limitations, such as occlusions due to cluttered or crowded scenes and varying illumination conditions. The vast literature landscape in this research area has widened even further in the last few years, due to the introduction and popularity of low-cost RGB-depth (RGB-D) cameras (such as the Kinect [7] and Asus Xtion [8]). This has enabled the development of new algorithms that integrate depth and colour cues to improve detection and tracking systems [9].

The aim of this survey paper is to summarise and focus on the area of multiple human *tracking (MHT) from the combination of colour (RGB) and depth (D)* data, given that cheap depth-enabled sensors are becoming ubiquitous in computer vision research and applications. The survey is not however limited to methods using active sensing RGB-D devices, but also encompasses state-of-the-art passive sensing stereo-based human tracking techniques, where colour and depth are again jointly relied upon to enable tracking.

We do not review methods based only on RGB features as that would need a dedicated survey of its own and would demand much greater space – for RGB only MHT, the reader is referred to the reviews presented by Dollar *et al.* [10] on colour-based pedestrian detection and Luo *et al.* [11] for colour-based multi-object tracking. The intention here rather is to address and summarise an area that is now of far-reaching interest to a huge community of researchers.

Four main computer vision topics were identified in [9] that could benefit from depth information: human activity analysis and recognition [12, 13], hand gesture analysis [14], three-dimensional (3D) mapping [15] and object detection and tracking. For example, the effect of occlusions can be reduced by using the 3D information contained in depth data, or more reliable features can be extracted in scenes undergoing illumination variations since such variations have low impact on depth sensors. Moreover, depth can be used to extract a richer description of the scene allowing to simplify the tracking problem, e.g. by adding physical constraints on human appearance and size. On the other hand, certain depth sensor characteristics, such as limited capture range, or scene characteristics, such as excessive natural light and reflective surfaces, reduce the reliability of depth data in some operating conditions, e.g. in outdoor scenarios. Colour and depth data can be significantly complementary, and hence their efficient combination and processing can dramatically reduce the effect of the problems that affect them individually. In this survey, we focus on the analysis of algorithms, and available datasets and software, which *combine colour and depth data* for MHT. Most previous survey papers on human tracking do not provide such coverage and are limited to one or other aspects of MHT. For example, in [1], an in-depth review of surveillance systems is provided, with particular focus on challenges in using large camera networks. In [16], pedestrian detection methods using colour-based approaches are surveyed while the pedestrian detection review presented in [5] is mainly focused on ADAS systems. The survey presented in [10] proposes an extensive evaluation of 16 pedestrian detectors that are based on a sliding window strategy. In [17], the focus of the survey is on algorithms for high-level crowd scene understanding. A review of human detection algorithms in video surveillance applications is presented in [18] where the main sub-modules of the human detection task are identified (object detection and object classification) and the state-of-the-art algorithms are appraised by describing the different strategies used in each sub-module. The survey presented in [19] summarises the advances in human body parts tracking for rehabilitation purposes. Table 1 lists the recent surveys that are related in some fashion to MHT.

To the best of our knowledge, the surveys most closely associated to ours here are those presented in [9, 11, 12]. These cover similar themes but come with certain limitations. The work in [9] reviews recent Kinect-based applications in computer vision, including a very brief survey of RGB-D based trackers. The review in [12] is focused on the recent advances on human activity analysis using depth imagery, while the problem of human detection and tracking is only marginally discussed. Finally, in [11], a general review of multiple object tracking is presented, but the analysis, dedicated to approaches that combine colour and depth data, is limited and brief.

It is worth noting that we do not consider general single object trackers based on combined depth and colour features, such as the recent works presented in [20–23], since they are more focused on the optimisation of appearance and motion models rather than facing the specific challenges of MHT, or are concerned with tracking inanimate objects. Furthermore, we do not include

Table 1 Recent related surveys (most recent first)

Year	Article	Topic
2016	Zhang *et al.* [13]	RGB-D dataset for action recognition
2014	Luo *et al.* [11]	MHT
2013	Chen *et al.* [12]	human activity analysis
2013	Han *et al.* [9]	recent Kinect applications
2013	Li *et al.* [17]	crowd monitoring
2013	Paul *et al.* [18]	human detection in surveillance
2013	Wang [1]	multi-camera video surveillance
2012	Dollar *et al.* [10]	pedestrian detection
2010	Geronimo *et al.* [5]	ADAS systems
2009	Enzweiler and Gavrila [16]	pedestrian detection
2008	Zhou and Hu [19]	human tracking in rehabilitation

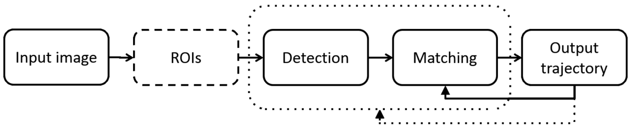

Fig. 1 *Common processing pipeline for MHT – the dashed-line stage is an optional step of the pipeline, while the dotted rectangle and arrow depict a variation of it*

detection only methods, e.g. [24–26], and methods that use depth only for MHT, e.g. [27–30], or depth and reflectance, such as [31]. Finally, we do not include in this review any work or dataset that is related to the analysis of people interaction, such as [32, 33], or action recognition, such as [34], as they are not focused on the problems and issues of MHT.

In summary, we provide here a review of the state-of-the-art on MHT algorithms that integrate depth and colour data, characterising them based on (a) trajectory representation and matching and (b) how they exploit depth information to improve various stages of the processing pipeline. We also provide a review of the constraints of use of these algorithms, and we examine existing online resources, i.e. benchmark datasets and source codes, and present a comparison of the very few such resources made available to the community. The audience of this survey is not limited to researchers working directly in the development of tracking algorithms, but also includes those who wish to employ a tracking method that is relevant to their application area, where colour and depth sequences are to be analysed, such as the very active research area of action recognition [12, 13], smart environments [35], health-care applications [36, 37], and applications mentioned in [9].

Next, in Section 2, we present the common processing stages of a typical MHT system, along with a variation on it employed by some works. Amongst other topics, we cover some generic descriptions of a person and introduce two characterisations of MHT systems based on their matching strategy and use of depth. These characterisations are then used in Sections 3 and 4 to survey state-of-the-art approaches. Practical issues, such as type of sensor, camera position, and speed of computation, are considered in Section 5. Section 6 presents an overview of online datasets and software resources for RGB-D MHT. We then compare existing evaluations derived from some of the works in this survey in Section 7. We highlight the main challenges within the current state-of-the-art of RGB-D MHT in Section 8 and conclude in Section 9.

2 Multiple people detection and tracking techniques in RGB-D data

In this section, we identify the main approaches to MHT from combined colour and depth data. We first present the processing pipeline that can be attributed to the greater set of works in the literature and then characterise the works we review based on (a) which trajectory representation is used and its matching, and (b) how and for which purpose depth data is exploited.

In MHT, detections of multiple people are normally aggregated into independent tracks, one for each person, in order to establish their respective trajectories. Tracks may contain position, motion, and appearance descriptions. We shall use the words ‘track’ and ‘trajectory’ interchangeably in the rest of this paper.

The common processing pipeline is illustrated in Fig. 1. MHT methods normally perform the stages indicated by the solid lines in Fig. 1, with first a detection stage that searches for occurrences of humans in a new frame, based on a generic description of a person (elaborated later below). It may possibly be preceded by an optional Regions of Interest (ROIs) selection stage (the dashed-line box in Fig. 1), that allows for the reduction of the search space. Then, a matching step associates these new detections to the trajectories based on a matching strategy and a similarity measure, computed from position and, more often than not, appearance.

There are numerous approaches to performing the matching process. These rely on the active trajectories to provide (i.e., effectively feedback) a representation of the target's motion and appearance to their matching stage (the solid arrow in Fig. 1). The pool of active trajectories is managed by the matching stage, with new trajectories created when detections cannot be associated to the existing ones, and old trajectories discontinued when certain termination criteria are met.

In a variation to the common pipeline, depicted by the dotted line and box in Fig. 1, the detection and matching stages may be directed by trajectories and their representations rather than by a generic representation of a person. Thus, currently tracked people are directly detected at the position predicted by their trajectory representation's motion model in a significantly reduced search space. In effect, this amounts to combined detection and matching. This variation of the MHT processing pipeline still requires a generic person description for initialising new trajectories by detecting people that are not yet tracked. Note that some methods also use a generic person description in the combined detection and matching stage in addition to the trajectory representation, in order to ensure that the tracks do not switch to background objects of similar predicted position and appearance to that of the target.

Section 3 provides a detailed description of implementations of the MHT pipeline (and its variation), including comparing different fulfilments of the matching stage. It should be stressed that both the main pipeline and its variation are by no means specific to RGB-D based methods, and the same can apply to MHT methods based on RGB data only.

Trajectory representations in MHT methods vary significantly between implementations as well, as illustrated in Fig. 2. Thus, although all reviewed methods employ a motion model in their trajectory representation, the use of an appearance model (e.g., colour histogram, texture, joint RGB and depth feature etc.) is optional, as represented by the dashed arrow (or blue sub-tree) in Fig. 2. Both motion and appearance models may be built from an observation in a single frame, or from richer information that accounts for the history of the target.

Two types of motion models may be identified. The first, denoted as ‘zero-velocity motion model’, assumes stationary position of the target, while the second describes their velocity, yielding a first order characterisation of their movements. Higher order motion models, such as one that includes the target's acceleration, would be equally possible, but are not addressed in this survey as no methods in RGB-D MHT were encountered that employed them. Static appearance models may be built from one or a few initial frames and remain fixed for the duration of the trajectory's lifetime, while dynamic appearance models may be derived from all previous observations of the target or from a sliding window. Such models are updated as new observations become available, in order to account for varying appearances, due to different body orientation relative to the sensor or changing illumination conditions. Yet these dynamic models could result in incorrect descriptions in case of failure in tracking, such as drifting. MHT methods may use any combination of these different possible (static and dynamic) motion and appearance models.

Depth data can be exploited to enhance RGB-based MHT. The methods that we review can be characterised by how and at which stage of the MHT pipeline they employ depth information. Indeed,

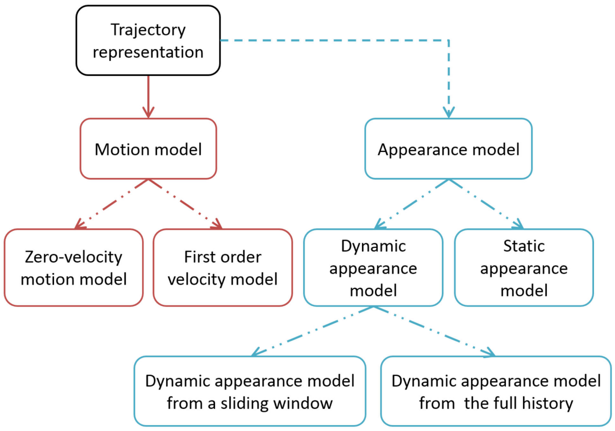
Trajectory representation
Motion model
Appearance model
Zero-velocity motion model
First order velocity model
Dynamic appearance model
Static appearance model
Dynamic appearance model from a sliding window
Dynamic appearance model from the full history

Fig. 2 *Categorisation of the different models that make up the trajectory representation used for matching. The dashed arrow denotes an optional model for the trajectory representation, while semi-dotted arrows indicate where one or the other of two possibilities is selected*

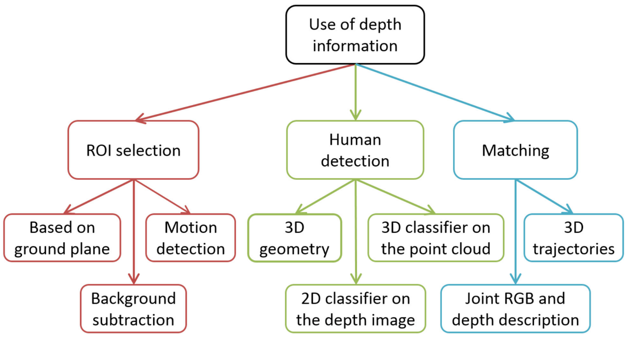

Fig. 3 *Categorisation of the uses of depth information in MHT methods from RGB-D data*

depth may support each and every stage of the MHT pipeline, as indicated in Fig. 3. It can help to specify *ROIs* in the image corresponding to 3D physical scene regions of significance, e.g. a doorway or passage, to help reduce the search space for the detection stage (left branch of Fig. 3). Depth information may also increase the robustness of *human detection*, by enhancing the generic description of a person with 3D shape information (middle branch of Fig. 3). Finally, depth can help in *matching* detected candidates and trajectories (right branch of Fig. 3), by providing the information needed to track people in 3D, and by further enriching the appearance descriptions of people, that are traditionally based on RGB information only. The various uses of depth information in published work will be detailed in Section 4.

The *generic description of a person* that drives the detection stage, is often made up of a number of RGB and depth cues. Then, in the detection stage, a cascade of RGB and depth based descriptors is applied to either the full image or ROIs, starting with the less computationally expensive ones, which are generally depth-based descriptors of the human shape. When using RGB information, the generic representations for a person often takes the form of a Histogram of Oriented Gradients (HOG) [38] descriptor of the full or upper body. Other examples of possible generic person descriptions from RGB data are provided by the poselet-based human detector of [39], the deformable part-based models of [40] (DPM) or use the Viola and Jones Adaboost cascade [41]. Table 2 summarises the different generic descriptions of people used by the various methods reviewed here.

Next, in Section 3, we review all RGB-D MHT methods known to us, leveraged on how they implement the MHT pipeline. We characterise these works based on their applied trajectory representation and matching strategy, following the categorisation proposed in Fig. 2. Then, in Section 4, we again review and characterise these *same methods* based on their adoption of depth information, describing the uses of depth for each stage of the MHT pipeline, according to Fig. 3.

3 Survey by MHT pipeline implementation

This section details how the pipeline for MHT, described in Section 2, has been implemented, including optional stages and variations. The emphasis is on complexity of trajectory representation (see Fig. 2) and matching. We may refer to depth data in this section for some of the works – the details of their use of depth is provided in Section 4.

3.1 Implementations of the main pipeline

We encountered only four works that build their trajectory representation exclusively from the previous frame [42–45]. The principle characteristics of their implementation of the MHT pipeline are indicated in the first four rows of Table 3.

Darrell *et al.* [45] present a stereo-based tracking approach using the target's position and size constancy from frame to frame. In particular, candidates are detected by using a segmentation approach that allows to identify connected component in the disparity images corresponding to regions in the 3D space with a typical volume occupied by a person facing the camera. For each detected region, a cascade of face and skin detectors, and geometric constraints, are applied to validate the target's head position. A long term model is generated by considering skin and face average colour, appearance colour histogram, face pattern and height extracted from depth data. These features are used to solve occlusions and target re-identification in case of targets re-entering in the scene.

Bansal *et al.* [42] first detect people after an ROI selection stage, using a combination of depth cues, and a HOG detector that is applied to a selection of edges obtained by preliminary template matching with several 2D contours of different body parts. Then, they match detections with trajectories from the previous frame by image patch-correlation. This is performed in the area of the image that contained in the previous observation of the person, after correction for camera motion estimated by visual odometry. Thus, the trajectory representation is made up of a zero-velocity motion model in the 2D image coordinates and an appearance model that consists of an image patch around the detection in the previous frame. This amounts to a dynamic appearance model built from a sliding-window of one frame-width.

Salas and Tomasi [43] detect and track all objects in ROIs that denote foreground, and then they select the paths that have, at some point in time, a detection with a high confidence score from a HOG based human detector. The matching stage is performed by dynamically building a directional connected graph of foreground object detections. These are organised into layers that correspond to the frames they originate from, and they are interconnected by the graph's edges in chronological order. The cost of an edge is the probability for its two nodes (or foreground detections) belonging to the same object. Based on these costs, the tracks are selected as the most strongly connected paths in the graph. A greedy algorithm is used for extracting individual paths, starting from the oldest remaining detection, and selecting the strongest connection locally between two adjacent node layers. The edge cost, used for matching, is estimated from the similarity of the colour signatures, measured using the Earth mover's distance [76], and from the distance between the 3D locations of both detections that is expected to be proportional to the elapsed time. Thus, the trajectory representation consists of a zero-velocity motion model, and an appearance model made up of the colour signature [76] of the blob in the previous frame.

