

People tracking with range cameras using density maps and 2D blob splitting

Izaak Van Crombrugge^a, Rudi Penne^{a,b} and Steve Vanlanduit^{a,*}

^a*Faculty of Applied Engineering, University of Antwerp, 2020 Antwerp, Belgium*

^b*Department of Mathematics, University of Antwerp, 2020 Antwerp, Belgium*

Abstract. A robust method is proposed for people tracking using range cameras: Density Map Tracking with Blob Splitting (DMT-BS). It was designed primarily for worker safety in industrial environments, but can be used in other applications as well. Multiple cameras can easily be added to resolve occlusions and to enlarge the observed area. The method could be used to track any moving object in an otherwise static environment as the detection does not rely on a specific human model. Its strength lies in its simplicity, making the behavior predictable and opening possibilities to be implemented on low-cost hardware. From the point cloud delivered by the depth sensor, a 2D density map is formed in floor coordinates followed by basic 2D-tracking. Robustness of this tracking is enhanced using a simple but effective blob splitting technique. Tests show that the camera position, depth noise, and extrinsic calibration errors have little influence on the tracker's performance. The proposed method was tested on three depth tracking datasets, reaching significantly better MOTA (Multiple Object Tracking Accuracy) scores when compared to two state-of-the-art depth-based trackers.

Keywords: Human tracking, depth camera, time-of-flight, RGB-D, industrial safety

1. Introduction

People tracking can play an important role in worker safety in industry. It can for example be used to slow down or to stop dangerous machines when human operators come too close, or to prevent the lowering of a heavy load when a person is still under it.

Safety should not be confused with security. Where the latter aims at detecting people with bad intentions, we focus on the first: protecting people from physical harm. This results in specific requirements for safety tracking. For example, a high recall is important so that no person is lost, but re-identification is not required as each person has equal value.

A tracking algorithm aimed at industrial safety applications has specific requirements:

- It must cope with changing lighting conditions.

- To avoid false alarms, it needs a robust background subtraction method.
- Temporary occlusions must not break the track.
- The needed processing power should be kept low to allow large scale implementation with many cameras.

Using range cameras helps in meeting these requirements. Range cameras are relatively insensitive to lighting effects such as shadows and illumination changes. The added depth data is a reliable base for foreground segmentation and provides an extra dimension to resolve occlusions. Using only depth data has an additional advantage that is often underestimated: it does not reveal the identity of the subjects. This is important to make the presence of 'cameras' in the workplace acceptable for workers and unions.

To make risk assessment possible, the algorithm should be as simple as possible so that its inner workings and its behavior can be understood by the technicians that will implement the safety system. It also makes implementation on low-cost hardware possible.

*Corresponding author: Steve Vanlanduit, Faculty of Applied Engineering, University of Antwerp, Groenenborgerlaan 171, 2020 Antwerp, Belgium. E-mail: steve.vanlanduit@uantwerpen.be.

In this paper, we set out a robust tracking method with minimal computational complexity.

2. Related work

Most camera tracking is done using RGB-cameras, using background subtraction, followed by a people detector [1,2]. These detectors – for example HOG [3, 4] or an SVM classifier [5] – require considerable computing power and struggle to deal with occlusion.

Range cameras – such as time-of-flight cameras or triangulation techniques [6,7] – offer extra spatial information, which can be of great help in dealing with occlusions. Some methods use one or more range cameras looking from a top-down perspective [8,9,10]. While this perspective is the easiest way to distinguish individual people, it limits the observed area and can be hard to realize in places with a low ceiling or no ceiling at all. For applications in industry, lower camera positions may be desired to allow for flexible camera placement.

Munaro and Menegatti [11,12] use a depth-based sub-clustering method. In their classification they also use color information and the method is computationally intensive, restricting the use in low-power systems. DPOM: Probability Occupancy Maps for Occluded Depth Images [13] is an algorithm with a very similar application domain, but is too slow to run in real-time.

Harville [14] as well as Muscoloni and Mattocchia [15] used height maps projected on the ground plane to get a virtual birds-eye view. Their approach is similar to the one we propose. While these methods are fast, raw height maps can be very noisy and filtering them requires additional processing power.

OpenPTrack [12] can combine multiple different cameras using various detectors for monochrome, RGB, and depth images in a tracking network. This allows for more flexibility but at the cost of increased complexity.

The purpose of this research is to develop a tracking algorithm that can robustly track operators in a working environment, while keeping complexity to a minimum. But this simplicity is missing in the current state-of-the-art people trackers.

3. Tracking algorithm

The proposed method assumes the use of range sensors that provide 3D point clouds, either directly, or as

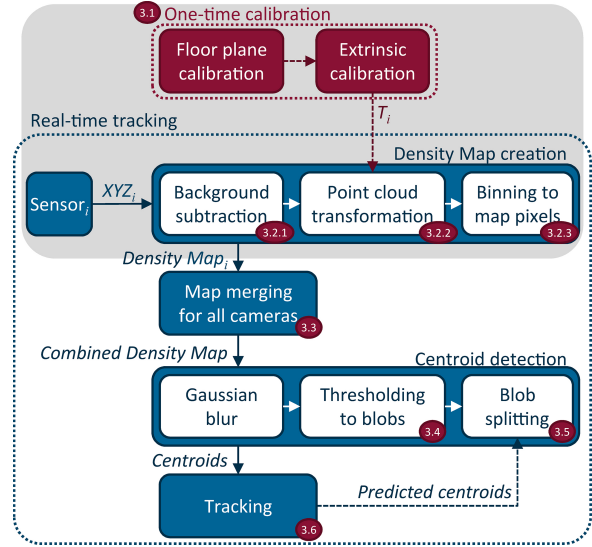


Fig. 1. Block diagram of the proposed tracking method.

the result of a computation from a depth map and an available intrinsic calibration.

