Machine Vision and Applications (2016) 27:303–315
DOI 10.1007/s00138-015-0739-1

SHORT PAPER

Counting pedestrians with a zenithal arrangement of depth cameras

Pablo Vera[1] · **Sergio Monjaraz**[1] · **Joaquín Salas**[1]

Received: 27 February 2015 / Revised: 4 November 2015 / Accepted: 9 November 2015 / Published online: 14 December 2015

Abstract Counting people is a basic operation in applications that include surveillance, marketing, services, and others. Recently, computer vision techniques have emerged as a non-intrusive, cost-effective, and reliable solution to the problem of counting pedestrians. In this article, we introduce a system capable of counting people using a cooperating network of depth cameras placed in zenithal position. In our method, we first detect people in each camera of the array separately. Then, we construct and consolidate tracklets based on their closeness and time stamp. Our experimental results show that the method permits to extend the narrow range of a single sensor to wider scenarios.

Keywords Counting pedestrians · Depth cameras · Cameras in zenithal position · Network of cameras

1 Introduction

Counting people is a basic operation for applications that includes surveillance, marketing, services, among others. For instance, imagine a scenario in which people enter a Metro subway station [28]. Knowing how many people there are on the platforms may enable administrative officials to send trains with enough frequency to establish a balance between the users' waiting time and the company's expenses. In another scenario, knowing the number of people entering a building can support the decision making processes for civil protection authorities to react with enough information in the event of an evacuation [8]. In some situations, mechanical turnstiles have been used but they tend to be intrusive. For situations where a no-contact solution is preferable, manual counting with a clicker could be an option, although a labor-intensive one. On the other hand, pass/no pass solutions with infrared sensors are attractive but error prone because there is a lack of a high-level understanding of the scene. Recently, computer vision techniques have emerged as a non-intrusive, cost-effective, and reliable solution to the problem of counting pedestrians [3,5,25]. In this article, we are interested on scenarios where several counters, based on the use of depth images, are placed in a zenithal position and operate simultaneously (see Fig. 1). This type of configuration could be required, for instance, in wide corridors where a single camera does not have a large enough field of view either because the ceiling is too low or because, due to its technical limitations, the sensor cannot obtain depth measurements from a sufficiently high viewpoint.

In the rest of the document, we detail our approach to the problem. In the next section, we frame it within the related literature, describing how research has been advancing the understanding of its different facets. Then, in Sect. 3, we detail how a single counter operates in our approach. Interestingly, our method has high performance and low computing requirements, both properties make it amenable for its implementation in off-the-shelf Single Board Computers (SBC). In Sect. 4, we tackle different aspects derived from the simultaneous use of multiple counters, e.g., calibration of camera parameters, expression of measurements in a global reference system, and the final expression of counts. Next, in Sect. 5, we show some experimental results where we compare the performance of our people detection method with related alternatives, and the use of a single pedestrian counter with a multi-camera system. In addition, we detail how the main parameters of our method are selected, discuss its scalability,

Joaquin Salas is on sabbatical leave at FI-UAQ.

✉ Joaquín Salas
salas@ieee.org; jsalasr@ipn.mx

[1] CICATA Querétaro, Instituto Politécnico Nacional, Cerro Blanco 141, Colinas del Cimatario, 76090 Querétaro, Mexico