Dan *et al.* [44] use depth information for detection. All detected candidates are then matched independently to detections in the previous frame by maximising a score that assesses both appearance similarity and closeness in 3D space. The trajectory representation used for matching is made up of an RGB-D based dynamic appearance model with a sliding window of one frame, along with a zero-velocity motion model. A backward/forward matching strategy is used, where all detections in frame t are matched to those in frame $t-1$ (backward matching), and vice versa (forward matching), which allows handling trajectory splits and merges, which may arise from the failure of detection in one direction that may match two people against the same candidate.

All four methods in [42–45] propose a crude motion model that does not describe a person's movements sufficiently well, although Salas and Tomasi [43] expects the distance travelled to be proportional to the time. The movement itself, and in particular its direction, are not captured by the trajectory representation. Thus, these methods are more likely to suffer from incorrect identifications when a track 'jumps' from one person to another, and from wrong detections being integrated into the tracks. In addition, both their motion and appearance models are made from the observations in the previous frame only. Hence, in case of occlusion, a person cannot be tracked any longer and the associated trajectory is automatically discontinued. A new, independent trajectory would have to be created if the person re-emerges.

The methods we present in the rest of this, and the following subsection, occupy rows five to the end of Table 3. These address the above issues (a) by proposing motion models that describe the motion of the target to the first order, and (b) by building appearance models from richer temporal information, which allow for maintaining consistent trajectory representations, and help prevent the model from changing dramatically in cases of temporary detection failure over a few frames.

In the work of Han *et al.* [46], the motion model determines target's velocity approximately by the mean and variance of its depth variations in the past ten frames. Their static appearance model is made up of colour and texture histograms for the torso and legs, generated at the first observation of a new person, with the torso and leg locations being detected using depth information. This trajectory representation is kept after the person leaves the scene, in order to allow re-identification in case of re-entry. People are first detected in ROIs, as objects within a pre-defined height range appearing for a number of successive frames, based on depth information. Their best matching trajectory is selected from a linear combination of the appearance similarity and the continuity of the depth variation. The former is assessed with the Bhattacharyya distance measure and the latter is expected to follow a Gaussian distribution with a mean and variance provided by the motion model, under the assumption of a constant speed.

Table 2 Types of generic descriptions of a person

Method	Depth descriptor	RGB descriptors
Bansal *et al.* [42]	✓	2D contour matching + HOG
Salas and Tomasi [43]	—	HOG
Dan *et al.* [44]	✓	—
Darrell *et al.* [45]	—	face and skin detector
Han *et al.* [46]	✓	—
Bajracharya *et al.* [47]	✓	—
Zhang *et al.* [48]	✓	HOG + poselet
Galamakis *et al.* [49]	✓	—
Liu *et al.* [50–52]	✓	joint RGB and height histogram + physical priors [52]
Luber *et al.* [53] and Linder and Arras [54]	✓	HOG
Ess *et al.* [55]	✓	HOG-based detectors
Jafari *et al.* [56]	✓	HOG
Muñoz Salinas *et al.* [57]	—	face detector
Munaro *et al.* [58, 59]	✓	HOG
Almazán and Jones [60, 61]	✓	—
Bahadori *et al.* [62]	—	temporal colour-based model
Beymer and Konolige [63]	✓	—
Satake *et al.* [64]	✓	SIFT
Vo *et al.* [65, 66]	✓	HOG + face detector
Harville [67]	✓	—
Muñoz Salinas *et al.* [68, 69]	✓	—
Muñoz Salinas *et al.* [70]	✓	Adaboost classifier for upper body + ellipse fitting at head location
Choi *et al.* [71, 72]	✓	HOG + face detector + motion detector + skin colour recognition
Migniot and Ababsa [73]	✓	—
Gao *et al.* [74]	—	HOG + DPM
Ma *et al.* [75]	—	HOG + DPM

In [47], Bajracharya *et al.* assume a target velocity of 2 ms^{-1} in any one direction, hence the motion model does not depend on the data. The appearance model of the trajectory representation is made up of the colour histogram of the last observation for the track. Matching is performed by comparing candidates, detected from depth information in ROIs, to trajectories, based on the colour histograms of the candidate and of the appearance model of the track, evaluated by the Bhattacharyya measure. Only trajectories that are predicted to be located close to the candidates are considered.

In all other RGB-D MHT methods reviewed next which apply the main MHT pipeline, motion is modelled as the position and velocity of the tracked person from the previous frame. The position, and sometimes the velocity, of the next observation are predicted from the model, and then compared with the positions of new detections during the matching stage. With the exception of [49, 55], the methods reviewed next carry out their predictions using Kalman filtering.

Some works find the best association of a detected candidate to a track independently for each detection or track. For example, in [48], Zhang *et al.* find people in ROIs using a cascade of RGB and depth-based detectors, where detected candidates from depth cues are verified by a HOG detector, and by the poselet-based human detector of [39] that detects body parts. This last detector is rather computationally expensive, hence it is only applied to detected candidates that cannot be associated with existing targets in the matching stage. The matching stage locates the best matching track or static background object for each new detected candidate, using a Directed Acyclic Graph (DAG) to handle the decision process. The DAG performs coarse matching by position similarity first and then finer matching to account for appearance similarity. The appearance is represented by a dynamic model, updated online by an AdaBoost algorithm. A classifier is trained by AdaBoost from weak nearest neighbour classifiers and colour histogram features, with positive and negative examples taken from previous observations of the target and of other people and objects, respectively. This model is kept after the person leaves the scene to enable future re-identification.

Similarly in [49], Galamakis *et al.* model motion as the target's speed, computed between the last two frames, and use it to predict the next position of the target, assuming a constant velocity. Following the matching strategy of [77], candidate detections, found by background subtraction, are associated with their nearest neighbour trajectories. Unlike [77] however, the distance to a trajectory combines the 3D distance to its predicted position and the appearance similarity, quantified as in [78] by a correlation metric. The appearance model comprises the hue and saturation histograms of the upper and lower body which are found by reference to the depth data. It is updated by linear combination of the current model and the new histogram. Liu *et al.* [50, 51] detect all candidate people in ROIs of a new frame from RGB-D data and then, for each track, select the best detected candidate by maximising a correspondence likelihood that is a linear combination of distance to the predicted position and appearance similarity, assessed by the Bhattacharyya measure. The appearance model of the trajectory representation is a joint colour and height histogram. The authors do not give any indication whether their appearance model is updated. To handle short-term occlusions, the trajectory is only terminated after 10 s of being lost.

Other works consider all possible associations of detections to tracks in order to find a global optimisation that takes into account possible interactions between tracks, such as crossing of trajectories and sharing of detections. In [53], Luber *et al.* build a tree of association hypotheses in a multi-hypothesis tracker (MuHyT) framework, where matching probabilities, for all past and current frames, are computed from closeness to position and velocity predictions, and from appearance similarity. The MuHyT grows a hypothesis tree, pruned to the k-best hypotheses at each iteration in order to prevent exponential growth of the tree. The current best hypothesis, that jointly describes all tracks, is then selected at each frame, following [79]. Similarly to [48], the appearance model relies on a colour and depth Adaboost classifier. Linder and Arras [54] propose an extension of the method in [53] for group tracking. In particular, to characterise group movements, they add to the MHT framework a set of coherent motion indicators, such as relative spatial distance, difference in velocity, and difference in orientation of two given tracks.

Beymer and Konolige [63] propose a combination of stereo-based background subtraction (see [80]) and a full body binary template to identify candidate targets. The binary template size is chosen according to the mean depth value of the foreground blob. A Kalman filter with a constant velocity model is used for tracking. A target's representation includes 3D space coordinates and two appearance models, a colour model and the average disparity. These models are linearly updated taking into account the confidence rate of the person detector module, such that it introduces a smoothing factor in the update process, hence limiting the models' drift. A similar approach was proposed by Bahadori *et al.* [62], using detected foreground regions and geometric constraints in their stereo setup to identify blobs containing candidate targets. For each blob, a fixed resolution and adaptive colour-based appearance model is generated, with each pixel modelled as a unimodal distribution in the colour space. Tracking is also performed with a Kalman filter, with a constant velocity model that takes into account the 3D depth position of the target and its appearance. The matching strategy is based on the minimisation of the distance, considering both position and appearance, between all the detected candidates and the current active tracks. The generation of new tracks and the termination of lost ones are managed by a finite state machine system.

Ess *et al.* [55] detect people in a Bayesian network that accounts for the probabilities of human presence, as output by a colour-based detector, given the scene geometry and a generic person geometry description, both provided by depth data. Areas around the next expected target locations also see their detection likelihood increased. Then, they also build multiple candidate tracks, from forward and backward matching hypotheses, following [81]'s tracking framework. These hypotheses are generated from position predictions by a constant velocity model and from appearance similarity measured using the Bhattacharyya distance on colour histograms. The best tracks are selected, while enforcing that each person detection can only be matched to one trajectory. The trajectory's appearance model used for matching is the mean colour histogram of all previous observations of the tracked person. Jafari *et al.* [56] use the same matching stage and trajectory representation. They perform detection in ROIs based on depth at a close range and using a HOG detector [82] in the far range.

Satake *et al.* [64] detect people by applying a classifier cascade to the RGB-D data. First, a set of three binary templates [83], containing frontal and side views of head and shoulders, are used to identify candidate regions in the disparity map. These are then validated and refined with a support vector machine (SVM) classifier trained on HOG features to detect humans. An extended Kalman filter is used to track the target in the 3D space. SIFT features [84] of the target are periodically collected to build an appearance model. Association between tracked targets and current

Table 3 Characterisation of the methods based on their MHT pipeline implementation. The number of frames indicated in the column 'Dynamic – sliding window' indicates the width of the window. For Liu *et al.* [50–52], it is not known if the appearance model is static or dynamic

Method	ROI selection	Pipeline variation	Motion model		Appearance model		
			Zero velocity	First-order velocity	Static	Dynamic-sliding window	Dynamic-full history
Bansal *et al.* [42]	✓		✓			✓ (1 frame)	
Salas and Tomasi [43]	✓		✓			✓ (1 frame)	
Dan *et al.* [44]			✓			✓ (1 frame)	
Darrell *et al.* [45]	✓		✓			✓	✓
Han *et al.* [46]	✓			✓		✓ (10 frames)	
Bajracharya *et al.* [47]	✓			✓		✓ (1 frame)	
Zhang *et al.* [48]	✓			✓			✓
Galamakis *et al.* [49]	✓			✓			✓
Liu *et al.* [50–52]	✓			✓	?	?	?
Luber *et al.* [53] and Linder and Arras [54]				✓			✓
Ess *et al.* [55]	✓			✓			✓
Jafari *et al.* [56]	✓			✓			✓
Muñoz Salinas *et al.* [57]	✓			✓			✓
Munaro *et al.* [58, 59]	✓			✓			✓
Almazán and Jones [60]	✓			✓		✓ (1 frame)	
Bahadori *et al.* [62]	✓			✓			✓
Beymer *et al.* [63]	✓			✓			✓
Satake *et al.* [64]	✓			✓		✓ (30 frames)	
Vo *et al.* [65, 66]	✓			✓			
Harville *et al.* [67]	✓			✓			✓
Almazán and Jones [61]	✓	✓		✓			✓
Muñoz Salinas [68]	✓	✓		✓			✓
Muñoz Salinas *et al.* [69]	✓	✓		✓			
Muñoz Salinas *et al.* [70]		✓		✓			✓
Choi *et al.* [71]	✓	✓		✓			
Choi *et al.* [72]	✓	✓		✓	✓		
Migniot and Ababsa [73]	✓			✓			
Gao *et al.* [74]	✓	✓		✓			✓
Ma *et al.* [75]	✓			✓		✓	

frame detections is performed by thresholding on the number of matching SIFT features.

Muñoz Salinas *et al.* [57] detect people from a face detector applied in ROIs selected from depth information. The face detector may suffer from false negatives (FNs) in non-fronto-parallel views, therefore it is only applied at the very end of the detection cascade, and only to detected candidates that cannot be associated with existing targets in the matching stage. The matching stage finds the globally optimal associations of detected candidates to existing tracks using the Hungarian method [85]. The matching likelihoods are computed from the distance to the predicted position and the similarity to the colour histogram appearance model estimated with the Bhattacharyya measure. This model is updated by linearly combining its current values and the new observed colour histogram. The track is discontinued if the new observation of the target is not encountered after a time-limit. Almazán and Jones [60] also use the Hungarian method to match candidates, detected from motion and size using depth information, to trajectories. The correspondence likelihood is based on the distance to the predicted position and on appearance similarity, evaluated using the Bhattacharyya measure. The appearance model combines a height histogram and the colour distributions of its bins, and it is updated every ten frames by replacing bins and their associated distributions by newly observed ones if available, i.e. if no occlusion happens.

Another method based on the Hungarian algorithm for matching detected and tracked objects is that of Vo *et al.* in [65], where the authors identify background areas with a depth-based occupancy grid system. Candidate targets' search space is limited to the foreground areas which is analysed with a cascade of classifiers, comprising face and skin detectors (see [66] for more details) and a full body HOG-based human detector [38]. Detected objects are tracked simultaneously with a compressive tracker and a Kalman filter. Munaro *et al.* [58, 59] find the optimal assignment of detections to tracks in a Global Nearest Neighbour framework. Their matching likelihoods are obtained from the distance to the predicted position and velocity, the probability of being a human as evaluated by a HOG-based human detector, and the similarity to the appearance model of the track. The latter is provided by an online Adaboost classifier trained on previous observations, and selects features in the colour histogram space. Harville [67] detects moving candidates by applying the background subtraction algorithm presented in [86] to RGB-D data. The detected foreground objects are projected to a 2D reference plane where occupancy and height maps are generated. A box filter system is applied to the occupancy map such that 3D clusters not corresponding to a volume occupied by an average adult are filtered out. Their tracking Kalman filter state includes position in the reference plane and the height and occupancy maps data. These features are linearly combined to calculate the matching score that it is used in the measurements and update phases of the Kalman filter.

Ma *et al.* [75] present a tracking approach where a set of HOG-based DPM detectors [40] is applied to both depth and colour images to detect body parts to enable their system to deal with a person's articulated motion. The conditional random field-based approach of [87] is used and extended to solve data association and trajectory estimation. In particular, person locations are inferred by minimising an objective function, which includes detection matching, spatial correlation, mutual exclusion, temporal consistence, and regularisation constraints. One interesting aspect of this method is that it can deal with flexible number and type of detectors.

3.2 Implementations of the variation of the pipeline

The detection of humans driven by generic full body descriptions, such as those mentioned earlier in Section 3.1, may sometimes be problematic, e.g. when there is partial occlusion which can significantly alter the appearance of the target. In Section 2 (also see Fig. 1), we stated that in a variation to the common pipeline, some works attempt to address such difficult detections by exploiting trajectories and their representations in a combined detection and matching stage to enable more robust detection. The trajectory representations provide descriptions of the targets, including first-order motion models that enable predictive tracking.

After an ROI selection stage, Almazán and Jones [61] use the mean-shift algorithm to find the ROI that best matches the appearance model of a target. For each trajectory, this search is initialised at the position predicted by a Kalman filter, and it is performed in the area defined by both the position variance estimated by the filter and by the ROI selection. The appearance model is made up of the colour histogram of the upper body region, comprising the head and the torso. After thresholding the 3D point cloud occupied by the person, the upper body region is estimated relative to the height of the 3D cluster from the ground plane. The corresponding colour histogram is then updated dynamically with each new observation as the weighted mean of the model at the previous frame and of the new histogram. The trajectory remains active until a number of frames after the target leaves the scene to allow its re-identification in case of temporary occlusion.