As depicted in Fig. 1, the algorithm consists of two main parts: a one-time calibration and the real-time position tracking, divided into several sub steps. The numbers in the figure refer to the relevant section. Multiple cameras can be used to resolve occlusions. The steps in the grey area are done for each camera i separately, the rest is done globally.

3.1. One-time calibration

Initially, the floor plane and extrinsic calibration need to be established. The floor plane is detected in the first camera depth image or alternatively the floor plane model can be updated in real time as described in [16]. The extrinsic calibration when using multiple range cameras is a well-studied subject where many good solutions exist [17,18]. The result is a rigid transformation for each camera that transforms points from individual camera coordinates to a common world coordinate system.

3.2. Density map creation

The core of the proposed method is the use of a density map, similar to an occupancy map. Point clouds from the different cameras are combined into a single density map. The steps to create the density map from a depth frame are visualized in Fig. 2a–e. The resulting density map is a 2D map that shows the density of the measured points, in a top-down view.

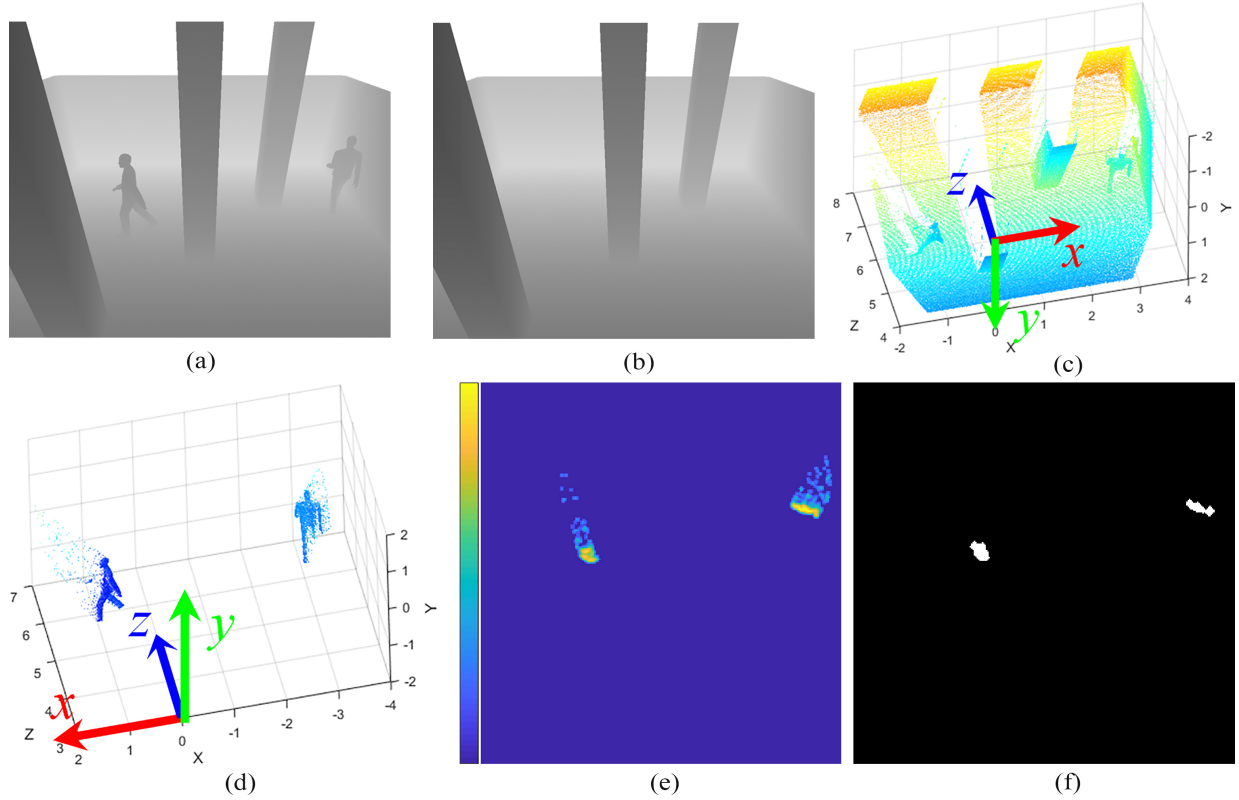


Fig. 2. Visualization of the different steps. (a) Z-values of the current frame. (b) The background model. (c) Full point cloud of the current frame in camera coordinates. (d) Foreground points transformed to world coordinates. (e) Density map after gaussian blur. (f) Segmented people in blob image.

3.2.1. Background subtraction

The foreground segmentation is done on the Z -image alone. Each point for which $Z < (BG - thr_{depth})$ is assumed foreground. Range cameras often produce “invalid” pixels, caused by over-, or underexposure. The pixels of the background model corresponding to the invalid pixels in the current Z -frame cannot be updated based on these invalid pixel values. Therefore, these pixels are “pushed back” into the background, see Eqs (3) and (4). Initially the first Z -frame is used as background. The background is updated using a simple but effective scheme. For all frames, only for the valid Z -pixels:

$$BG = (1 - a) \cdot BG + a \cdot Z \quad (1)$$

$$BG = (1 - b) \cdot BG + b \cdot \max(BG, Z) \quad (2)$$

During the initialization, for the invalid Z -pixels:

$$BG = (1 - b) \cdot BG + b \cdot z_{\max} \quad (3)$$

After the initialization, for the invalid Z -pixels:

$$BG = (1 - a) \cdot BG + a \cdot z_{\max} \quad (4)$$

With BG the background “image”, Z the current Z -frame, a and b the normal update rate and the *fast* update rate respectively ($a \ll b$), and $z_{\max}$ a chosen maximum background value. The values used in the experiments are summarized in Table 1.