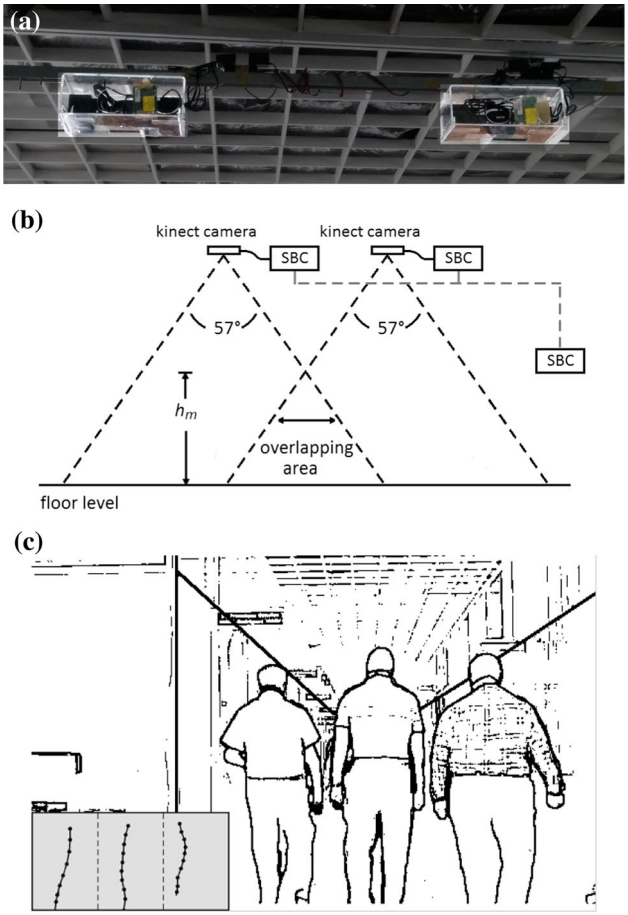

Fig. 1 Counting pedestrians with a zenithal arrangement of depth cameras. An array of two or more cameras can be used to extend the area covered by a single one. **a** Two counters in their housing. **b** A single board computer (SBC) connected to each camera is used to detect pedestrians. Another SBC is used to summarize information and count them. In the case of a Kinect camera, the horizontal field of view covers about 57°. The overlapping area starts at a height h_m above the floor level. **c** Three persons walk in a wide area not covered by a single camera. Their paths are plotted on the *rectangle* at the *left-bottom corner*, where the *dashed lines* separate the area in three regions corresponding to the areas that are visible by exclusively one camera (*left* and *right* regions) and the overlapping area (region at the *middle*). In this case, the SBCs connected to a camera can detect only two persons each, while the third SBC combines the detections, and combine them into paths to count the three persons

and illustrate some special cases. Finally, we conclude and describe future directions of research.

2 Related literature

Currently, detection and regression have emerged as two clearly defined sets of techniques to count people. In detection-based techniques, machine learning methods are used to train a classifier, or to create a discriminant rule, which in turn is used to assess whether an instance of the object is in the image. A review of these techniques is presented by Dollar et al. [3]. The databases they used for testing include pedestrians located in position fronto-parallel to the camera, with large variations in scale, and where occlusion is very frequent. In general, promising results have been obtained for classifiers based on Adaboost [21,28] and SVM [4]. Still, there is some space for improvement for the methods when confronted with images containing partially occluded pedestrians.

At their end, regression techniques employ machine learning techniques to infer crowd size from image features. These methods can be used when there is a considerable level of occlusion, the images have low resolution, or when privacy concerns rise high. Ryan et al. [25] present a review of these methods. Some of the approaches include Gaussian process regression (GPR) [1,27], linear regression [24], K-nearest neighbors [16], and neural networks [2,11]. These results show that local features, where regression is solved locally, outperform holistic features, where global image features are used to describe each frame, and that GPR shows an interesting edge over the other methods. Either using detection or regression methods, Ferryman and Ellis stress the importance of using standardized data sets and present an evaluation of performance for crowd analysis methods using the PET2009 database [5].

Spinello and Arras [26] detect people from a combination of depth and color images obtained from a lateral view point. From the depth images, they obtain histograms of oriented depths. From the color images, they compute the histogram of oriented gradients. Both descriptors are fed to SVM classifiers and their corresponding resulting margins are combined probabilistically. Also working on the problem of detecting people in frontal view images, Yu et al. [29] proposed the simplified local ternary patterns (SLTP) descriptor. This descriptor corresponds to codes extracted from the relationship between the horizontal and vertical depth differences with respect to a user-defined threshold. The descriptor is feed into an SVM for classification. Focusing on the problem of detecting people from zenithal cameras, Rauter [23] compared SLTP features with a method he proposed. His method identifies candidates by locating peaks in the depth map, refines the estimation of the position of their centers, computes a histogram of depth differences as descriptor, and uses this descriptor to train an SVM classifier to detect people. Tracking is done using a mixture of criteria that includes primarily nearest neighbor association, and which is complemented with heuristics combining tracks and detections. At their end, using Haar-like features based on the head and shoulder profile and an Adaboost classifier, Zhu and Wong [33] introduce a method for people detection from top-view depth images. After detection, they use a Kalman filter to track trajectories.