All other methods we review here that employ this variation of the MHT pipeline implement their first-order motion model in a particle filter framework. A potential drawback of this is that particle filters tend to be computationally expensive and may require optimisations to achieve practical running times. In the works of Muñoz Salinas *et al.* [68–70] one particle filter is used per track, using a constant speed model to predict the next location of the target, and new target observations are searched for by maximising a detection probability. In [68, 69], candidates are identified in ROIs based on depth information, wherein the probability of the presence of (any) person is computed based on heuristic rules on the number of points in a cluster and its maximal height. To compute the probability of detecting the tracked person, this human presence probability is combined with an interaction factor that allows handling trajectory crossings by imposing a minimal separation between the positions of different people. In [68], the detection probability also includes the Bhattacharyya appearance similarity measure, while in [69] it uses a measure of confidence on depth. Hence, the trajectory representation in [69] does not include any appearance model, and in [68] it models appearance by the colour histogram of the cluster. This model is updated with new observations that have high detection and matching confidence by the linear combination of the previous model and of the new histogram. This confidence condition avoids the model being updated when the detection contains parts of a different person, in case of close interaction between people. In [70], the detection probability is made up of three terms. It includes the probability of being a frontal-facing human, firstly by verifying that the cluster may be approximated by a vertical plane at the expected distance from the camera, and secondly, by evaluating the fitting of an ellipse on the RGB image in order to validate the presence of the elliptical shape of a head at this position. It also uses the Bhattacharyya appearance similarity measure to compare to the trajectory representation's appearance model, made up of two (colour) histograms inside two ellipses of pre-defined sizes and respective positions that represent the head and torso respectively. This appearance model is updated dynamically as in [68].

In all three methods in [68–70], new tracks are initialised when unknown targets are detected based on the use of generic person descriptions. In [68], heuristic rules on the size and height of clusters are used. In [69], confidence on depth is added and new trajectories are initialised only after a few consecutive detections. In [70], the detection of new people is first performed by an Adaboost classifier trained on RGB images to detect upper bodies which are verified by heuristics on their width and planarity using depth information. Tracks are kept for a number of frames after occlusion or departure.

In [73], Migniot and Ababsa use a top-down view of a depth camera and propose a 2D model composed of two ellipsoids corresponding to the head and shoulder regions which are obtained by simply thresholding the depth data. The Chamfer distance between the observed regions and the ellipsoidal models is then used to assign the particle weights in their particle filtering tracker. In case of multiple persons in the scene, an independent tracker is created for each target.

Choi *et al.* [71, 72] use particle filtering with Reversible Jump Markov Chain Monte Carlo (RJ-MCMC) sampling to track multiple people simultaneously, as well as static non-human objects (obstacles) and the camera's position. Given the positions and velocities of all tracked targets and the results from generic person detectors applied to ROIs, at each iteration a move is attempted to initialise, delete or update a trajectory. The moves are sampled from the space of possible moves, one at a time, and the likelihood of the modified solution is estimated. Moves are accepted or rejected similar to MCMC sampling until the chain converges. The moves are guided by the probability of continuous tracking, based on a smooth target's motion constraint, which may also account for people interactions [72], and the probability of being a human, as computed by a combination of HOG-based human detection, face and motion detection, skin colour and 2D shape recognition. While Choi *et al.* [72] accounts for the person's appearance in the likelihood, by computing the distance from a target-specific appearance-based mean-shift tracker [88] that uses colour information, Choi *et al.* [71] do not use any appearance model. The appearance model for the tracker in [72] is static though and built from a small number of consecutive frames, in order to minimise tracking drifts.

In the pedestrian tracking system presented by Gao *et al.* [74], a layered graph model is used to estimate pedestrian trajectories in RGB-D sequences. The colour-based classifier of [40] is used to detect target candidate regions from which several features such as 3D position, appearance, and motion are extracted. The layered graph nodes represent the detected regions, and the edges the feature similarity. By minimising the cost function of the graph, using a heuristic searching algorithm, the pedestrian trajectories are obtained.

3.3 Summary and discussion

In this section, we presented details of how the MHT pipeline has been implemented for RGB-D data by researchers in this area, especially focusing on trajectory representation and matching. Table 3 provides a summary of the main characteristics of each method. Robust trajectory representation is crucial, especially to solve occlusions, and the majority of existing methods include a dynamic model to cope with it. Only four of the reviewed methods, i.e. [42–45], use a very simple zero velocity motion model base on the information contained in two consecutive frames (top four rows of Table 3). Thus, these methods are more likely to suffer from trajectory ID switches, and from wrong detections being integrated into the tracks. Furthermore, in case of occlusion, a person cannot be tracked any longer and the associated trajectory is automatically discontinued. The rest of the methods reviewed deploy more complex motion and appearance models that take into account the target evolution over time. All the reviewed approaches, i.e. [46–75], rely on a first-order motion model, generally based on Kalman filters – an assumption quite common in human tracking. Appearance models are dynamically updated and are usually based on RGB data. Several methods, such as [42–47, 60, 64, 75], generate them by applying a sliding window on the recent history of the target trajectory to estimate and update their model. Such approaches help towards a consistent trajectory representation and prevent the model from changing dramatically in cases of temporary detection failure over a few frames. Finally, among the reviewed methods we identified seven different works [61, 68–72, 74] based on a modification of the standard tracking pipeline (see the third column in Table 3), where their combined detection and matching stages are directed by trajectories and their motion model representations, rather than by a generic representation of a person.

4 Survey by use of depth information

The works that we review in this paper seek to improve the stages of ROIs selection, human detection, and matching by the use of depth information as an additional cue. In this section, we outline how the use of depth, in combination with RGB information, can improve each and every stage of MHT.

4.1 Use of depth in ROI selection

Amongst the reviewed methods, all those that select ROIs rely heavily on depth and the additional information it provides on the scene geometry to identify the areas where people may potentially be found. As illustrated in the left branch of Fig. 3, we distinguish three categories of depth-based ROI selection methods, i.e. those based on the estimation of the ground plane, those that model the scene's background, and those that detect motion. We now look at these categories in turn, and characterise the reviewed methods based on their ROI selection method – see columns two to four of Table 4.

4.1.1 Use of ground/ceiling plane: The assumption that people are usually located in areas of limited height above the ground plane can greatly reduce the search space for detection, and this strategy has been used in many of the reviewed works. Munaro *et al.* [59] estimate the ground plane using a Hough-based method [89], and select as ROI the volume above the ground plane at typical human height.

Liu *et al.* [50, 51] detect 3D points that are local height maxima, located at a reasonable distance from the ground. ROIs are defined as vertical cylinders of fixed size centred on these maxima. In [52], the same authors filter these positions by using a fast approach that applies typical head sizes and geometry to remove false candidates.

Ess *et al.* [55] estimate the ground plane jointly with object detection in a Bayesian network. The ground plane is inferred from the bounding boxes of detected objects and the depth-weighted median residual between the ground plane estimate in the previous frame and the lower regions of the depth image. Jafari *et al.* [56] first produce a rough estimation of the ground plane based on the known height of the camera, and then they project onto this plane the points that have a relative height of no more than 2 m. 3D points that project in dense areas of the initial plane are excluded, and the remaining points are used to fit a more accurate plane to the ground surface using RANdom SAmpling Concensus [90]. They then classify the remaining points into three different classes (‘object’, ‘free space’, and ‘fixed structure’ that usually denotes walls) based on their height and on their density when projected onto the ground plane. ROIs are searched for amongst the points labelled as ‘object’, by clustering them based on the connectivity of their ground projections and by retaining the clusters that contain a high enough number of points. They are then divided into sub-clusters that are likely to contain single humans, using the quick-shift algorithm [91] that groups points around local maxima in the density of their ground projections. Similarly, Bansal *et al.* [42] also classify the 3D points into ‘object’, ‘ground’, and ‘overhanging structure’ (e.g., walls) using the distribution of heights in the cells of a grid superposed on the ground plane. The associations between these distributions and cells’ labels are learnt off-line by kernel density estimation. Finally, a smoothing is applied to the pixels’ labelling using a Markov random field that penalises neighbouring pixels that have different labels. Pixels labelled as ‘object’ are used in the detection phase to validate candidate detections.

Detecting and removing the ground plane from a point cloud also facilitates the clustering of the remaining points into separate objects, since they are no longer connected to each other through the floor. Bajracharya *et al.* [47] project all 3D points onto a ground plane, presumably estimated based on known camera height and orientation. The resulting map is used to select areas of high density as ROIs of foreground points. Zhang *et al.* [48] exploit the known height of their camera to produce a rough estimation of the ground and ceiling planes, similarly to [56]. Then, at each new frame, they use the previously estimated planes to select 3D points within a distance threshold to the planes, which are used in a RANSAC algorithm to refine the planes’ estimations. After removing the ground and ceiling planes, the remaining points are clustered, first by isolating regions of similar depths around local maxima in the depth distribution, and then, for each region, by extracting connected components in the image plane. Munaro *et al.* [58] estimate the ground plane using a RANSAC-based least square method that is updated at each new frame to compensate for possible movements of the camera. The authors do not provide a detailed description of their RANSAC-based plane fitting stage. After suppressing the ground plane from the point cloud, they cluster the remaining points from their Euclidean distances. To avoid over-segmentation of objects, the neighbour clusters in ground plane coordinates are merged. Humans belonging to the same cluster are separated later in the detection stage, as will be explained in Section 4.2.

Choi *et al.* use a similar strategy in [71, 72] for detection. After ground plane removal, they cluster 3D points and then select the clusters whose heights are within an acceptable range. HOG-based detectors of both the upper and full bodies, and a face detector are then applied to the clusters to generate their weak, initial detection hypotheses. In [72], a skin colour detector, a motion detector, and a detector based on upper body shape from depth are also used. Galamakis *et al.* [49] estimate the ground plane (without stating how) to discard any ROI points obtained by background subtraction that would be located on or close to the ground. Bahadori *et al.* [62] apply off-line calibration to map their fixed stereo camera disparity data to the 3D world coordinate system. Their resulting reference plane is used to track moving objects by using 3D spatial information. A similar calibration is applied also in the stereo system in [67].

4.1.2 Background subtraction: While similar to ground plane removal, the background subtraction strategy has the advantage of producing ROIs that are more likely to contain humans and to exclude static objects. Its drawback is that it requires learning a model of the background, and updating it in case of moving cameras or variable background. In the latter case, people need to be moving in the scene faster than the background model is updated, to be detected as moving objects. A background model that employs depth may be more robust than a colour one to modifications of appearance that are not correlated with changes of the scene's geometry, such as due to illumination variations [9, 92, 93].

In [60, 61], Almazán and Jones initialise a background model from the first few frames of a sequence. The model is then updated progressively by a linear combination of the model's and current depth values where foreground objects are detected, without modifying background areas that are assumed to remain unchanged. The result is that new background objects are eventually added to the model after they enter the scene, with the risk of adding stationary people when they stop moving for a significant amount of time. Foreground points are detected when the difference of their depth value with that of the model exceeds a threshold, which was empirically established in [61], and that accounts for the measured standard depth variation of the sensor as a function of the distance in [60]. Foreground points are then projected onto a coarse horizontal grid, whose cells, which contain a high enough number of foreground points are selected as ROIs. Galamakis *et al.* [49] also detect foreground points based on their difference with the depth values of a background model. No information is provided on the creation and possible update of the model. In a multi-camera setup, a global 3D coordinate system is used, and foreground points are reconstructed using triangular meshes. Triangles that are too close to the estimated ground plane are discarded. A top-down view of this scene – which in effect is a projection onto the floor – is generated using a GPU and used as a 2D map of the ROI clusters.

Muñoz Salinas *et al.* define a background model in [57] as a height map, i.e. the map of maximum height for each ground plane coordinate. This model is built as the median of ten consecutive maps, and it is updated every 10 s. This update rate is chosen empirically based on the observed people's dynamics to reduce the risk of introducing a person who is temporarily standing in the scene into the model. Background subtraction is performed by selecting the points whose height are above the model's value. Foreground points are clustered using their projection on the

Table 4 Characterisation of the reviewed methods based on their use of depth information

Method	ROI selection			Human detection			Matching	
	Ground plane	Background subtraction	Motion detection	3D geometry	2D depth classifier	3D depth classifier	3D tracking	Joint RGB-D description
Bansal *et al.* [42]	✓					✓		
Salas and Tomasi [43]		✓					✓	
Dan *et al.* [44]				✓			✓	✓
Darrell *et al.* [45]				✓				
Han *et al.* [46]			✓	✓				✓
Bajracharya *et al.* [47]	✓			✓		✓	✓	
Zhang *et al.* [48]	✓			✓			✓	
Galamakis *et al.* [49]	✓	✓		✓			✓	✓
Liu *et al.* [50–52]	✓			✓ only in [52]		✓	✓	✓
Luber *et al.* [53] and Linder and Arras [54]					✓		✓	✓
Ess *et al.* [55]	✓			✓			✓	
Jafari *et al.* [56]	✓				✓		✓	
Muñoz Salinas *et al.* [57]		✓					✓	
Munaro *et al.* [58, 59]	✓			✓	✓		✓	
Almazán and Jones [60]		✓		✓			✓	✓
Bahadori *et al.* [62]	✓	✓	✓	✓			✓	✓
Beymer and Konolige [63]		✓		✓	✓		✓	
Satake *et al.* [64]					✓		✓	
Vo *et al.* [65, 66]		✓		✓				
Harville [67]	✓	✓		✓		✓	✓	
Almazán and Jones [61]		✓		✓			✓	
Muñoz Salinas *et al.* [68, 69]		✓		✓			✓	
Muñoz Salinas *et al.* [70]				✓			✓	✓
Choi *et al.* [71, 72]	✓		✓		✓		✓	
Migniot and Ababsa [73]				✓	✓			
Gao *et al.* [74]				✓			✓	
Ma *et al.* [75]				✓	✓		✓	✓

ground plane, and the clusters that occupy a suitably large area and that contain enough points are selected as ROIs. These ROIs are used as human detections for the matching stage. A colour-based face detector initialises new tracks. In [67], Harville applies the mixture of Gaussian-based foreground segmentation method of [86] to their stereo-based RGB-D data. The foreground objects are then projected to a 2D reference plane where occupancy and height maps are generated. These features are then used to track the foreground blobs with Kalman filters in the 2D reference plane.

Salas and Tomasi [43] use the background model introduced by Gordon *et al.* [94] to combine colour and depth in a 4D Gaussian mixture model. Foreground points are detected as those that are more than 3σ away from the nearest background mode, and large clusters of 3D connected components are selected as ROIs. These ROI clusters are validated as humans or rejected in the detection stage by a colour-based HOG detector. Muñoz Salinas *et al.* in [68, 69] use a similar model that was defined in [86], which is updated by excluding points that belong to detected people. The foreground points are projected onto the ground plane for use in the detection and matching stages, and regions around local density maxima in this plane are selected as ROIs. Bahadori *et al.* [62] propose a very simple unimodal background model by exploiting both intensity images and the estimated stereo disparity. The background model is dynamically adapted after a short initial phase where moving objects are assumed to be not in the scene. Moving object blobs are obtained by subtracting the actual data from the background model and then projecting it to the reference plane to be tracked.

Beymer and Konolige [63] employ the stereo-based background subtraction algorithm proposed in [80]. The background model is initialised with an empty scene. The foreground regions are then segmented to extract dominant disparity layers, assuming that different people in the scene may be located at different distances from the camera. The obtained blobs are then processed by the person detection module. Vo *et al.* [65, 66] identify the background areas combining the navigation information of the moving robot and a depth-based occupancy grid. Background areas are excluded from the candidate search space, speeding up the next steps of their algorithm.

4.1.3 Motion detection: For indoor applications, it may be reasonable to assume that moving objects are likely to be human, and to select areas with motion as ROIs for human detection. Our previous discussion on the respective sensitivities of depth and colour to appearance changes for background subtraction also applies here, and motion may be detected more reliably using depth than from colour only. Thus, the authors such as Choi *et al.* [71, 72] detect changes in 3D point clouds of consecutive frames, represented in octrees, following the method proposed in [95]. The motion term in their estimation of human presence likelihood is then the ratio of moving pixels in the candidate region.

Han *et al.* [46] apply the same foreground detection technique as in [60, 61], but they use the previous frame as the background model. Thus, their foreground points selection is equivalent to selecting moving objects between two successive frames. The moving points are then clustered into ROIs based on the continuity of their values in the depth image.

4.2 Use of depth in human detection

Depth information has been found by many authors to provide cues for human detection that are complementary to colour-based appearance information. These cues mostly describe the 3D shape of the target, and they can be taken advantage of by (a) direct comparison against simple geometrical characteristics of a human shape, or (b) through the classification of 2D and 3D features, as detailed next. Columns five to seven of Table 4 summarise the reviewed methods based on their exploitation of these depth cues for people detection.