Updating the background for valid pixels comprises two steps. First a weighted average of the background and the current frame is made in Eq. (1) with a small weight a for the current frame. Then the background is mixed with the maximum of the background and the current frame in Eq. (2). This accelerates the background update for pixels that are further away in the current frame than in the background.

3.2.2. Point cloud transformation

The point cloud is transformed from camera coordinates (Fig. 2c) to a common world coordinate system (Fig. 2d) using the predetermined rigid transformation T . This point cloud has the floor in the xz -plane.

3.2.3. Binning to map pixels

The ideal vantage point for people tracking is looking straight down from a height. This way, people can

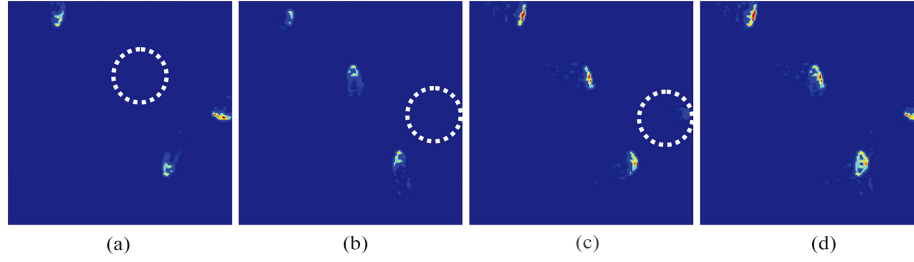


Fig. 3. Maps of different cameras (a)–(c) and the combined map (d). Because of occlusions, in each camera 1 person is not detected.

Table 1
Used parameters and their values

Param	Value	Unit	Description
a	0.001		Normal update rate
b	0.1		Fast update rate
thr_{depth}	0.2	m	Threshold for background subtraction
z_{max}	10	m	Background distance
t_{init}	60	frames	Initialization time
S	40	px/m	Density Map scale
K	1.4		Empirical constant to normalize to 1
σ	$0.02 \cdot S$	px	Standard deviation of Gaussian blur
$thr_{density}$	0.5		Density threshold
$R_{closing}$	$0.1 \cdot S$	px	Radius of morph. closing (10 cm)
$R_{opening}$	$0.05 \cdot S$	px	Radius of morph. opening (5 cm)
$d_{max\ match}$	$1.0 \cdot S$	px	Max. dist. to match detection to track
$d_{max\ split}$	$0.3 \cdot S$	px	Max. dist. from prediction to blob
t_{split}	20	frames	Minimum age of track to split blobs

easily be distinguished. Such an aerial position is often hard to realize. However, using the 3D point cloud this view can be reconstructed. This is the major advantage of using range data.

From the transformed point cloud (Fig. 2d), a density map is created (Fig. 2e). The size of this density map depends on the region of interest and the map scale S , in pixels per meter. The point cloud is projected on the floor plane (xz -plane) by binning the points into map pixels, based on their x and z values.

In the experiments, only the points whose distance to the floor is greater than 1 meter are kept. Standing and walking people's legs are removed from the point cloud. Body parts lower than 1 meter are often occluded and do not offer any additional information on top of the rest of the body. This height threshold can simply be altered or removed altogether, for example to include crouching people.

The further objects are from the camera, the less points they represent in the image. This is an inverse-square ratio. To compensate for this, each point (above height threshold) contributes to the map as follows:

$$map(row_i, col_i) = map(row_i, col_i) + d_i^2 \quad (5)$$

with d_i the z -distance between the camera and point P_i . This way, similar objects at different distances still have a similar density in the density map.

After the binning, the density map is normalized to be independent of the camera's focal length f and map scale S . The smaller the scale, the larger the bins, the more points will accumulate per bin resulting in a higher density. With larger focal length objects are projected onto a larger image surface. Therefore, density $\propto 1/S^2$ and density $\propto f^2$. To compensate for this:

$$map_{norm} = map \cdot K \cdot S^2 / f^2 \quad (6)$$

The empirical constant K is added, so that the normalized map can be thresholded by 1.0 later on.

3.3. Map merging

In the point cloud transformation step, the point cloud of each camera is transformed to a common world coordinate system. The merged map is simply the pixel-wise maximum of the individual density maps. An example is shown in Fig. 3.

3.4. Centroid detection

For each object, the centroid of its projection on the map is tracked. The steps to create the blobs are visualized in Fig. 2e and f. First the density map is blurred using a gaussian blur with a standard deviation of $2 \text{ cm} \cdot S$. Then the smoothed density map is thresholded: all bins with a value greater than 1.0 are deemed to belong to objects.

The resulting black and white image is then filtered using a morphological closing (disk with radius $10 \text{ cm} \cdot S$) followed by a morphological opening (disk with radius $5 \text{ cm} \cdot S$). The disk sizes were determined empirically but they are universal because of the scale factor and the map normalization. The 2D centroids of the resulting blobs are used as the actual detections.

3.5. Blob splitting

The key to make this simple method work reliably is the effective blob splitting scheme, illustrated in Fig. 4.

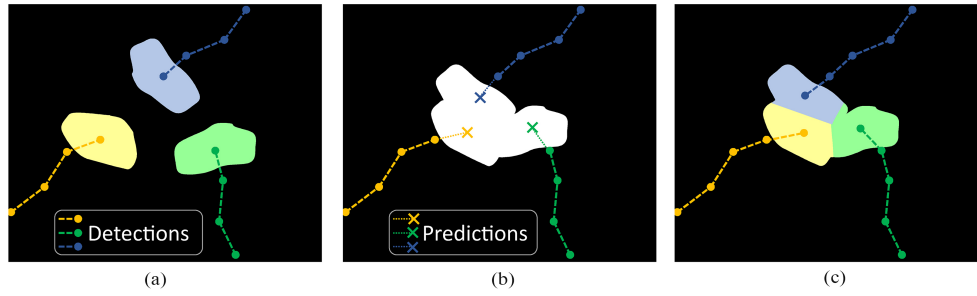


Fig. 4. Example of blob splitting. a. Three tracked objects approach each other. b. Blobs overlap and the predicted centroids share a merged detection. c. The merged detection is split into three separate detections based on proximity to the predicted centroids.