Recently, some researchers have explored the use of depth sensors in zenithal arrangements to count people. For

instance, Zhang et al. [31] detect heads as regions with a well. They illustrate their method as a water filling-like process where water rises uphill and travel downwards until can descend no more. Then, the amount of water in a particular well determines whether a person has been detected or not. Similarly, Galčík and Gargalík [7] find people by detecting maxima in the depth images followed by region growing, which is limited by a predefined depth threshold. The regions found are considered heads if they pass criteria related to size, roundness, and evidence of being above shoulder-like structures. At our end, we perform a morphology geodesic reconstruction [20] to eliminate candidates which depth is below a certain threshold. Then, we declare positive detection using size and fitness to a circle. Although, we stress that our main contribution is in the use of multiple counters, based on depth images, in a zenithal arrangement, we show that our head detection method exhibits a slightly better performance than the previous two methods in terms of precision and recall and still is able to execute at a significant number of frames per second in an SBC.

Most often, the use of a network of cameras has been applied to reduce the effects of occlusion. For instance, Maddalena et al. [16] present a method where the number of pedestrians is inferred from the projection of the foreground in a previously calibrated network of cameras. In their work, the cameras are placed lateral to the scene and they cooperate to overcome occlusion. In our problem, we use the network of cameras with an emphasis in an increment of the field of view. Other uses of depth cameras in zenithal arrangements include the prediction of the occurrence of a pedestrian through a network of sensors, by Porzycki et al. [22]. Once a pedestrian is detected in one camera, his/her movement parameters are estimated and a simulated agent is initialized. As the scene progresses, another camera may realize that the same individual is being observed based on his/her expected occurrence in its field of view. There is not an area of overlap between the sensors field of view. To our knowledge, ours is the first reported system that count people using a cooperating network of depth cameras placed in zenithal arrangement.

3 A single counter

In our method, we first detect people with each camera of the array separately. We associate each detection to a single pixel location on the image and use tracklets to trace the path of a person observed from a single camera. A tracklet is a chronologically ordered sequence of observations of a single identity. On a tracklet, we record the locations where detections occur and the time stamp of the image was captured. For cameras with overlapped field of view, we propose a method, discussed in Sect. 4, to combine tracklets corresponding to the same identity.

3.1 People detection

We perform the detection step on each frame captured by the depth camera. Typically, depth images contain some regions with missing depth values for objects that are not within the device operating range. Specifically, this may be mainly due to occlusions at object edges, direct sunlight on the scene and noise due to a low infrared radiation level reflected by an object. To solve this issue, we fill these regions by interpolating from the known values using the depth map recovery method of Galčík and Gargalík [7].

On a depth image, $\mathbf{Z}$, taken by a camera with zenithal view (Fig. 2a), several objects, including heads of people, produce regions with deeper depths than the background. From $\mathbf{Z}$, we obtain an image, $\mathbf{Z}_h$, by increasing the depth of each local minimum by a value of h or until the depth of the region border is reached (see Fig. 2b). Then, we apply morphology geodesic reconstruction [20], which basically consists on segmenting out regions satisfying $\mathbf{Z}_h - \mathbf{Z} \geq \tau$ where $0 \leq \tau \leq h$ is a predefined value. We use a value of h that is somewhat less than the maximal expected difference between a person's height and the height of his shoulders. This allows that when a person passes below the camera only the region corresponding to his head will be segmented (Fig. 2c). The value of τ should be chosen to segment most of the region which could correspond to a person's head, but to avoid the segmentation of regions for which the difference in depth between their borders and the local minimum is small. We further filter out regions belonging to the background (i.e., not moving objects) using the background subtraction algorithm of Zivkovic [34], updating the background model with each new frame. As an illustration, consider the dashed horizontal red line in Fig. 2a. The depth profile is illustrated in Fig. 2b. There, each local minimum depth is increased by $h = 0.2$ m defining the blue profile $\mathbf{Z}_h$. After the logic expression $\mathbf{Z}_h - \mathbf{Z} \geq \tau$ is evaluated, with $\tau = 0.1$ m (Fig. 2c), we are left with some detected head candidates (Fig. 2d).