4.2.1 3D geometrical properties: To speed up the detection process, many authors apply a cascade of small detectors to the ROIs, starting from the most lightweight ones, followed by the more computationally intensive ones on the few remaining candidates not dealt with by the earlier stages. A very fast and popular early detection stage is the assessment of the ROI clusters against simple geometrical constraints that are determined empirically. In [47], Bajracharya *et al.* select ROI clusters based on the expected width, height, and depth variance of a standing adult. Then a classifier on 3D features further refines the selection of clusters that have a human-like shape. In [48], Zhang *et al.* first verify that the height of objects as well as the number of points in their clusters, are within the expected ranges for a human target. Then, a random selection of normals to the cluster's surface vote to discard vertical (e.g., wall) and horizontal (e.g., tables) surfaces. Finally, a HOG and SVM-based detector is used to validate the remaining human candidates using RGB data. In [58, 59], Munaro *et al.* consider that ROI clusters may contain several humans, or a miscellany of humans and background objects. They extract sub-clusters that are likely to contain individual humans by detecting heads, denoted as local height maxima that follow heuristic rules on their distance from the scene floor and on the minimal separation with others. These initial detections are then validated or rejected as humans by a HOG-based detector on RGB data. Vo *et al.* in [65, 66] apply different geometric constraints to limit the search space of their skin and face detector modules. In particular, features like size and height (considering sitting and standing person) are used.

In [70], Muñoz Salinas *et al.* detect the upper body using an Adaboost classifier with Haar-like features in colour images, and then confirm these detections by verifying their width and planarity. For each positive classification, a binary mask of the upper body shape is applied to the depth image in order to compute the mean and standard deviation of the depth inside the template shape. These are used to estimate the probability for human detection, following heuristic assumptions on the expected width and planarity of a person. When confirmed by this test on depth distribution, the new detections are used to initialise new tracks. Ess *et al.* [55] combine the output of a (or any) colour-based person detector with depth-based cues in a Bayesian network, where the object detection probability depends on the probability provided by the colour-based human detector, the probability of human presence given the scene geometry (using ROIs), and the probability of the detected object to be a human given its 3D geometrical properties, evaluated based on typical human height. The detection is also refined around the estimated location of the colour detector by imposing a uniform depth inside its bounding box, when the depth is sufficiently available.

Other works, such as [44, 46, 73] limit their human detection stage to an assessment of the geometry of ROI clusters to achieve an even higher frame rate. In [44], Dan *et al.* detect humans in top-down depth views by selecting local height maxima within a specified range that show the characteristic empirically-determined shape and size of head and shoulders when seen from above. A similar top-down camera approach has been used by Migniot and Ababsa [73] where the head and shoulder area is obtained by thresholding the depth data and fitting two ellipsoids to the identified regions. This model is then used to estimate body and head orientation.

Han *et al.* [46] only evaluate the height of moving ROI clusters, assuming that human-sized objects that move in an indoor environment are likely to be humans. Thus, ROI clusters, which have a height within a specified range that does not change significantly over five frames, are selected as human detections. A similar approach is also presented in [62, 64, 67] where the height of the detected 3D blobs is used to discard moving objects that are unlikely to be people. Similarly, Almazán and Jones [60, 61] select ROI clusters of moving points that have a high density when projected onto the ground. In [61], detections are defined as areas of a pre-defined size around local density maxima in the ground projection map of the ROI points. In [60], a blob detection technique is used, with smoothing and hole filling of the projected points into blobs, as well as filtering out those blobs that have a projected points density below a certain threshold. The authors note that the depth resolution of their sensors decreases with distance, producing an increasing spread of measured depth values around the exact values, i.e. a stretching of blobs on the line-ofsight of the camera. Thus, the blobs are first normalised in a polar coordinate system. The density threshold is chosen as a function of depth, in order to compensate for the perspective effect that decreases the number of points in the blobs with distance. Galamakis *et al.* [49] also select blobs in their top-down 2D view of ROIs. Morphological operations are applied to a binary mask of their 2D view, and blobs that are too small are discarded.

In [68, 69], Muñoz Salinas *et al.* compute the likelihood of human presence in candidate regions (i.e., regions around local density maxima of foreground points projected onto the ground plane), based on the maximal height and the number of points in these regions that follow empirically established expected values and associated standard variations. This likelihood is used both for detecting new people to initialise tracks and for tracking existing targets.

Darrell *et al.* [45] segment the disparity stereo map with a simple combination of a gradient operator and thresholding based on the typical volume occupied by a person facing the camera. Large connected components are then considered as possible human candidates, and head locations are estimated at the top of each connected component. A combination of face and skin detectors is used to rule out false detections.

Liu *et al.* [52] improved the detection performance of their previous algorithms [50, 51] both in terms of accuracy and processing speed by using a cascade of classifiers based on 3D geometry data. They use a very fast filter based on empirical thresholds on the typical human head size. A second classification stage, based on a ring-wedge mask detector [96], is then applied to identify the head and shoulders regions.

Ma *et al.* [75] use a pool of classifiers based on colour and depth data to detect the human body and its articulated parts. The 3D spatial structure of the tracked object's parts is taken into account such that the detector pool learns pre-determined configurations, and hence the system is able to cope with pose variations.

4.2.2 Classifier of 2D features in the depth map: Classifiers that are traditionally used on grey-level or colour images may also be applied to depth data, or disparity maps in stereo imaging, to recognise the 2D shape of a human. Munaro *et al.* [59] apply a Haar-like feature classifier in a cascade to both the colour image and the disparity map, to exploit different and independent features that increase the detection rate and reduce the number of false positives (FPs). Luber *et al.* [53] introduce a variant of the HOG detector for depth maps, the Histogram of Oriented Depth (HOD), that they use in an SVM classifier to compute probabilities which are linearly combined with the ones obtained from a classical HOG-based SVM classifier on RGB data to detect humans. A similar approach is used in [75] where several DPM classifiers [40] trained on HOG features extracted from the depth maps are used in the detection phase.

Template matching in depth images has also been used [56, 63, 64, 71, 72] for recognising the 2D shape of the upper body. Choi *et al.* [71, 72] compute the likelihood of the depth image to contain a person by template matching of the thresholded depth map with the upper body shape. This probability is combined with the output of a HOG-based detector, and of face, skin colour, and motion detectors, to obtain the human presence likelihood term in their tracking algorithm. Similarly, Jafari *et al.* [56] perform template matching of the depth map with a depth template of the upper body. This depth-based detection is used in close-range images, while a colour-based HOG detector allows for detecting people at a further range where depth sensors may not operate satisfactorily. In [63], the binary template is applied in a classic fashion to foreground blobs and a person is detected when the response is above a certain threshold. Satake *et al.* [64] apply a set of three binary templates [83], containing frontal and side views of head and shoulders, to the disparity map. The sum of squared distance criterion is then used to select human candidates. Detections are checked by using an SVM classifier trained on HOG features.

4.2.3 Classifiers on the 3D point cloud: Similarly to [56, 71, 72], template matching of a human shape may also be performed in 3D. Bansal *et al.* [42] adapt a 3D template to the camera viewpoint, before its correlation with the disparity map is computed for each ground plane coordinate. Local maxima in this correlation map, together with neighbouring correlation values above 60% of the associated maximum, are selected as initial detection candidates. These regions are refined by discarding points with divergent depths, and by selecting areas with a high density of edges in the colour image.

Bajracharya *et al.* [47] apply a linear classifier to a number of features derived from the 3D points of detected candidates. Some of these features capture the variance of the height of the points within the candidate, and the object's size and extent. Three rotationally invariant features also account for the eigenvalues of the point cloud's covariance matrix.

To avoid making hard assumptions on the shape of a human body or upper body, Liu *et al.* [50, 51] train an SVM classifier on two features computed from the height and colour distributions of 3D points. Their features are a histogram of the heights of the upper body, and a joint colour and height histogram of the head, respectively. The upper body and head points are found in regions of pre-defined sizes in the ROI clusters. Harville [67] apply a box filter, set by considering the average adult human height and torso width, to the occupancy map corresponding to the segmented 3D foreground clusters. The peak of the response is thresholded to discard FP detections.

4.3 Use of depth in matching

This section reviews how the use of depth information reduces ambiguities for establishing correspondences of detected people against existing tracks through (a) the provision of 3D trajectories, and (b) by enhancing description of the target in combination with colour. These two uses of depth information for matching in the reviewed methods are summarised in the last two columns of Table 4.

4.3.1 3D tracking: The majority of methods reviewed in this survey construct trajectories in the 3D space to facilitate 3D tracking. This allows better handling of trajectory crossings when objects move past each other in the scene in the camera's viewpoint. We now highlight the role of depth position information in the matching stage, which was described earlier in Section 3.

Dan *et al.* [44] place their camera on the ceiling with a top-down view, therefore the 2D coordinate system of the image can be seen as a good approximation of the 2D ground plane coordinate system. Then, they match detected 3D shapes in adjacent frames from their degree of physical overlap. Galamakis *et al.* [49] also track people in a top-down view of the 3D scene rendered from multiple views, by comparing their distance to predicted target positions on the 2D ground plane.

Gao *et al.* [74] employ depth data to build a 3D layered graph model of the scene to solve possible occlusions, and thus, they boost their proposed tracking algorithm. In [43], Salas and Tomasi exploit 3D location information for computing one-to-one correspondences between candidates, by including a term based on their separating distance in their appearance and motion based correspondence likelihood formulation. Similarly, the various authors of [47, 48, 50, 51, 53, 55–60, 62–64] all perform matching by determining the 3D distance of a detected candidate to its predicted position. In [61, 67–72, 75], 3D position predictions are used to initialise the search for targets in the 3D space.

Only three works described in Section 3 do not exploit the 3D trajectory information. In [46], Han *et al.* use similarities in colour, and variations of depth across two adjacent frames, in order to compute matching correspondences, without taking into account the 3D coordinates. In [42], although Bansal *et al.* use 3D coordinates for ROI selection for their human detection stage, and for camera motion estimation, their matching stage is performed in 2D. Similarly, Vo *et al.* [65] implement their compressive tracking and Kalman filter by considering the target's movements only in the image plane.

4.3.2 Joint RGB and depth description: The fusion of colour and depth information allows for more reliable correspondence of detected candidates to tracks. An example of such fusion is in Luber *et al.* [53], where they build their model from a combination of three possible features: Haar-like features in intensity and depth images, and *Lab* colour feature in the RGB image. Several such features are calculated from small rectangles, randomly sized and positioned inside the bounding box of the detected person. A combination of a few of these features is selected by on-line boosting to produce a classifier that attempts to distinguish the tracked person from its surroundings. Liu *et al.* [50, 51] use a joint colour and height histogram of the full body as their appearance model. The likelihood of new detected candidates to match this model is computed using the Bhattacharyya similarity measure. Similarly, Almazán and Jones [60] model people's appearances from a height histogram associated with colour distributions for each histogram bin, approximated by 3D Gaussians in the RGB space. Dan *et al.* [44] assess the correspondence between two detected candidates by linearly combining a Bhattacharyya measure of similarity of their colour histograms, and the overlap of the 3D shapes of both candidates. This last value, in addition to accounting for the distance in the 3D space between the candidates, may also capture the similarity of their shapes if their 3D locations are close enough. Beymer and Konolige [63] use an intensity model and average disparity value to describe a person. Both are linearly updated, and their drift is limited by applying their person detector confidence as a smoothing factor.

Han *et al.* [46] propose the use of depth for generating an appearance model, where a silhouette obtained from depth information helps in isolating the relevant parts of the body from which a colour-based appearance model is built. The neck and waist are identified as local width minima of the silhouette along the vertical direction. They divide the colour image of the person into head, torso, and legs areas. Torso and legs colours are then used to build the appearance model, by concatenating histograms of colour and texture for both regions. Galamakis *et al.* [49] also exploit depth to produce a two-part colour histogram model of upper and lower body, using their textured mesh representation obtained during their ROI selection stage. The mesh is divided into lower and upper body parts at an empirically determined height.

In [70], Muñoz Salinas *et al.* assume planarity of standing people in order to compute a single-valued depth term of the RGB and a depth based likelihood of a target detection. The mean depth of a candidate region is assessed against a normal distribution with mean equal to the predicted target's distance to the camera, and standard deviation chosen heuristically and decreasing with an increased confidence in depth (measured as the proportion of pixels in the region that have a depth measure). Two depth terms are computed for the head and torso separately. The detection likelihood of a target also includes the comparison to two colour histogram-based appearance models for the head and torso, respectively, using the Bhattacharyya measure, and the assessment of the fitting of an ellipse on the colour image at the expected head location, using image gradients.

4.4 Summary and discussion

In this section, we analysed how the use of depth, in combination with RGB information, is used to increase performance of MHT methods. In Table 4, it can be seen how depth information has been used in all the stages of the pipeline. Particularly common is to use depth data to identify candidate regions of interest containing humans. This is usually achieved by identifying the ground plane and then clustering regions perpendicular to the plane [42, 47–52, 55, 56, 58, 59, 62, 67, 71, 72] or by using background subtraction [43, 49, 57, 60–63, 65–69]. Both approaches are valid only under certain assumptions, such as uncluttered scenes (to allow ground plane estimation) or static cameras (for more accurate background/foreground segmentation). ROI selection is essentially a pre-detection stage, applied to reduce false detections and computational demand. The identified ROIs are then validated with more specific human detectors with the majority of them based on 3D geometric assumptions on typical human size, such as [44–47, 49–52, 55, 58–63, 65–70, 73–75], and some based on adapting standard human detectors to handle depth data, such as [53, 54, 56, 58, 59, 63, 64, 71–73, 75]. Only a few methods use complete template models based on 3D information [42, 47, 50–52, 67]. Although the reviewed approaches seem to perform well in typical video-surveillance like scenarios, the majority of these works detect and track humans while they are walking or standing. A huge challenge lies in tracking people engaged in other activities, or maintaining tracking while they undergo drastic pose changes, e.g. sitting to standing, which would be necessary in other applications, such as long-term health monitoring [37]. Depth data is also fundamental at the matching stage to the majority of the methods where detections are assigned to existing or new tracks based on a distance metric, e.g. [43, 44, 47–52, 55, 56, 58–64, 67–72, 74, 75]. Other approaches base their matching strategy on a combination of colour and depth descriptors [45, 46, 49–54, 60, 62, 70, 75]. We believe this latter approach leads to better target description and is more suited to complex environments, such as smart-homes, where e.g. more varied human interaction and pose changes occurs.

5 Considerations on the practical applications of MHT

The methods presented in this survey are, almost always, customised for specific scenarios or application areas by employing assumptions on aspects, such as the position of the camera(s) (e.g., static or mobile, top-down or head-level view), the geometry of the scene, and the generic description of a person. To guide the reader in their choice of RGB-D MHT method, we next outline the conditions of use of the reviewed methods. These are also summarised in Table 5.

5.1 Type of depth sensor

Historically, depth has been mostly obtained from passive stereo cameras, which offered a cheaper alternative to other technologies such as active sensor cameras. Depth from stereo vision is still widely used in MHT methods, such as [42, 45, 47, 55–57, 62–64, 67–70]. The recently introduced and affordable Kinect camera (and those like it) generate depth from structured light and are more convenient to use than stereo vision for indoor scenes, since they do not require calibration and an elaborate computation of a disparity map. Thus, computer vision researchers are increasingly adopting such cheaper and more immediate technology for RGB-D MHT when it can sufficiently serve their purpose, such as in [43, 44, 46, 48–51, 53, 56, 58–61, 65, 66, 71–73, 75].

Most of the methodologies presented in this survey could use passive and active sensors interchangeably, including those that extract features directly from disparity maps, as disparity and depth maps have similar properties. However, the optimal conditions of use for both types of sensors differ significantly, both in terms of depth range and illumination conditions. For example, the depth range of structured light cameras tends to be more limited than that of stereo vision, and they are also more sensitive to infra-red light, which makes them unsuitable for outdoor uses. On the other hand, colour-based stereo cameras require good illumination conditions and they may not operate in dark environments for example. Moreover, the additional processing time required to obtain disparity from stereo data can be critical for real time applications. These particularities were highlighted by Jafari *et al.*, who used both sensors in [56] to track people in close and far ranges. The second column of Table 5 shows the type of sensor used in the works in this survey.