When two or more blobs merge, the information contained in the merged blob is still used to estimate the centroids of the individual objects.

First the merged blobs are detected. When detections and tracks are matched using the Hungarian algorithm [19], at least one of the tracks whose blobs are merged will not be matched to a detection. For those unmatched tracks, it is tested if their predicted location falls close to a blob. Given that no detection was assigned to the track even though the prediction falls close to a blob, it is assumed that this track's blob has merged with another track's blob.

In the next step, all merged blobs are split using Euclidean Voronoi partitioning. A blob is split by assigning each of its pixels to the track whose prediction is nearest to that pixel, based on the Euclidian distance.

Only blobs that have been in view for enough frames, will be split. In the experiments this minimum track age was 1 s. Objects that have merged blobs when entering the view, will be tracked as a single object. As soon as their blobs detach, they will be tracked as individual objects. From then on, the tracker will split them if they would merge again.

3.6. Tracking

All tracking is done in 2D using world coordinates in the floor plane. The centroid positions are predicted using a Kalman filter assuming constant velocity. These predictions are matched with the detections using the Hungarian algorithm with the Euclidian distance between the detection and the prediction as cost.

3.7. Note on segmentation of humans and objects

The proposed method does not have an explicit classification step to discriminate between humans and objects. However, following properties of the method do filter out most objects:

- Stationary objects will over time disappear into the background as a result of the background subtraction method.
- All objects that are below the height threshold (1 meter) are excluded from the tracking.
- Objects that are significantly smaller than humans, result in too low a density in the density map to be included in the tracking.

As a result, only large (human size) moving objects are tracked.

4. Experimental evaluation

The experiments can be divided into two categories:

- Comparison to state-of-the-art depth-based trackers (see Section 4.2) on real-world datasets (see Section 4.1).
- A sensitivity analysis using simulated data (see Section 4.3).

4.1. Real-world datasets

While many RGB datasets are available for people tracking, RGB-D datasets that also report the 3D ground truth are less common. Two such datasets were used. The Kinect Tracking Precision dataset (KTP) [11] provides a number of sequences with up to 5 people in a lab setting. Only the sequences where the camera is static were used in this evaluation. The scenarios are:

- 1 person walking back and forth (390 frames)
- 3 people walking random paths (850 frames)
- 2 people side-by-side straight path (160 frames)
- 1 person running across the room (150 frames)
- 5 people grouping together (487 frames)

The ground truth had to be cleaned up manually because some detections were missing. The recording was made at 30 fps.

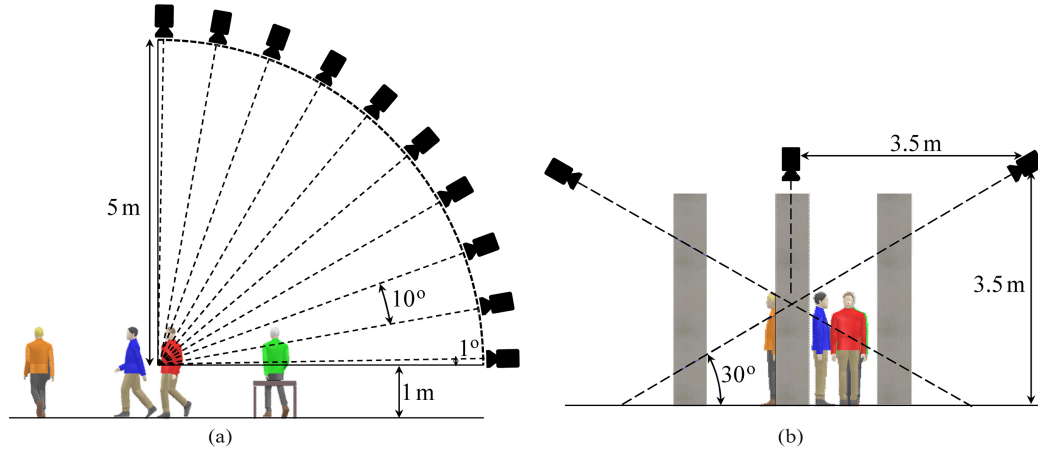


Fig. 5. Design of the scenes. (a) Scene 1: Different camera heights and angles. (b) Scene 2: Placement of multiple cameras for multi-camera tracking.

In [13] a RGB-D dataset is presented that was recorded using the newer Kinect camera. It was captured in two different environments:

- One in a lab (EPFL-LAB) of around 1000 frames with at most 4 people.
- A more challenging one of around 3000 frames in a corridor (EPFL-CORRIDOR) with up to 8 people.

The ground truth of this dataset was first corrected: some objects were missing when they were occluded. The framerate varies between 20 and 30 fps.

4.2. Baseline methods

The proposed method (DMT-BS) is compared to following depth-based tracking methods to validate it against comparable state-of-the-art.

- DPOM: Probability Occupancy Maps for Occluded Depth Images [13] is an algorithm with a very similar application domain to that of the proposed method. It optimizes the probabilities of presence using a generative model that predicts the distribution of the depth images.
- KINECT2: The result from the human pose estimation of the Kinect for Windows SDK. What method is used exactly is unknown. Despite the closed-source nature of this software, it is still a valuable comparison baseline because the Kinect depth sensor is often used in research.

Precautions were taken to guarantee a fair comparison. Both compared methods only use depth information. The results for DPOM and KINECT2 were obtained by the authors of the DPOM paper, taking the limitations of the KINECT2 algorithm into account.