5.2 Camera position

5.2.1 Handling of moving cameras: Some applications rely on static sensors that provide a stable background model, e.g. [43, 49, 57, 60–63, 67–69], especially when this model is static itself and not updated on-line to account for camera movements, as in [43, 60, 61] and we presume in [49]. Some methods attempt to update the background model continuously, e.g. [57, 62, 63, 67–69]. Although these MHT methods did not present any experiments with mobile cameras, the authors of [57] state that their method has been developed with ‘human–mobile robot’ interaction in mind, and that their background modelling technique is especially appropriate for mobile devices. Indeed, these models may be able to adapt to camera motion, provided that the update rate is faster. The implementation of this strategy is not easy, since, as discussed in Section 4.1, an update rate that is too fast would be likely to result in slow people being included in the background model. Thus, as the authors explain, it has to be tuned depending on the application.

On the contrary, methods that do not assume a static or nearly static background can generally be used with a mobile camera, such as a PTZ or one mounted on a mobile robot. Some methods assume that both person and camera motion are smooth, and they treat their combined effects as that of a single speed for the tracked person relative to the camera [48, 58, 64, 70]. Others exploit the global 3D coordinate system provided via the depth dimension in order to track the camera's movements. Choi *et al.* [71] and Vo *et al.* [65] compute the position of the camera using the ROS library [97] in order to project target locations onto the global 3D coordinate system, and Bansal *et al.* [42], Bajracharya *et al.* [47], Ess *et al.* [55], and Jafari *et al.* [56] do the same using visual odometry. In [55], the visual odometry algorithm is improved by feedback from the tracker which helps avoid using areas that are likely to contain moving objects. In their later work, Choi *et al.* [72] estimate both the motion of the camera and the humans in the scene in their combined detection and matching stage. In general, these approaches assume that the camera is moving, at least locally, on a mostly flat ground plane.

Note that the works in [50, 51, 57, 68, 69, 75], although not tested with mobile cameras, perform tracking in a global 3D coordinate system similarly to [42, 56, 71], and we believe could therefore apply the same mobile camera-handling strategy if combined with camera motion estimation. These approaches can be successful in a moving camera scenario if the camera position requirements (discussed in the following section) can be generally met.

The works in [46, 53, 59] also could employ the same ‘smooth relative-speed strategy’ as [48, 58, 70]. The possibility of using mobile cameras with the reviewed methods is indicated in the third column of Table 5.

5.2.2 Handling of multiple cameras: The works in [53, 56, 58, 63, 64, 67, 74] can exploit information from multiple cameras simultaneously and fuse detections from independent cameras at the matching stage. This requires the relative positions and orientations of the cameras to be known or estimated off-line. In particular, [58] has been extended to the multi camera scenario in [98, 99]. This strategy would be accessible to all methods that apply the main MHT pipeline and perform tracking in a 3D global coordinate system.

This multi-sensor data fusion strategy is not possible in works that apply the variation of the MHT pipeline since they do not perform the detection and matching stages sequentially, such as in [61, 68–72]. However, in [60, 61, 68, 69], detection and matching are performed on a global representation of data on the 2D ground plane, which is generated in [60, 61, 69] from the point clouds of several cameras. The methods in [62, 70–72] use 2D colour image-based people detectors, but they track people in a 3D space. Thus, they could use multiple cameras if all the transformations from the individual image spaces to the global 3D space are known. Similarly to [60, 61, 68, 69], Galamakis *et al.* [49] detect and then

Table 5a Continued

Method	Sensor type	Handle a mobile camera	Handle multiple cameras	Camera location constraints	Real-time	Processing hardware	Require flat ground	Other special requirements
Bansal *et al.* [42]	stereo	✓	✗	roughly frontal view	10 fps	CPU Intel Dual-Core	✓	none identified
Salas and Tomasi [43]	structured light	✗	not tested	roughly frontal view	no data	no data	✓	limited to standing people
Dan *et al.* [44]	structured light	✗	not tested	vertical top-down view	55 fps on QVGA stream	CPU 2.4 GHz, 4 GB RAM	✓	limited to standing adults
Darrell *et al.* [45]	stereo	Not tested	✗	roughly frontal view	12 fps	no data	✗	none identified
Han *et al.* [46]	structured light	Not tested	✗	roughly frontal view	10 fps (2 people)	CPU Dual core 2.53 GHz, 4 GB RAM	✗	limited to standing adults –people must be moving to be detected
Bajracharya *et al.* [47]	stereo	✓	not tested	none identified	5–10 fps	no data	✗	limited to standing adults
Zhang *et al.* [48]	structured light	✓	not tested	roughly frontal view	7–15 fps	CPU 2.0 GHz, 4 GB RAM	✓	none identified
Galamakis *et al.* [49]	structured light	✗	✓	multiple views from different angles are desirable	real-time (no exact data)	GPU NVIDIA GTX680	✗	none identified
Liu *et al.* [50–52]	structured light	not tested	not tested	none identified	30–50 fps	CPU i5–2500, 8 GB RAM	✓	may be limited to standing adults
Luber *et al.* [53] and Linder and Arras [54]	structured light	not tested	✓	roughly frontal view	no data	no data	✗	may be limited to standing people
Ess *et al.* [55]	stereo	✓	not tested	roughly frontal view when using HOG based detectors	3 fps	GPU nVidia GeForce 8800 and CPU 2.66 GHz	✓	none identified
Jafari *et al.* [56]	combined stereo and structured light	✓	✓	roughly frontal view	18–24 fps	CPU i7-3630QM and GPU NVIDIA GeForce GT650m, 12 GB RAM	✓	may be limited to standing people in the far range

Table 5b Conditions of use of the presented methods

Method	Sensor type	Handle a mobile camera	Handle multiple cameras	Camera location constraints	Real-time	Processing hardware	Require flat ground	Other special requirements
Muñoz Salinas *et al.* [57]	stereo	not tested	not tested	roughly frontal view	10 fps	CPU 3.2 GHz Pentium IV	✓	people only detected in a close range (by face detector) but tracked on a larger range
Munaro *et al.* [58, 59]	structured light	✓	✓	roughly frontal view	30 fps [59], 26 fps [58]	CPU Xeon E31225 3.10 GHz [58]	✓	minimal separation between people's heads: 30 cm – may be limited to standing adults
Almazán and Jones [60, 61]	structured light	✗	✓	none identified	no data	no data	✗	stationary people may not be detected after a while
Bahadori *et al.* [62]	stereo	✗	not tested	fixed to ceiling, pointing down at 30°	10 fps	CPU 2.4 GHz	✓	may be limited for other camera configurations, 3D coordinate system calibration
Beymer and Konolige [63]	stereo	✗	not tested	parallel to the ground floor	10 fps	no data	✗	none identified
Satake *et al.* [64]	stereo	✓	not tested	none identified	9 fps	no data	✗	developed for wheel-chair applications, camera placed around 1 m height
Vo *et al.* [65, 66]	structured light	✓	✗	none identified	23 fps	CPU 2.4 GHz	✗	developed for robot applications, camera placed around 1 m height
Harville [67]	stereo	✗	not tested	none identified	8 fps	CPU 750 MHz	✓	none identified
Muñoz Salinas *et al.* [68, 69]	stereo	not tested	✓	none identified	15 fps [68] and 100 fps [69] (4 people)	AMD Turion 3200, 1 GB of RAM	✓	none identified
Muñoz Salinas *et al.* [70]	stereo	✓	not tested	frontal view at head level	23 fps (3 people)	AMD-K7 2.4 GHz	✗	limited to standing people
Choi *et al.* [71, 72]	structured light	✓	not tested	roughly frontal view	5–10 fps	GPU	✗	may be limited to adults
Migniot and Ababsa [73]	structured light	✗	not tested	vertical top-down view	40 fps	CPU 3.1 GHz	✗	none identified
Gao *et al.* [74]	not specified	✓	not tested	developed for ADAS applications, camera placed around 1 m height	40 fps on CPU	✗	none identified	
Ma *et al.* [75]	structured light	✓	not tested	none identified	no data	no data	✗	hand labelled data of body parts to train DPM

track people in a common 2D coordinate system. The fourth column in Table 5 indicates which of the reviewed works can (or could) handle multiple cameras.

5.2.3 Requirements on the camera's position and orientation: Methods that use human detectors that are trained on specific view positions and angles, such as HOG trained from roughly frontal views, require similar views of people. This is the case in [42, 43, 48, 53, 56, 58, 59, 70], and also in the implementation of [55], although the authors stress that other colour-based detectors can be used. Similarly, the works in [57, 71, 72] employ face detectors, and require a roughly frontal view of the face to be visible in a significant number of frames. The methods in [45, 57, 70] were specifically designed for a camera located at head (or just under head) height. In particular, the work in [70] assumes the human shape as seen by the camera can be approximated by a vertical plane. Han *et al.* [46] also require a frontal view for analysing human silhouettes, as explained earlier in Section 4.3. The methods in [44, 73] operate on a top-down view due to their specific detection strategy centred around monitoring humans seen from above. In [49], a sufficient coverage of the scene by multiple cameras at various viewing angles is preferred to produce 3D textured meshes of humans. In [62], the camera is placed on the ceiling at an angle of 30°, so as to reduce occlusions as upper body parts are always visible. A similar camera position is used in the e-health application presented by Ma *et al.* in [75]. However, their proposed system, based on different DPM classifiers, is able to deal with considerable variations of human body pose, hence ensuring also a certain invariability to camera viewpoints. Beymer and Konolige [63] propose a 3D motion model based on the assumption that the stereo camera is placed parallel to the ground floor. In [64], the system has been specifically designed for a wheel-chair navigation system, and the stereo camera is placed at an approximate height of 1 m. The various requirements and limitations of the camera position and orientation are summarised in column five of Table 5.

5.3 Speed of computation

The works in [43, 53, 60, 61, 75] provide no computational information. Harville *et al.* [67] report a processing rate of 8 fps, however, this is obtained using obsolete hardware and it would dramatically improve if tested on current workstations. The rest of the methods we review claim real-time performances, with the exception of Ess *et al.* [55], who report a running time of 300 ms per frame on a GPU, plus an additional (off-line) 30 s for the colour-based human detector. Their method can be used with other, more efficient, colour-based detectors.

Running times and the hardware platforms used, when available, are reported in columns six and seven of Table 5. Methods that use stereo vision suffer from the overhead of deriving disparity maps, while depth information is readily available from structured light-based sensors. Some authors, such as Bansal *et al.* in [42], speed-up this computation using a GPU.

Works such as in [46, 68, 69] report performances that vary significantly according to the number of people being tracked. This is particularly the case in methods that use multiple trackers for individual people, such as the 3D Kalman filter in [64] or particle filters in [68–70, 73]. Such methods also need to establish a trade-off between the number of particles used and the accuracy and robustness of tracking. Jafari *et al.* [56] exploit both depth and colour information in complementary distance ranges, and speed-up the total process from 33 (on GPU) to 18 fps by applying the computationally intensive colour detector only in far ranges (over 7 m) where the depth-based detector does not operate.

Finally, Liu *et al.* [51] report a processing rate range of 30–50 fps, without the use of GPU hardware, for their detection and tracking system. In addition, in their more recent work [52], they boosted the detection phase by using a cascade of classifiers on top of their depth-colour histogram model. This meant that their detection module can operate in a range of 77–140 fps, however, no rate is given for the entire detection and tracking processing.

5.4 Specific constraints

5.4.1 Flat ground: Methods that detect ROIs based on an estimation of the ground plane, as detailed in Section 4.1, cannot handle environments where the ground cannot be approximated by a plane. This is the case, e.g. of staircases, where Munaro *et al.* report worse results in [58]. Similarly, the methods in [42, 56] classify the scene into general categories that include a flat ground and vertical structures, and would most likely not generalise well to a staircase environment.

In [44, 73], people are detected by thresholding the distance of their head to the camera, which has to be within an acceptable range. Therefore, although there is no hard constraint on ground planarity, varying ground level can influence the head-to-camera distances significantly.

Methods, such as those in [47, 49–51, 57, 60, 61, 68, 69], project detections onto a flat ground plane. In [47, 49, 60, 61], ROIs are not selected based on height from this plane [In [47], ROI clusters are selected based on their absolute height, and in [49], only a few points close to the ground plane are discarded, not the full ROI clusters.], so the flat ground assumption does not need to correspond to reality. However, this is not the case in [50, 51, 57, 68, 69] where people have to be located in a relatively narrow band above the floor to be detected. In [62, 67] a reference plane is used to track 3D blobs in a real-world coordinate system. As the camera and real-world scene are calibrated, the reference plane does not have to be necessarily flat. Column seven in Table 5 indicates which of the methods operate only in ground plane scenarios.

5.4.2 Constraints on pose: Several works, e.g. [44, 46, 47, 50, 51, 57–59, 62, 63, 65, 67–69, 71–73] select ROIs based on height and volume assumptions derived from a model of a standing adult person. Such methods may not be able to detect and track, e.g. children, adults with abnormal heights, and sitting people, if the acceptable ranges for height and volume are not chosen appropriately. This is the case, e.g. for Choi *et al.* [71] and Han *et al.* [46], who filter heights in ranges of 1.3–2.3 and 1.5–2 m, respectively. Other authors, such as Zhang *et al.* [48] and Liu *et al.* [50, 51], accept quite larger range of values (0.4–2.3 and 0.6–2 m for height, respectively), to prepare their ROIs to handle children or people who may not be standing. Ess *et al.* [55] suggest the possibility of detecting children by increasing the standard deviation of their normal height distribution.

Methods that use full-body detectors such as HOG and HOD, i.e. [43, 53, 56, 58, 59, 65], may also struggle to detect sitting people if these detectors are trained on standing people only. To alleviate this shortcoming, Choi *et al.* [71, 72] combine full-body and upper-body detectors, in order to cope with both occlusions of the lower part of the body and various poses. Multiple different DPM detectors based on HOG features are used to deal with deformable body pose (e.g. sitting, bending etc.) in [75]. Jafari *et al.* [56] also apply an upper-body detector based on a depth template, as described in Section 4.2, and Liu *et al.* [50, 51] detect people based on a model of height of the upper body. Similarly, Zhang *et al.* [48] use a poselet-based detector [39] to identify body parts, and Bansal *et al.* [42] perform matches of several local contours, thus allowing the detection of people in arbitrary poses. In [70], detected candidates are checked against a planar model of a standing person using depth information. Thus, sitting people would be rejected by the human detector.

5.4.3 Miscellaneous: Munaro *et al.* [58, 59] distinguish people in close interaction based on the separation of their heads, which needs to be at least 30 cm. This constraint is generally easily respected, especially in a public environment. Han *et al.* [46] detect people based exclusively on movement and on their height (see Section 4.2). Therefore, motionless people would not be detected. In [57], new people are detected by a face detector, and the authors set the detector to only scan the close range area (0.5–2.5 m) to speed-up the process. Tracking is still performed on the full space, but would be initialised only after the person enters this detection region. Constraints on body pose and other miscellaneous constraints are stated in the last column of Table 5.

5.5 Preprocessing the depth map

Depth maps tend to suffer from noise and areas of missing values, whatever the sensor, and result in inhomogeneous point clouds. A few of the reviewed works correct these deficiencies before exploiting depth information.

5.5.1 Depth map denoising and completion: Zhang *et al.* [48] suppress outliers from the 3D point clouds by removing the points that have only a few neighbours. In Galamakis *et al.* [49], the overlapping views of the structured light sensors create interferences that add noise to the scene. The authors report that in their setup this noise is predominantly on the ground plane and negligible on humans, and use an estimate of the ground plane to eliminate any points close to the ground in their ROI selection stage. The method proposed by Dan *et al.* in [44] detects people based exclusively on their 3D shape in the depth map, and so can underperform when faced with missing depth values. To close the holes in their map, they first apply morphological operations to the binarised depth values, obtained by thresholding the heights above the ground plane, and then a nearest neighbour interpolation is used to recover the depth values in the gaps that were filled in the binary map.

5.5.2 Voxel grid filtering: Works, such as Jafari *et al.* [56] and Almazán and Jones [60], which consider the number of points in ROI clusters have to take into account the perspective effect that makes the density of points depend on the distance to the camera. Munaro *et al.* [58] address this issue to produce a homogeneous density of points in the volume space by re-sampling their 3D point cloud before ROI detection and clustering. Thus, they ensure that the number of points in a cluster depends only on the size of the object within it, rather than on a combination of its size and distance to the camera. In [65], Vo *et al.* reduce their initial search space by subsampling their colour and depth images.

5.5.3 Fusion of point clouds: Jafari *et al.* [56] obtain richer 3D point clouds by combining those obtained over a time window of five to ten frames, using their mobile camera motion, estimated by visual odometry. In [69], Muñoz Salinas *et al.* merge the ground plane representation of overlapping views from several sensors by retaining the points in a global coordinate system that have the highest confidence. Galamakis *et al.* [49] fuse foreground points of overlapping views in a global coordinate system during their ROIs selection stage. Note that works, such as [60, 61], which fuse foreground points of non-overlapping views, do not require specific manipulation of the point clouds and only need to calibrate their cameras' positions and orientations.

6 Online resources: benchmark datasets and software

In this section, we provide an overview of publicly available benchmark datasets and source code, with a summarised list provided in Tables 6 and 7, respectively.

6.1 Dataset resources

The ETH dataset from [100] contains stereo sequences obtained by a pair of AVT Marlin F033C cameras mounted on a mobile platform. Images are acquired at 640 × 480 resolution at 14 fps. The corresponding disparity maps are obtained by using the stereo algorithm presented in [102], but are not available for download. The dataset is composed of five sequences recorded in very busy pedestrian zones, and these have been manually annotated every four frames by labelling only pedestrians that are greater than 60 pixels in height. The ground truth does not contain tracks IDs, hence only the detectors' performances can be obtained. An example of the ETH stereo data is shown in Fig. 4. The ETH dataset has been also used to validate MHT algorithms based on the use of colour data only, especially in the MOT challenge [103].

The dataset presented in [25, 53] is obtained by using static cameras, positioned 1.5 m high, in a large university hall. We refer to this dataset as the University Hall Dataset (UHD). An array of three Kinect devices, with non-overlapping fields of view to avoid IR interferences, is used to record people passing through the university hall. Due to the Kinect sensor's range limitations, depth data is not available beyond a certain range in the hall. The image resolution is 640 × 480, with synchronised sequences recorded at 30 Hz. This rather small dataset is composed of three sequences each ∼1130 frames in length. There are 3021 instances of people, and 31 tracks have been manually annotated as ground truth (for detection and tracking). An example of this UHD data is shown in Fig. 5.

The RGB-D tracking dataset presented in [72] contains two different scenarios captured with Kinects, one static and one mobile. We refer to this dataset as the StanfordRGB-D dataset. The first scenario, the Kinect office, contains 17 sequences with the camera placed 2 m high in an office. These videos contain different occlusion scenarios and human poses. The second scenario, the Kinect mobile, contains 18 sequences with people performing daily activities in offices, corridors, and hallways. These sequences were recorded with the camera mounted on a mobile platform (a PR2 robot). In both sets, human positions are hand-annotated (four images every second) with bounding boxes around their upper bodies – hence, both detection data and targets ID are included. Ground truth odometry information of the camera's location in 3D space is also available. An example of the StanfordRGB-D dataset for the static camera scenario is given in Fig. 6.

Table 6 RGB-D benchmark datasets – in all cases resolution = 640 × 480 and frame rate = 30 Hz (except ETH [100] = 14 Hz)

	Device	Number of Sequences	Number of Frames	Ground truth	Comments
ETH [100]	stereo device (AVT Marlins F033C)	8	5017	YES manually annotated detection only	minimum pedestrian size 48 pixels, calibration and odometry data available
UHD [25, 53]	Multiple Kinect 1 static	3	3390	YES Manually annotated detections and 31 people tracks	part of the scene out of depth range
StanfordRGB-D [72]	Kinect 1 static and mobile	17 (static) 18 (moving)	≃157,500	YES manually annotated four images per second, detections and tracks	Camera positioned 2 m high for the static sequences. Ground truth odometry of camera location available
KTP [58, 101]	Kinect 1 static and mobile	5	8475	YES manually annotated and infrared marker groundtruth, detections and tracks	device placed at robot level, sequences with different complexity
SD [50, 51]	Kinect 1 static	1	3000	YES manually annotated one image per second, only detections	Camera positioned 2.2 m high, 30° inclination. Cluttered and crowded scenes
KingstonRGB-D [60]	Multiple Kinect 1 static	6	≃6000	YES manually annotated, detections and tracks	Cameras positioned around 2 m high. Cameras'calibration matrices available

Table 7 Software resources

	Processing arch.	Processing rate, fps	Dependencies requirements	Availability
Luber *et al.* [25, 53][a]	CPU	—	Eigen2	Partial: depth-based detector module and integration with Kinect not available
Choi *et al.* [72][a]	CPU+GPU	5–10	Opencv, boost	Partial: depth-based detector module not available
Munaro *et al.* [58, 101]	CPU	23–28 detector only, 19–26 detector + track	boost, eigen3, flann, Openni, PCL	Partial: only detector module available, manual initialisation of ground plane required. Integrated with PCL
Jafari *et al.* [56][a]	CPU+GPU	24 without HOG-GPU 18 with HOG-GPU	FOVIS, Openni x64, CImg, CUDA, KConnectedComponentLabeler, boost, eigen3, ImageMagick++	Partial: missing modules for stereo data processing to estimate tracked camera position and projections. GPU–CPU processing to enable far distance detections
Munaro *et al.* [98, 99]	CPU	30	ROS, PCL	Full for live camera network, but no plug and play module to test offline data. Multi-camera and multi-device support for calibration and synchronisation

[a] The authors of the paper were contacted (who responded) to clarify details about their software release.

Fig. 4 *ETH stereo dataset example*

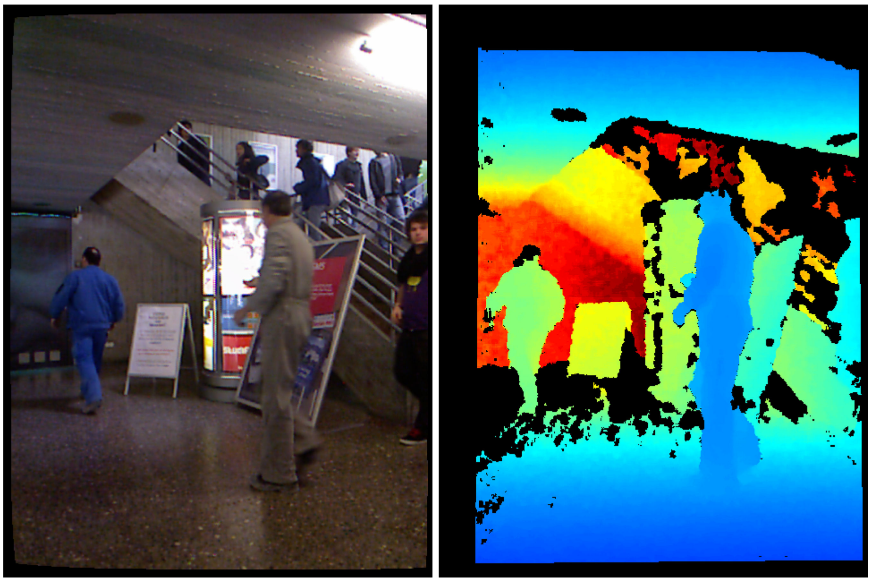

Fig. 5 *RGB-D UHD dataset example*

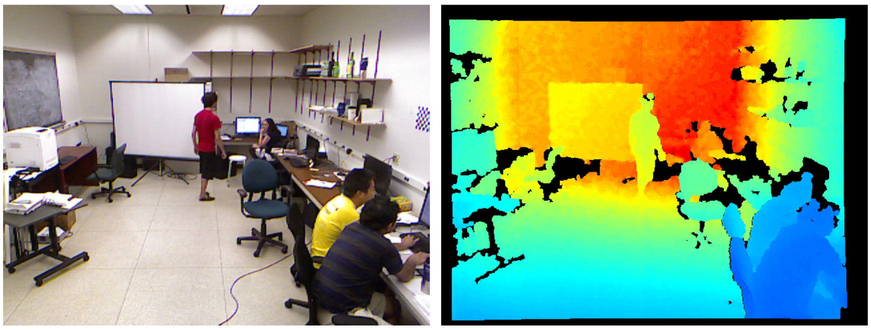

Fig. 6 *StanfordRGB-D dataset example*

The Kinect Tracking Precision (KTP) dataset proposed in [58, 101] was acquired with a Microsoft Kinect, at a resolution of 640 × 480 and recorded at 30 Hz, on board a mobile platform. It contains 5 different sequences, exhibiting 14,766 instances of humans in 8475 frames. Both manually labelled 2D image and metric ground truth (for detection and target IDs) are provided, and 3D positions are also available since an infrared marker was placed on every subject's head. Fig. 7 shows an example frame from the KTP dataset.

The dataset in [50, 51] contains ten sequences recorded with a Kinect sensor in an indoor (shop) environment, and we refer to it as the SD dataset. The device was mounted at 2.2 m high with about a 30° tilt towards the floor and the sequences were recorded at 30 Hz at a resolution of 640 × 480. The groundtruth, produced once every 30 frames, does not contain target ID information, and thus only detection accuracy can be tested. An example of the SD dataset is displayed in Fig. 8.

A recent dataset introduced in [60] was obtained using three static Kinect devices, all positioned at about 2 m high in a lab with non-overlapping views. We refer to this multi-camera dataset as the KingstonRGB-D dataset. The sequences contain people moving in the lab, individually or in numbers, with paths crossing at times. The dataset comprises six 1000-frame sequences split equally into a training set and a test set. The cameras' calibration matrices and the data to obtain a wider planar map of the scene are also available. The ground truth supplies detections and target IDs for all the different views. An example of the KingstonRGB-D dataset is shown in Fig. 9.

To recapitulate, only the ETH dataset [100] is based on stereo data, while the others presented here have all been recorded using the Kinect and hence contain only indoor scenes. As also

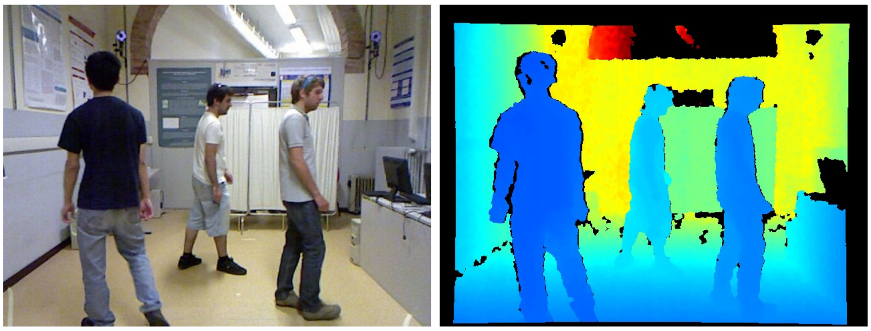

Fig. 7 *RGB-D KTP dataset example*

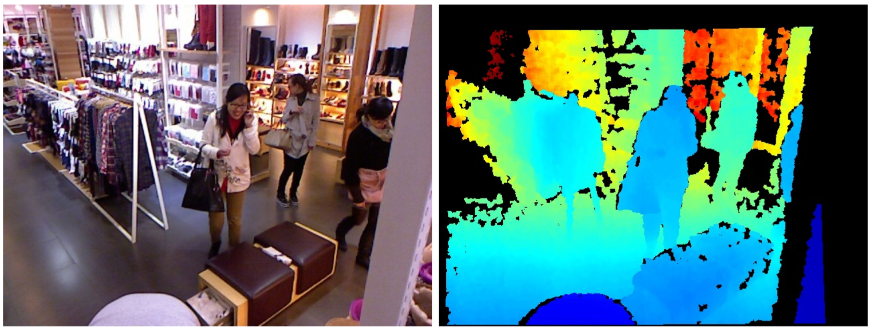

Fig. 8 *RGB-D SD dataset example*

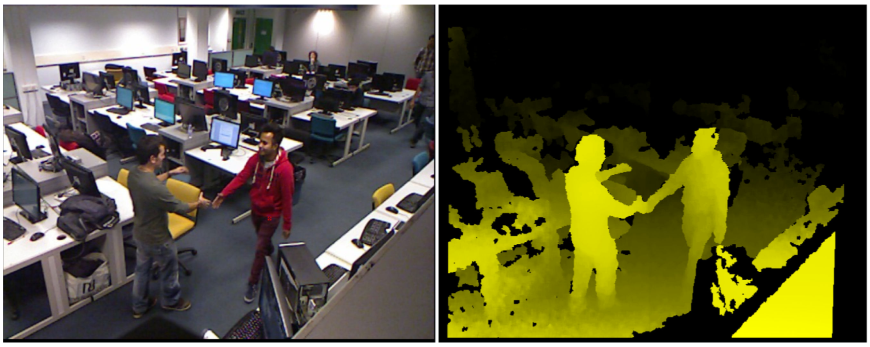

Fig. 9 *RGB-D KingstonRGB-D dataset example*

highlighted in Section 5.2, in most of the proposed approaches the position of the camera facilitates the acquisition of frontal views of the moving human. Only in the dataset presented in [50, 51, 60, 72] is the camera placed close to the ceiling, giving a top-oblique view of the scene. This setup yields a more realistic example of a typical surveillance camera location.

6.2 Software resources

There are only very few software resources for RGB-D MHT tracking publicly available. A list of these can be found in Table 7. The source code for the method presented in [53] is available for Linux platforms – however, it does not provide all the functionalities described in [53]. For example, the code corresponding to the detection module based on HOD features (described in Section 4.2) has not been released, but the code for the tracking core, based on MHT (Section 3.1) and the online adaboost detector (Section 3.1) are available and are integrated with a laser range scanner. Despite the source code being incomplete, this resource is still very useful as the missing HOD module can be developed by the interested researcher starting from one of the available colour-based HOG versions and then by using the UHD data to train the classifier. The code can also be easily ported onto a Windows environment as its only dependency, Eigen, is available for both Linux and Windows. The authors of [53] do not provide details of computational performance of their method.

The authors of [72] provide the source code for their tracking module, based on an RJ-MCMC particle filter (Section 3.2), but some of their proposed detectors (Section 4.1), in particular their depth-based silhouette (Section 4.2), are not made available or integrated into the main processing loop of their software. Their method also runs on Linux, but we have ported it to Windows as the main dependencies needed to run it, OpenCV and Boost, are available on both platforms.

Munaro *et al.* in [58, 101] have integrated the detector stage of their tracking methodology into the Point Cloud Library (PCL) [104]. This integration with such a widely used library, is one of the main advantages of this source code as it can be easily ported to both Windows and Linux. They indicate a processing throughput of 19 fps on an Intel i5-520M 2.40 GHz CPU and 26 fps on an Intel Xeon E31225 3.10 GHz CPU; in both cases a 4 GB DDR3 memory was used. These remarkable results can be associated with the specific optimisation approaches used, e.g. as stated in Section 5.5, the algorithm in [58] dramatically reduces the point cloud size by a subsampling procedure and by eliminating ground plane points as described in Section 4.1.

Recently, this software package has been upgraded by Munaro *et al.* [98, 99] to support a multi-camera RGB-D system. The new software library, OpenPTrack, is compatible with Microsoft Kinect and Mesa SwissRanger and can achieve real-time tracking of people at 30 Hz. Each sensor stream independently detects people, while tracking is performed in a central unit by fusing the contribution of all the network nodes. The detection and tracking software, however, is not easily accessible as a plug and play module. The algorithms presented in [58, 59, 101] are included in OpenPTrack.

Table 8 ETH dataset detection results

	LAMR	Modified_LAMR
Ess *et al.* [55]	**0.645**	0.527
Bansal *et al.* [42]	—	0.612
Choi *et al.* [72]	—	**0.434**
Munaro *et al.* [58, 101]	0.663	0.592

Jafari *et al.* [56] provide the source code for their method which imposes different dependencies as shown in Table 7. The OpenNI library is used as their interface for both the Kinect and Asus Xtion sensors. Their system is based on both a short-range depth-based human detector [105] running at 24 fps on a single CPU and a far-range HOG-based human detector [82] (see also Section 3.1) which must run on a GPU. Their experimental results were obtained using an Intel i7-3630QM with 12 GB RAM and an NVIDIA GeForce GT650m GPU. The main advantage of this software resource is the possibility to activate the two different detection modules independently. This adaptability offers the opportunity to balance accuracy and processing speed, and the possibility to avoid using modules when not necessary, e.g. the longer range detector in an indoor environment. Note the system requires calibration and odometry data to operate.