4.3. Simulation setup

Sensitivity analyses are done in following sections:

- Camera height and angle (4.3.1, scene 1)
- Number of combined cameras (4.3.2, scene 2)
- Depth noise (4.3.3, scene 2)
- Errors in extrinsic calibration (4.3.4, scene 2)

These sensitivity analyses are done using simulated data to make sure that only the investigated parameter is varied and all else stays exactly the same. Two scenes are used, as shown in Figs 5 and 6. Scene 1 is only used to evaluate the camera angle sensitivity. Scene 2 is used in the other simulation experiments.

The proposed tracking method was tested on simulation data generated in V-REP [20]. The simulated range sensor was modelled to have the same properties as the Microsoft Kinect for Xbox One time-of-flight camera: 512×424 , $f \approx 362$. To generate realistic flying pixels, the images were rendered at double resolution and downsampled using bilinear interpolation. Random noise was added following the same distribution as we measured on three real Kinects [21], confirming the results of [22]. The simulated frame rate was 20 fps.

4.3.1. Sensitivity to camera height and angle

Tracking results are often strongly dependent of the height and angle of the camera. When a camera is placed low and at a shallow angle, tracking will be more challenging as there will be more occlusions. When the camera is placed high and at a downward angle, it has an unoccluded bird's-eye view.

To evaluate the performance of the tracker for different camera angles, the exact same simulation was

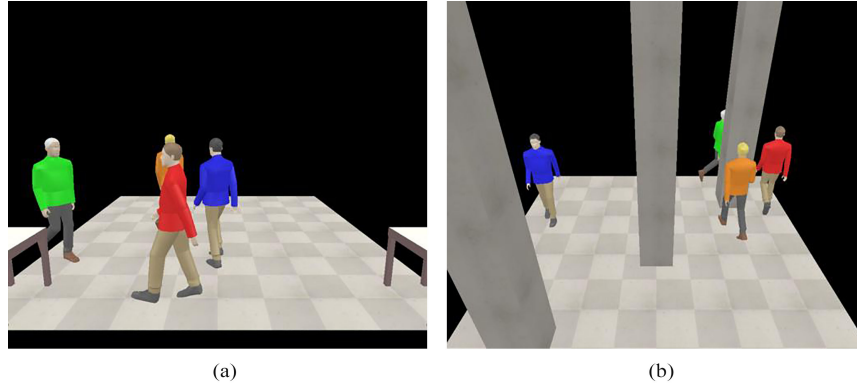


Fig. 6. Example images of the simulated scenes. (a) Scene 1: Different angles scene at a 10° angle. (b) Scene 2: Multiple cameras and multiple occlusions.

performed with the camera at 10 different camera angles ranging from 0° to 90° in steps of 10° as shown in Fig. 5a. Due to practical considerations, the minimum and maximum angles were 1° and 89° respectively, but the effect of this small angle difference is negligible. The scene consists of four people walking on a plane of 5 by 5 meters for a duration of 100 seconds (2000 frames).

4.3.2. Single versus multiple camera performance

Tracking can be problematic in scenes with lots of occlusions. Good placement of extra cameras can be an effective solution. The main purpose of using multiple cameras is to resolve occlusions. While it can be used to extend the observed area, there must always be an overlap because the algorithm does not perform re-identification.

In this experiment, the added value of extra cameras is evaluated. A challenging test scene was made with four people walking on a plane of 5 by 5 meters and three large pillars. Four cameras were placed symmetrically as shown in Fig. 5b. The tracking of the same sequence is performed four times, each time using one extra camera. Seven sequences of 100 seconds each were simulated and evaluated. The results were calculated over all frames of the set.

4.3.3. Sensitivity to depth noise

To test the influence of depth noise on the tracking performance, scene 2 is used with two cameras. The original noise level is the same as that of the Kinect for Xbox One. This noise level depends on the depth of the measured point and on the distance of the image pixel from the image center. At 5 m this noise has a standard deviation of 5 mm for most of the pixels.

The same simulation (7 sequences of 2000 frames)

is run 12 times with Gaussian noise with σ varying from 0 to 200 mm.

4.3.4. Sensitivity to extrinsic calibration errors

In realistic scenarios, it is near impossible to obtain a “perfect” extrinsic calibration between multiple cameras. Scene 2 is used with two cameras. To test the influence of inaccuracies in the extrinsic calibration, an error is introduced in the transformation between the cameras. This is a worst-case error: an incorrect value of the rotation of the second camera around the vertical axis. This error is varied between 0 and 3° in steps of 0.5° , so the simulation (7 sequences of 2000 frames) is run six times.

4.4. Evaluation metrics

To quantify the results, the CLEAR MOT metrics [23] are used in the implementation of the MOTChallenge DevKit [24,25]. TP , FP and FN are the total number of true positives, false positives and false negatives, considering all detections in all frames. Precision and recall are defined as usual:

$$P = \frac{TP}{TP + FP} \text{ and } R = \frac{TP}{TP + FN}. \quad (7)$$

Where Φ is the number of fragmentations and T the total number of ground truth detections, the Multiple Object Tracking Accuracy (MOTA) is defined as

$$MOTA = 1 - \frac{FN + FP + \Phi}{T}. \quad (8)$$

A detection is matched with a ground truth object when the distance between the two is no more than threshold distance t_d . The number of such matches in frame t is c_t and $d_{t,i}$ is the distance between target i and its assigned ground truth object. Multiple Object Tracking Precision is then defined as