7 Comparative evaluations

We now consider how various works have used the datasets introduced in the previous section for evaluating their methods. Two types of comparative evaluations are presented next – the first attempts to compare different published works on a publicly available dataset (or part of it), while the second presents within-method comparative results by switching one or more of the method's components off. Unfortunately, we are not able to compare the results of the software listed in Table 7 due to the limitations outlined in the previous section.

7.1 Inter-method comparative results

Two publicly available datasets have been used by more than one published work, the Eidgenössische Technische Hochschule and the UHD datasets.

7.1.1 ETH: The stereo ETH dataset has been used by [42, 47, 55, 56, 58, 72] to test their specific methods, with many utilising different sequences, and metrics, for evaluation. Bearing this in mind, Table 8 displays the log-average miss rate (LAMR) [10] results which are focused on people detection accuracy for the ETH-Bahnhof sequence of the ETH dataset. LAMR is computed by averaging the miss rate versus the FP per image (MR-FPPI) graph in the range $[10^{-2}, 1]$ in the FP axis. In particular, we use the reported MR-FPPI results in [42, 55, 58, 72] to extrapolate the LAMR values (second column of Table 8). Note, the MR-FPPI results reported in [42, 72] are not available for the entire range, and for this reason we estimate the Modified_LAMR (third column of Table 8) by considering a smaller interval in the range of [0.056…1]. The best Modified_LAMR result is obtained by Choi *et al.* [72].

The sequence ‘Sunny day’ of the ETH dataset is used to test the methods proposed in [42, 47, 56]. The results are reported with graphs of ‘recall versus FP per image’. As reported by Jafari *et al.* [56], their method achieves the best results – e.g. for a fixed FPPI value of 0.5, their recall rate is $\simeq 0.85$ which is greater than $\simeq 0.7$ by Bajracharya *et al.* [47] and $\simeq 0.5$ by Bansal *et al.* [42].

Table 10 StanfordRGB-D dataset detection results

Method		LAMR	
		Static camera	Mobile camera
Choi *et al.* [72]	full	0.60	0.601
	no depth	0.844	0.858
	no Hog	0.657	0.695
	no Face	0.612	0.608
	no skin	0.626	0.629
	no motion	0.592	0.637
Vo *et al.* [65]		**0.52**	**0.514**

7.1.2 University Hall Dataset: The UDH dataset was used to evaluate the methods proposed in [53, 58, 72, 75] tested with only colour-based features. Tracking performance is reported by considering the CLEARMOT metrics [106] for which two indexes are given – the multiple object tracking accuracy (MOTA) index estimates the tracking error by considering the FNs, FPs and mismatches, and the multiple object tracking precision (MOTP) index which measures how well exact target positions are estimated. FP, FN ratios and identity switches are also reported. Table 9 shows the results reported in [53, 58, 101]. While the method proposed by Luber *et al.* [53] guarantees best performance in term of MOTA, FP and FN, the method of Ma *et al.* allows to dramatically reduce the number of identity switches. Similar top performance is obtained by Munaro and Menegatti in [101] who state that their poor performance on this dataset is mainly due to mis-detection of people on the staircase sequence as it breaks the flat ground assumption that is central for this approach (as described in Section 4.1). When ignoring these mis-detections in the stairs, as well as re-annotating some ground truth which were believed to be incorrect, the authors reported an improved MOTA result of 88.9%. This result cannot be used for comparative evaluation here as the ground truth is modified. For the MOTA metric, the methods presented in [72, 75] lead to significantly low scores.

7.1.3 StanfordRGB-D: The StanfordRGB-D dataset has been used by its creators in [72], and by Liu *et al.* [51] and Vo *et al.* [65], to evaluate the detection accuracy of their proposed approaches to MHT. Choi *et al.* [72] present their results in terms of MR-FPPI and the LAMR for the two different scenarios (fixed camera and mobile platform), obtained by averaging across the different sequences. Table 10 summarises the LAMR values, reported in [72], for the two scenarios: static camera (second column) and moving camera (third column). After the full method in the first row of the table, each row reports the results obtained by turning off one of the detectors (see Section 3.2). As shown, the depth cue is the most important for the system, where the LAMR value increases by around 0.25 in both scenarios when this detector is not employed. The HOG-based detector is also significant to the final performance of the system, while the impact of the other detectors is less. The full system obtains the same LAMR value of around 0.6 for both scenarios. The recent results obtained by Vo *et al.* [65] show that for both scenarios (moving and static cameras) the proposed approach outperforms the results obtained by the RJ-MCMC method in [72]. Liu *et al.* [51] only report their results in terms of MR-FPPI and thus it is not possible to precisely calculate Modified_LAMR values.

Table 9 UHD dataset tracking results

	MOTP, %	MOTA, %	FP, %	FN, %	ID Sw.
Luber *et al.* [53]	—	**78.0**	**4.5**	**16.8**	32
Munaro *et al.* [58, 101]	**73.7**	71.8	7.7	20.0	**19**
Choi *et al.* [72] (only colour)	57.6	20.2	20.9	57.6	**1.28**
Ma *et al.* [75]	70.4	26.9	13.9	57	**2.1**

Table 11 KTP dataset tracking results

	MOTP, %	MOTA, %	FP, %	FN, %	ID switch
full (HSV) [101]	84.2	85.8	0.7	12.5	**53**
no sub. [101]	84.2	83.0	**0.6**	15.9	56
full (RGB) [101]	84.2	86.1	0.8	12.7	60
full (CIELab) [101]	84.2	86.5	0.9	**12.2**	56
full (CIELuv) [101]	**84.2**	**86.7**	0.9	12.9	65

Table 12 SD dataset tracking results

	Lost tracks	ID switches
depth–colour tracker [50, 51]	**13**	**1**
depth tracker [50, 51]	14	15
colour tracker [50, 51]	15	6

7.2 Intra-method comparative results

Three datasets have been compared on variations of the same method providing comparative results.

7.2.1 KTP: The KTP dataset was prepared and used by Munaro and Menegatti [101] to evaluate the tracking performance of their method [58] with the CLEARMOT metrics. In Table 11, we present some of the results reported in [101]. The first row shows the results obtained by the algorithm presented in [58] by using all its components, including the sub-sampling strategy described in Section 5.5. This strategy guarantees real-time performance (see Section 6) at little loss of performance in comparison to when not subsampling the point cloud (second row). The last three rows contain the results for using different colour spaces as input to the colour classifier (see Section 3.1). The authors claim that the best results are obtained with the HSV colour space, especially for the reduction of identity switches.

As previously mentioned, in [99], Munaro *et al.* present an extended software library containing the algorithms presented in [101] that is able to cope with different depth devices. In [99] they also evaluated the performance of their tracking algorithm by using different devices. They present results for three different sequences recorded with both the Kinect (based on structured light technology) and the recent Kinect V2 (based on time-of-flight technology), and one other time-of-flight device (SR4500). The time-of-flight sensors both did better than the first generation Kinect, while Kinect V2 performed better than the SR4500 due to its higher resolution depth representation.

7.2.2 SD: In [50, 51], the authors first compare colour-only to the depth–colour detector, reporting the value of the break-even point (i.e., where precision is equal to the recall in the PR curve) of 93% when the depth–colour detector is used, compared to 52% when the standard colour-based HOG detector is employed. The tracking results presented by the authors in [50, 51] are reported in Table 12. They show how the proposed method based on depth–colour combination guarantees a better performance, for both lost tracks and ID switches, with respect to the proposed tracker relying only on colour or depth solely to solve the data association problem.

7.2.3 KingstonRGB-D: This dataset has been used only by its creators in [60] to estimate the performance of their tracking method. They evaluate their methods by considering some of the metrics proposed in [107], i.e. correct detected tracks (CDT), false alarm tracks (FAT), track detection failure (TDF), and ID switches. In addition, the F1-score metric is used as a summarising metric. The results obtained by the authors in [60] are reported in Table 13 and demonstrate that the proposed tracking strategy based on a colour and depth appearance model (described in Section 4.3) is able to outperform an alternative tracking strategy that uses depth data only.

Table 13 KingstonRGB-D dataset tracking results

	CDT	FAT	TDF	F1	IDSw.
depth/spatial model [60]	27	18	35	0.5	60
colour+depth [60]	**40**	**5**	**19**	**0.77**	**15**

8 Challenges

In summary, depth data is a fundamental cue that can bring more reliability to MHT methods, but there are many challenges that the vision community needs to address to advance this area further.

To start with, it is important that this research community can generate for itself standard and diverse datasets to cover all kinds of application areas (e.g., surveillance, health monitoring, pedestrian tracking etc.) that can help it to evaluate old and new algorithms in a consistent fashion. However, this predicates on researchers to make their data and software more widely available, and report their methodology and processes in a reasonably reproducible fashion.

There are still many challenges where depth can be explored further. For example, depth can be a fundamental tool for better (partial) occlusion detection while tracking, so we should expect to see some creative uses of depth information to achieve higher accuracy rates for MHT – perhaps even in busy scenes depending on the camera viewpoint. Developments on resilient part-based tracking of humans will also help in better occlusion handling.

Depth sensors' accuracy is limited to a certain range of distance, hence another important challenge is handling of scale while tracking. The better occlusion and scale handling, the greater the diversity of applications colour appearance and depth-based tracking can contribute to. Indoor applications, such as in-home health monitoring may be well served by active sensors, whereas outdoor or longer range applications, such as surveillance monitoring, would be handled by passive sensors. Improvements to the detection range and technology of active sensors will help to overcome shortcomings in scale handling in indoor environments, as already evidenced by the new Kinect V2 compared to the first generation Kinect.

Humans have articulated parts so they will be observed in a variety of poses in various scenarios, compounded by the fact that they also interact with each other. The majority of current works, if not all, track humans while they are walking or standing. A huge challenge lies in tracking people engaged in other activities, e.g. to monitor their routine for health monitoring, or maintaining tracking while they undergo drastic pose changes, e.g. if they bend down, sit down and then stand up, or perform certain prescribed exercises.

Other challenges include more regular issues, such as developing better features and more elaborate adaptive and dynamic methodologies (e.g., such as by applying deep learning techniques).

9 Conclusion

This survey provided an overview of all existing works known to us that fuse RGB and depth data for MHT. It is a snapshot of the current works in the last few years, along with data and software resources, as well as some comparative results. MHT is still a relatively young but quickly progressing area where the availability of cheap depth sensors is a huge contributing factor to the regeneration of old, and creation of new, human detection and tracking methods. The analysis and the results reported in the review demonstrates that depth data is fundamental to boost RGB-only MHT methods in terms of both detection accuracy and tracking reliability as depth data introduce very powerful spatial cues (3D shapes and 3D locations) that are also less sensitive to scene illumination conditions. Moreover, the combined colour–depth appearance model can be used to describe humans also at region level. Further, despite the processing of the additional depth cue, real-time performance can still be maintained, as depth data allows for the significant reduction of the search space for both detector and tracker modules, even when simple heuristic rules are used.

10 Acknowledgments

This work was performed under the SPHERE IRC project funded by the UK Engineering and Physical Sciences Research Council (EPSRC), Grant EP/K031910/1. No new datasets were generated during this work. The details of all the publicly available datasets referred to in this work can be found in Section 6.

11 References

[1] Wang, X.: ‘Intelligent multi-camera video surveillance: a review’, *Pattern Recognit. Lett.*, 2013, **34**, (1), pp. 3–19

[2] Zabulis, X., Grammenos, D., Sarmis, T., *et al.*: ‘Multicamera human detection and tracking supporting natural interaction with large-scale displays’, *Mach. Vis. Appl.*, 2013, **24**, (2), pp. 319–336

[3] Cardinaux, F., Bhowmik, D., Abhayaratne, C., *et al.*: ‘Video based technology for ambient assisted living: a review of the literature’, *J. Ambient Intell. Smart Environ.*, 2011, **3**, (3), pp. 253–269

[4] Chaaraoui, A.A., Climent-Prez, P., Flrez-Revuelta, F.: ‘A review on vision techniques applied to human behaviour analysis for ambient-assisted living’, *Expert Syst. Appl.*, 2012, **39**, (12), pp. 10873–10888

[5] Geronimo, D., Lopez, A., Sappa, A., *et al.*: ‘Survey of pedestrian detection for advanced driver assistance systems’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010, **32**, (7), pp. 1239–1258

[6] Lu, W.-L., Ting, J.-A., Little, J., *et al.*: ‘Learning to track and identify players from broadcast sports videos’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013, **35**, (7), pp. 1704–1716

[7] Microsoft Corporporation. Kinect for Xbox 360, 2009

[8] Asustek Computer Inc. Xtion PRO LIVE, 2009

[9] Han, J., Shao, L., Xu, D., *et al.*: ‘Enhanced computer vision with Microsoft Kinect sensor: a review’, *IEEE Trans. Cybern.*, 2013, **43**, (5), pp. 1318–1334

[10] Dollar, P., Wojek, C., Schiele, B., *et al.*: ‘Pedestrian detection: an evaluation of the state of the art’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, **34**, (4), pp. 743–761

[11] Luo, W., Zhao, X., Kim, T.: ‘Multiple object tracking: a review’, CoRR abs/1409.7618, 2014, Pre-Print Version. URL http://arxiv.org/abs/1409.7618

[12] Chen, L., Wei, H., Ferryman, J.: ‘A survey of human motion analysis using depth imagery’, *Pattern Recognit. Lett.*, 2013, **34**, (15), pp. 1995–2006

[13] Zhang, J., Li, W., Ogunbona, P.O., *et al.*: ‘RGB-D-based action recognition datasets: a survey’, *Pattern Recogn.*, 2016, **60**, pp. 86–105

[14] Suarez, J., Murphy, R.: ‘Hand gesture recognition with depth images: a review’. RO-MAN, 2012, 2012, pp. 411–417

[15] Endres, F., Hess, J., Sturm, J., *et al.*: ‘3-D mapping with an RGB-D camera’, *IEEE Trans. Robot.*, 2014, **30**, (1), pp. 177–187

[16] Enzweiler, M., Gavrila, D.: ‘Monocular pedestrian detection: survey and experiments’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009, **31**, (12), pp. 2179–2195

[17] Li, T., Chang, H., Wang, M., *et al.*: ‘Crowded scene analysis: a survey’, *IEEE Trans. Circuits Syst. Video Technol.*, 2015, **25**, (3), pp. 367–386

[18] Paul, M., Haque, S.M.E., Chakraborty, S.: ‘Human detection in surveillance videos and its applications – a review’, *EURASIP J. Adv. Signal Process.*, 2013, **2013**, (1), pp. 176

[19] Zhou, H., Hu, H.: ‘Human motion tracking for rehabilitation: a survey’, *Biomed. Signal Proc. Control*, 2008, **3**, (1), pp. 1–18

[20] Garća, G.M., Klein, D.A., Stückler, J., *et al.*: ‘Adaptive multi-cue 3D tracking of arbitrary objects’, *Pattern Recognit.*, 2012, **7476**, pp. 357–366

[21] Song, S., Xiao, J.: ‘Tracking revisited using RGBD camera: unified benchmark and baselines’. IEEE Conf. on Computer Vision, 2013, pp. 233–240

[22] Wang, Q., Fang, J., Yuan, Y.: ‘Multi-cue based tracking’, *Neurocomputing*, 2014, **131**, pp. 227–236

[23] Zhong, B., Shen, Y., Chen, Y., *et al.*: ‘Online learning 3D context for robust visual tracking’, *Neurocomputing*, 2015, **151**, Part 2, pp. 710–718

[24] Walk, S., Schindler, K., Schiele, B.: ‘Disparity statistics for pedestrian detection: combining appearance, motion and stereo’. European Conf. on Computer Vision, 2010, pp. 182–195

[25] Spinello, L., Arras, K. O.: ‘People detection in RGB-D data’. 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, 2011, pp. 3838–3843

[26] Wang, C., Liu, H., Ma, L.: ‘Depth Motion Detection–A Novel RS-Trigger Temporal Logic based Method’, *IEEE Signal Process. Lett.*, 2014, **21**, (6), pp. 717–721

[27] Xia, L., Chen, C.-C., Aggarwal, J.: ‘Human detection using depth information by Kinect’. Computer Vision and Pattern Recognition Workshops, 2011, pp. 15–22

[28] Stahlschmidt, C., Gavriilidis, A., Velten, J., *et al.*: ‘Applications for a people detection and tracking algorithm using a time-of-flight camera’, *Multimedia Tools Appl.*, 2016, **75**, (17), pp. 10769–10786

[29] Bagautdinov, T., Fleuret, F., Fua, P.: ‘Probability occupancy maps for occluded depth images’. IEEE Conf. on Computer Vision and Pattern Recognition, 2015, pp. 2829–2837

[30] Fosty, B., Crispim-Junior, C.F., Badie, J., *et al.*: ‘Event recognition system for older people monitoring using an RGB-D camera’. Workshop on Assistance and Service Robotics in a Human Environment, 2013

[31] Dondi, P., Lombardi, L., Cinque, L.: ‘Multisubjects tracking by time-of-flight camera’. Conf. on Image Analysis and Processing, 2013, vol. **8156**, pp. 692–701

[32] Yun, K., Honorio, J., Chattopadhyay, D., *et al.*: ‘Two-person interaction detection using body-pose features and multiple instance learning’. IEEE Conf. on Computer Vision and Pattern Recognition Workshops, 2012, pp. 28–35

[33] Xu, N., Liu, A., Nie, W., *et al.*: ‘Multi-modal & multi-view & interactive benchmark dataset for human action recognition’. ACM Conf. on Multimedia, 2015, pp. 1195–1198

[34] Shahroudy, A., Liu, J., Ng, T.-T., *et al.*: ‘NTU RGB+D: a large scale dataset for 3D human activity analysis’, arXiv preprint arXiv:1604.02808

[35] Grenader, E., Gasques Rodrigues, D., Nos, F., *et al.*: ‘The VideoMob interactive art installation connecting strangers through inclusive digital crowds’, *ACM Trans. Inter. Intell. Syst.*, 2015, **5**, (2), pp. 7:1–7:31

[36] Stone, E.E., Skubic, M.: ‘Fall detection in homes of older adults using the Microsoft Kinect’, *IEEE J. Biomed. Health Inf.*, 2015, **19**, (1), pp. 290–301

[37] Zhu, N., Diethe, T., Camplani, M., *et al.*: ‘Bridging e-health and the internet of things: the SPHERE project’, *IEEE Intell. Syst.*, 2015, **30**, (4), pp. 39–46

[38] Dalal, N., Triggs, B.: ‘Histograms of oriented gradients for human detection’. IEEE Computer Vision and Pattern Recognition Conf., 2005, pp. 886–893

[39] Bourdev, L., Malik, J.: ‘Poselets: body part detectors trained using 3D human pose annotations’. IEEE Int. Conf. on Computer Vision, 2009, pp. 1365–1372

[40] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., *et al.*: ‘Object detection with discriminatively trained part based models’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010, **32**, (9), pp. 1627–1645

[41] Viola, P., Jones, M.: ‘Robust real-time face detection’, *Int. J. Comput. Vis.*, 2004, **57**, (2), pp. 137–154

[42] Bansal, M., Jung, S.-H., Matei, B., *et al.*: ‘A real-time pedestrian detection system based on structure and appearance classification’. IEEE Int. Conf. on Robotics and Automation, 2010, pp. 903–909

[43] Salas, J., Tomasi, C.: ‘People detection using color and depth images’. Mexican Conf. on Pattern Recognition, 2011, pp. 127–135

[44] Dan, B.-K., Kim, Y.-S., Suryanto, J.-Y., *et al.*: ‘Robust people counting system based on sensor fusion’, *IEEE Trans. Consum. Electron.*, 2012, **58**, (3), pp. 1013–1021

[45] Darrell, T., Gordon, G., Harville, M., *et al.*: ‘Integrated person tracking using stereo, color, and pattern detection’, *Int. J. Comput. Vis.*, 2000, **37**, (2), pp. 175–185

[46] Han, J., Pauwels, E.J., de Zeeuw, P.M., *et al.*: ‘Employing a RGB-D sensor for real-time tracking of humans across multiple re-entries in a smart environment’, *IEEE Trans. Consum. Electron.*, 2012, **58**, (2), pp. 255–263

[47] Bajracharya, M., Moghaddam, B., Howard, A., *et al.*: ‘A fast stereo-based system for detecting and tracking pedestrians from a moving vehicle’, *The Int. J. Robot. Res.*, 2009, **28**, (11-12), pp. 1466–1485

[48] Zhang, H., Reardon, C., Parker, L.: ‘Real-time multiple human perception with color-depth cameras on a mobile robot’, *IEEE Trans. Cybern.*, 2013, **43**, (5), pp. 1429–1441

[49] Galamakis, G., Zabulis, X., Koutlemanis, P., *et al.*: ‘Tracking persons using a network of RGBD cameras’. Int. Conf. on Pervasive Technologies for Assistive Environments, 2014, pp. 63:1–63:4

[50] Liu, J., Liu, Y., Cui, Y., *et al.*: ‘Real-time human detection and tracking in complex environments using single RGB-D camera’. IEEE Int. Conf. on Image Processing, 2013, pp. 3088–3092

[51] Liu, J., Liu, Y., Zhang, G., *et al.*: ‘Detecting and tracking people in real time with RGB-D camera’, *Pattern Recognit. Lett.*, 2015, **53**, pp. 16–23

[52] Liu, J., Zhang, G., Liu, Y., *et al.*: ‘An ultra-fast human detection method for color-depth camera’, *J. Vis. Commun. Image Represent.*, 2015, **31**, pp. 177–185

[53] Luber, M., Spinello, L., Arras, K.O.: ‘People tracking in RGB-D data with on-line boosted target models’. Int. Conf. on Intelligent Robots and Systems, 2011, pp. 3844–3849

[54] Linder, T., Arras, K.O.: ‘Multi-model hypothesis tracking of groups of people in RGB-D data’. IEEE Conf. on Information Fusion, Salamanca, Spain, 2014, pp. 1–7

[55] Ess, A., Leibe, B., Schindler, K., *et al.*: ‘Robust multiperson tracking from a mobile platform’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009, **31**, (10), pp. 1831–1846

[56] Jafari, O., Mitzel, D., Leibe, B.: ‘Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras’. IEEE Conf. on Robotics and Automation, 2014, pp. 5636–5643

[57] Muñoz Salinas, R., Aguirre, E., Garća-Silvente, M.: ‘People detection and tracking using stereo vision and color’, *Image Vis. Comput.*, 2007, **25**, (6), pp. 995–1007

[58] Munaro, M., Basso, F., Menegatti, E.: ‘Tracking people within groups with RGB-D data’. IEEE/RSJ Conf. on Intelligent Robots and Systems, 2012, pp. 2101–2107

[59] Munaro, M., Lewis, C., Chambers, D., *et al.*: 'RGB-D human detection and tracking for industrial environments'. Int. Conf. on Intelligent Autonomous Systems, 2014, pp. 1655–1668

[60] Almazán, E., Jones, G.: 'A depth-based polar coordinate system for people segmentation and tracking with multiple RGB-D sensors'. IEEE ISMAR Workshop on Tracking Methods and Applications, 2014

[61] Almazán, E., Jones, G.: 'Tracking people across multiple non-overlapping RGB-D sensors'. IEEE Conf. on Computer Vision and Pattern Recognition Workshops, 2013, pp. 831–837

[62] Bahadori, S., Iocchi, L., Leone, G., *et al.*: 'Real-time people localization and tracking through fixed stereo vision'. Innovations in Applied Artificial Intelligence, 2005, pp. 44–54

[63] Beymer, D., Konolige, K.: 'Real-time tracking of multiple people using stereo'. IEEE Conf. on Computer Vision Workshops, 1999, pp. 1076–1083

[64] Satake, J., Chiba, M., Miura, J.: 'Visual person identification using a distance-dependent appearance model for a person following robot', *Int. J. Autom. Comput.*, 2013, **10**, (5), pp. 438–446

[65] Vo, D.M., Jiang, L., Zell, A.: 'Real time person detection and tracking by mobile robots using RGB-D images'. IEEE Conf. on Robotics and Biomimetics, 2014, pp. 689–694

[66] Vo, D.M., Masselli, A., Zell, A.: 'Real time face detection using geometric constraints, navigation and depth-based skin segmentation on mobile robots'. IEEE Symp. on Robotic and Sensors Environments, 2012, pp. 180–185

[67] Harville, M.: 'Stereo person tracking with adaptive plan-view templates of height and occupancy statistics', *Image Vis. Comput.*, 2004, **22**, (2), pp. 127–142

[68] Muñoz Salinas, R.: 'A Bayesian plan-view map based approach for multiple-person detection and tracking', *Pattern Recogn.*, 2008, **41**, (12), pp. 3665–3676

[69] Muñoz Salinas, R., Medina-Carnicer, R., Madrid-Cuevas, F.: 'A. Carmona-Poyato, People detection and tracking with multiple stereo cameras using particle filters', *J. Vis. Commun. Image Represent.*, 2009, **20**, (5), pp. 339–350

[70] Muñoz Salinas, R., Garća-Silvente, M., Carnicer, R.M.: 'Adaptive multi-modal stereo people tracking without background modelling', *J. Vis. Commun. Image Represent.*, 2008, **19**, (2), pp. 75–91

[71] Choi, W., Pantofaru, C., Savarese, S.: 'Detecting and tracking people using an RGB-D camera via multiple detector fusion'. IEEE Conf. on Computer Vision Workshops, 2011, pp. 1076–1083

[72] Choi, W., Pantofaru, C., Savarese, S.: 'A general framework for tracking multiple people from a moving camera', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013, **35**, (7), pp. 1577–1591

[73] Migniot, C., Ababsa, F.: 'Hybrid 3D–2D human tracking in a top view', *J. Real-Time Image Process.*, 2016, **11**, (4), pp. 769–784

[74] Gao, S., Han, Z., Li, C., *et al.*: 'Real-time multipedestrian tracking in traffic scenes via an RGB-D-based layered graph model', *IEEE Trans. Intell. Transp. Syst.*, 2015, **16**, (5), pp. 2814–2825

[75] Ma, A.J., Yuen, P.C., Saria, S.: 'Deformable distributed multiple detector fusion for multi-person tracking', arXiv preprint arXiv:1512.05990, 2015

[76] Rubner, Y., Tomasi, C., Guibas, L.: 'The earth mover's distance as a metric for image retrieval', *J. Int. Comput. Vis.*, 2000, **40**, (2), pp. 99–121

[77] Argyros, A.A., Lourakis, M.I.: 'Real-time tracking of multiple skin-colored objects with a possibly moving camera'. European Conf. on Computer Vision, 2004, pp. 368–379

[78] Padeleris, P., Zabulis, X., Argyros, A.: 'Multicamera tracking of multiple humans based on colored visual hulls'. IEEE Conf. on Emerging Technologies Factory Automation, 2013, pp. 1–8

[79] Cox, I.J., Hingorani, S.L.: 'An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1996, **18**, (2), pp. 138–150

[80] Eveland, C., Konolige, K., Bolles, R.: 'Background modeling for segmentation of video-rate stereo sequences'. IEEE Computer Vision and Pattern Recognition, 1998, pp. 266–271

[81] Leibe, B., Schindler, K., Cornelis, N., *et al.*: 'Coupled object detection and tracking from static cameras and moving vehicles', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2008, **30**, (10), pp. 1683–1698

[82] Sudowe, P., Leibe, B.: 'Efficient use of geometric constraints for sliding-window object detection in video'. Computer Vision Systems, 2011, pp. 11–20

[83] Satake, J., Miura, J.: 'Robust stereo-based person detection and tracking for a person following robot'. ICRA Workshop on People Detection and Tracking, 2009, pp. 1–10

[84] Lowe, D.: 'Object recognition from local scale-invariant features'. IEEE Conf. on Computer Vision, 1999, pp. 1150–1157

[85] Kuhn, H.: 'The Hungarian method for the assignment problem', *Naval Res. Logist. Q.*, 1955, **2**, pp. 83–97

[86] Harville, M., Gordon, G., Woodfill, J.: 'Foreground segmentation using adaptive mixture models in color and depth'. IEEE Workshop on Detection and Recognition of Events in Video, 2001, pp. 3–11

[87] Milan, A., Schindler, K., Roth, S.: 'Multi-target tracking by discrete-continuous energy minimization', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016, **38**, (10), pp. 2054–2068

[88] Comaniciu, D., Meer, P.: 'Mean shift: a robust approach toward feature space analysis', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002, **24**, (5), pp. 603–619

[89] Chambers, D.R., Flannigan, C., Wheeler, B.: 'High-accuracy real-time pedestrian detection system using 2D and 3D features'. SPIE Defense, Security, and Sensing, 2012, vol. **8384**, pp. 83840G–83840G–11

[90] Fischler, M.A., Bolles, R.C.: 'Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography', *Commun. ACM*, 1981, **24**, (6), pp. 381–395

[91] Vedaldi, A., Soatto, S.: 'Quick Shift and kernel methods for mode seeking'. European Conf. on Computer Vision, 2008, pp. 705–718

[92] Camplani, M., Salgado, L.: 'Background foreground segmentation with RGB-D Kinect data: an efficient combination of classifiers', *J. Vis. Commun. Image Represent.*, 2014, **25**, (1), pp. 122–136

[93] Camplani, M., del Blanco, C.R., Salgado, L., *et al.*: 'Advanced background modeling with RGB-D sensors through classifiers combination and inter-frame foreground prediction', *Mach. Vis. Appl.*, 2014, **25**, (5), pp. 1197–1210

[94] Gordon, G., Darrell, T., Woodfill, J.: 'Background estimation and removal based on range and color'. IEEE Conf. on Computer Vision and Pattern Recognition, 1999

[95] Kammerl, J.: 'Octree Point Cloud Compression in PCL', 2011

[96] Ganotra, D., Joseph, J., Singh, K.: 'Modified geometry of ring-wedge detector for sampling Fourier transform of fingerprints for classification using neural networks', *Opt. Lasers Eng.*, 2004, **42**, (2), pp. 167–177

[97] Quigley, M., Conley, K., Gerkey, B.P., *et al.*: 'ROS: an open-source robot operating system'. ICRA Workshop on Open Source Software, 2009

[98] Munaro, M., Horn, A., Illum, R., *et al.*: 'OpenPTrack: people tracking for heterogeneous networks of color-depth cameras'. IAS Workshop on 3D Robot Perception with Point Cloud Library, 2014, pp. 235–247

[99] Munaro, M., Basso, F., Menegatti, E.: 'OpenPTrack: open source multi-camera calibration and people tracking for RGB-D camera networks', *Robot. Auton. Syst.*, 2016, **75**, Part B, pp. 525–538

[100] Ess, A., Leibe, B., Schindler, K., *et al.*: 'A mobile vision system for robust multi-person tracking'. IEEE Conf. on Computer Vision and Pattern Recognition, 2008, pp. 1–8

[101] Munaro, M., Menegatti, E.: 'Fast RGB-D people tracking for service robots', *Auton. Robots*, 2014, **37**, (3), pp. 227–242

[102] Felzenszwalb, P.F., Huttenlocher, D.P.: 'Efficient belief propagation for early vision', *Int. J. Comput. Vis.*, 2006, **70**, (1), pp. 41–54

[103] Leal-Taixé, L., Milan, A., Reid, I., *et al.*: 'MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking', arXiv:1504.01942 [cs]ArXiv: 1504.01942

[104] Rusu, R.B., Cousins, S.: '3D is here: Point Cloud Library (PCL)'. IEEE Int. Conf. on Robotics and Automation, 2011, pp. 1–4

[105] Mitzel, D., Leibe, B.: 'Close-range human detection and tracking for head-mounted cameras'. British Machine Vision Conf., 2012, pp. 8.1–8.11

[106] Bernardin, K., Stiefelhagen, R.: 'Evaluating multiple object tracking performance: the CLEAR MOT metrics', *J. Image Video Process.*, 2008, **2008**, pp. 1:1–1:10

[107] Szczodrak, M., Dalka, P., Czyzewski, A.: 'Performance evaluation of video object tracking algorithm in autonomous surveillance system'. Int. Conf. on Information Technology, 2010, pp. 31–34