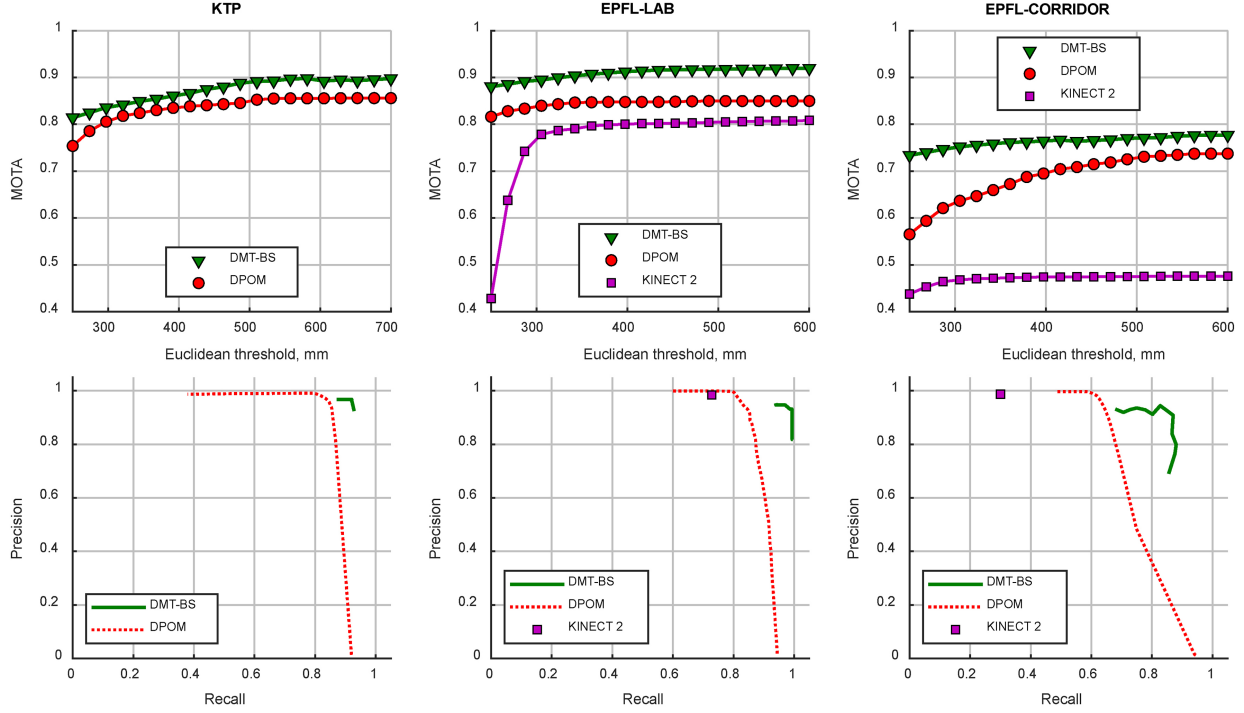


Fig. 7. Tracking performance compared to state-of-the-art trackers on different datasets. The precision-recall curve of the proposed method (DMT-BS) is so small because there is only a limited range of relevant thresholds.

$$MOTP = 1 - \frac{\sum_{t,i} d_{t,i}}{t_d \cdot \sum_t c_t}. \quad (9)$$

In all experiments, t_d is set to 0.5 meter. The MOTA score will be the main evaluation metric: it is a fitting metric to express the performance of a tracker aimed at industrial safety as it penalizes false negatives, false positives, and track fragmentations.

5. Experimental results

5.1. Results of comparison on real-world datasets

Notwithstanding the simple nature of the proposed method, it is still the most effective tracker in the experiments on three datasets. All trackers had a lower performance on the EPFL-CORRIDOR dataset. This is due to the challenging nature of this dataset: the camera is not positioned in a high position (only eye height) and at times many people are present, causing lots of occlusions.

DMT-BS has overall the highest MOTA score. The precision-recall graphs in Fig. 7 also show that it has a high recall while maintaining a good precision, important for safety applications.

5.2. Results of sensitivity analysis using simulations

In Fig. 8 it is shown that the proposed method performs well at different angles. The lower performance at lower camera angles is caused by the people occluding each other. The proposed tracker still handles shallow angles very well as it only loses track if an object is completely occluded for many frames.

The results in Fig. 9 show a strong increase in the MOTA score when adding a second range camera, while the addition of a third or fourth camera does not bring a notable gain. In the tested scene, two cameras were enough to eliminate all occlusions. The improvement that an extra camera brings will be scene dependent since it relates directly to the possible occlusions.

Figure 10 shows that the results are stable up to a Gaussian noise with a standard deviation of 50 mm. This is ten times that of a Kinect. The original background subtraction failed with higher noise levels, resulting in a drastic performance drop. This is caused by false positive rates in the foreground detection as the noise causes static objects to “move” too much. To show the performance of the tracker at higher noise levels, the original background subtractor was replaced with the Matlab vision.ForegroundDetector. This is a detector based on Gaussian mixture models. The de-

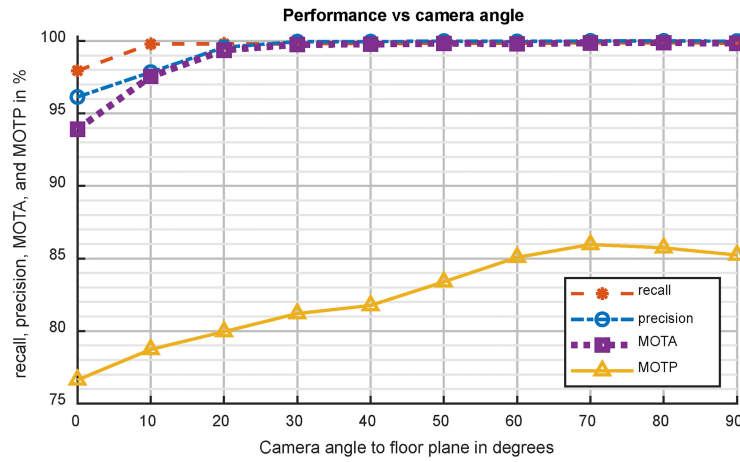


Fig. 8. Tracking performance versus camera angle.

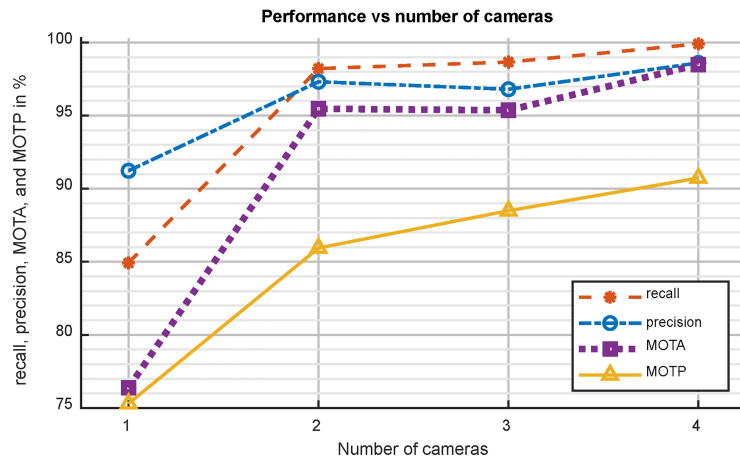


Fig. 9. Tracking performance using one or multiple range cameras.

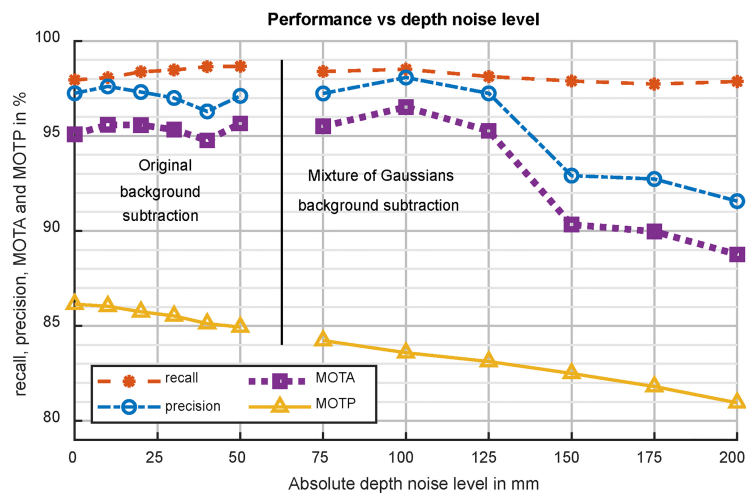


Fig. 10. Tracking performance with different levels of depth noise. For higher noise levels, a different background subtraction method was used.

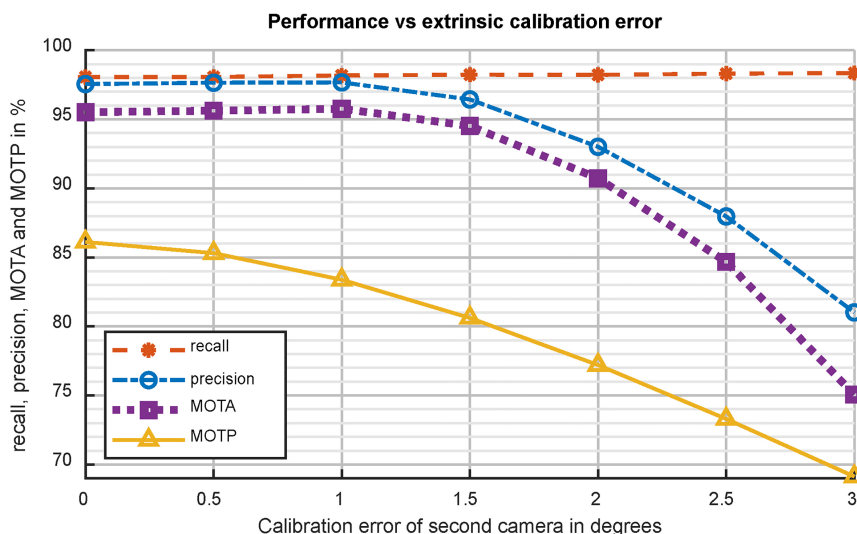


Fig. 11. Tracking performance versus extrinsic calibration error.

fault settings were used, and the number of Gaussians was set to 2. Using this background subtraction, solid MOTA scores were obtained up to a noise level of 125 mm before decreasing below 95%.

The proposed method can handle a calibration error of up to 1° without losing any accuracy, as shown in Fig. 11. This is enough when using an accurate calibration method, such as [18] where an accuracy is obtained of less than 2 mm and 0.6° .

6. Conclusions

Simplicity and robustness were the goals for the designed tracking algorithm. Simplicity is innate in the proposed approach as it is set out in Section 3. Only basic algebra is used to transform the points and combine them in a density map using binning. After simple thresholding, blobs are formed by basic morphological operations. The most “complex” techniques that were used, are the Kalman filter and the Hungarian algorithm in the tracking; both simple lightweight matrix algorithms in practice.

Its robustness is confirmed in the experiments. The simulation experiments show that the proposed method copes well with varying camera angles. A sensitivity analysis for the camera noise and extrinsic calibration error shows a tolerance for depth noise as large as 10 times the noise of the Kinect ToF camera and immunity to a calibration error of up to 1° .

Applying DMT-BS to three datasets, show an overall better performance than two state-of-the-art meth-

ods: DPOM [13] and the Kinect SDK human pose estimator. These results are remarkable as the proposed method is of a much lower complexity than the compared algorithms.

It must be noted that (large) objects that are left behind in the scene, will be labeled as static humans. This is a known limitation of DMT-BS which limits its application to cases where left behind objects are not allowed or it is desired to detect the just like humans. Future work will focus on resolving this.

References

- [1] Benito-Picazo J, Domínguez E, Palomo EJ, López-Rubio E, Ortiz-de-Lazcano-Lobato JM. Motion detection with low cost hardware for PTZ cameras. *Integrated Computer-Aided Engineering* 2018, 26: 21–36. doi: 10.3233/ICA-180579.
- [2] Hsu W-Y. Automatic pedestrian detection in partially occluded single image. *Integrated Computer-Aided Engineering* 2018, 25: 369–79. doi: 10.3233/ICA-170573.
- [3] Lacabex B, Cuesta-Infante A, Montemayor AS, Pantrigo JJ. Lightweight tracking-by-detection system for multiple pedestrian targets. *Integrated Computer-Aided Engineering* 2016, 23: 299–311. doi: 10.3233/ICA-160523.
- [4] Ciarelli PM, Salles EOT, Oliveira E. Human automatic detection and tracking for outdoor video. *Integrated Computer-Aided Engineering* 2011, 18: 379–90. doi: 10.3233/ICA-2011-0383.
- [5] Li D, Xu L, Goodman ED, Xu Y, Wu Y. Integrating a statistical background-foreground extraction algorithm and SVM classifier for pedestrian detection and tracking. *Integrated Computer-Aided Engineering* 2013, 20: 201–16. doi: 10.3233/ICA-130428.
- [6] Wu S, Zhang G, Zhu M, Jiang T, Neri F. Geometry based three-dimensional image processing method for electronic

- cluster eye. *Integrated Computer-Aided Engineering* 2018, 25: 213–28. doi: 10.3233/ICA-180564.
- [7] Tirronen V, Neri F, Kärkkäinen T, Majava K, Rossi T. An enhanced memetic differential evolution in filter design for defect detection in paper production. *Evolutionary Computation* 2008, 16: 529–55. doi: 10.1162/evco.2008.16.4.529.
- [8] van Oosterhout T, Englebienne G, Kröse B. RARE: people detection in crowded passages by range image reconstruction. *Machine Vision and Applications* 2015, 26: 561–73. doi: 10.1007/s00138-015-0678-x.
- [9] Stahlschmidt C, Gavrilidis A, Velten J, Kummert A. Applications for a people detection and tracking algorithm using a time-of-flight camera. *Multimedia Tools and Applications* 2016, 75: 10769–86. doi: 10.1007/s11042-014-2260-3.
- [10] Burbano A, Bouaziz S, Vasiliu M. 3D-sensing distributed embedded system for people tracking and counting. In: *Proceedings – 2015 International Conference on Computational Science and Computational Intelligence, CSCI 2015*, 2016, pp. 470–5. doi: 10.1109/CSCI.2015.76.
- [11] Munaro M, Menegatti E. Fast RGB-D people tracking for service robots. *Autonomous Robots* 2014, 37: 227–42. doi: 10.1007/s10514-014-9385-0.
- [12] Munaro M, Basso F, Menegatti E. OpenPTrack: Open source multi-camera calibration and people tracking for RGB-D camera networks. *Robotics and Autonomous Systems* 2016, 75: 525–38. doi: 10.1016/j.robot.2015.10.004.
- [13] Bagautdinov T, Fleuret F, Fua P. Probability occupancy maps for occluded depth images. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE; 2015, pp. 2829–37. doi: 10.1109/CVPR.2015.7298900.
- [14] Harville M. Stereo person tracking with adaptive plan-view templates of height and occupancy statistics. *Image and Vision Computing* 2004, 22: 127–42. doi: 10.1016/j.imavis.2003.07.009.
- [15] Muscoloni A, Mattoccia S. Real-time tracking with an embedded 3D camera with FPGA processing. In: *2014 International Conference on 3D Imaging (IC3D)*, IEEE; 2014, pp. 1–7. doi: 10.1109/IC3D.2014.7032593.
- [16] Van Crombrugge I, Azza IB, Penne R, Van Barel G, Vanduit S. Fast ground detection for range cameras on road surfaces using a three-step segmentation. vol. 10617 LNCS. Springer, Cham; 2017. doi: 10.1007/978-3-319-70353-4_41.
- [17] Foix S, Alenya G, Torras C. Lock-in Time-of-Flight (ToF) cameras: a survey. *IEEE Sensors Journal* 2011, 11: 1917–26. doi: 10.1109/JSEN.2010.2101060.
- [18] Fornaser A, Tomasin P, De Cecco M, Tavernini M, Zanetti M. Automatic graph based spatiotemporal extrinsic calibration of multiple Kinect V2 ToF cameras. *Robotics and Autonomous Systems* 2017, 98: 105–25. doi: 10.1016/j.robot.2017.09.007.
- [19] Munkres J. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics* 1957, 5: 32–8. doi: 10.1137/0105003.
- [20] Rohmer E, Singh SPN, Freese M. V-REP: a versatile and scalable robot simulation framework. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE; 2013*, pp. 1321–6. doi: 10.1109/IROS.2013.6696520.
- [21] Terven JR, Córdova-Esparza DM. Kin2. A Kinect 2 toolbox for MATLAB. *Science of Computer Programming* 2016, 130: 97–106. doi: 10.1016/j.scico.2016.05.009.
- [22] Corti A, Giancola S, Mainetti G, Sala R. A metrological characterization of the Kinect V2 time-of-flight camera. *Robotics and Autonomous Systems* 2016, 75: 584–94. doi: 10.1016/j.robot.2015.09.024.
- [23] Bernardin K, Stiefelwagen R. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP Journal on Image and Video Processing* 2008; 1–10. doi: 10.1155/2008/246309.
- [24] Milan A, Leal-Taixe L, Reid I, Roth S, Schindler K. MOT16: A Benchmark for Multi-Object Tracking 2016.
- [25] Milan A, Ristani E. MOTChallenge DevKit 2018. <https://motchallenge.net/devkit>.