For the remaining regions, $\Lambda = \{A_1, \ldots, A_r\}$, we consider as pedestrians only those satisfying a criterion of size and shape. We apply the constraint $A_{\min} \leq \texttt{Area}(A_i) \leq A_{\max}$ to a region which projected on the floor is $\texttt{Area}(A_i)$, where $A_{\min}$ and $A_{\max}$ are the minimum and maximum allowed values, respectively. For each region that satisfies this constraint, we extract the coordinates of the region's respective contour pixels and apply RANSAC [6] to find the circumference that best fits the contour points. We use two parameters, r and t, to evaluate the shape criterion: a head is detected if the number of contour pixels that fall within a distance of r pixels from the circumference is equal to or greater than t. As a result, we record the center of the fitted circumference as a single point location. The algorithm stops processing the current frame when there are no

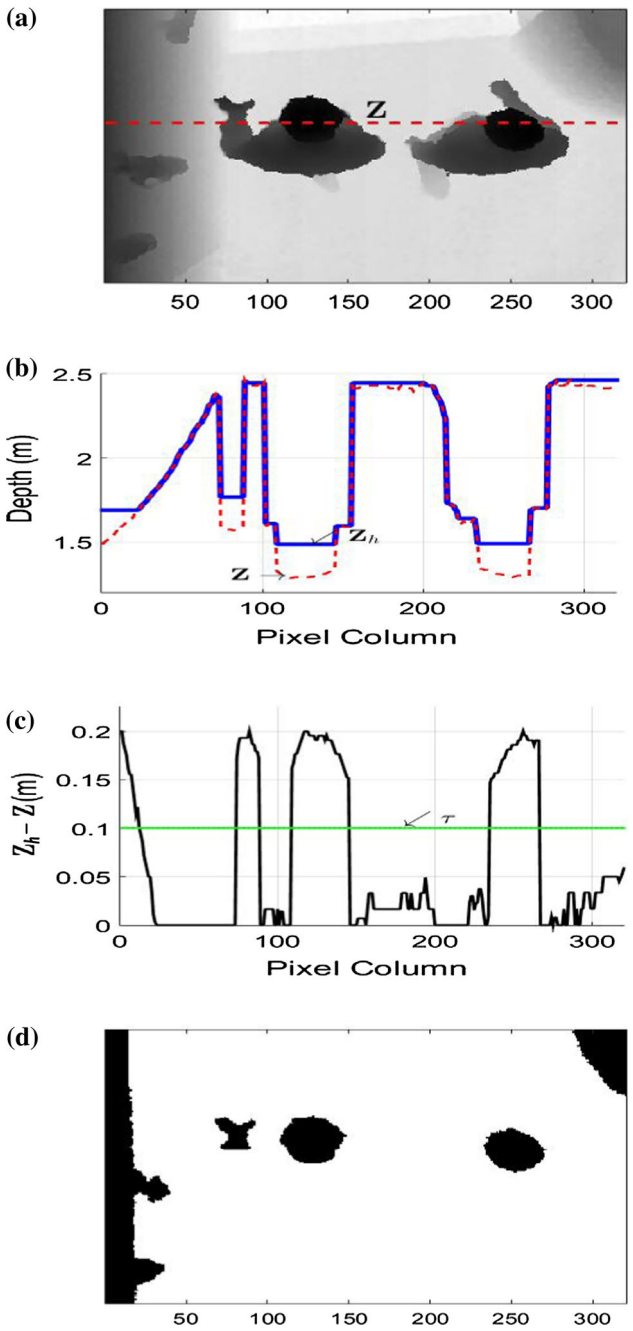

Fig. 2 People's head detection (best seen in color). Our method is based on morphology geodesic reconstruction [20]. As an illustration, consider the *dashed horizontal red line* in **a**, **Z**. The local minima are raised by a maximum of $h = 0.2$ m, resulting in $\mathbf{Z}_h$ (**b**). Head candidates are obtained by applying a threshold $\tau = 0.1$ m on the difference $\mathbf{Z}_h - \mathbf{Z}$ (**c**). In **d**, we show the result of applying the corresponding 2D procedure to the image in **a**

more remaining regions to be analyzed. Algorithm 1 provides another perspective about how this process is carried out. Each unit is in charge of detecting pedestrians as observed from its camera. A global unit coordinates the detection of pedestrians across the different units, as detailed in Algorithm 2.

Call : $\Omega \leftarrow$ `DetectHeads` (**Z**)
input : A depth image **Z**
output: A set of positions Ω where the heads were detected

$\Omega = \{\}$;
//fill holes in **Z** using [7]
Z $\leftarrow$ `FillRegions` (**Z**);
//increase the depth of **Z** by h
$\mathbf{Z}_h \leftarrow$ `FillLocalMinima` (**Z**, h);
//segment out regions satisfying $\mathbf{Z}_h - \mathbf{Z} \geq \tau$
$\Lambda \leftarrow$ `MorphologyGeodesicReconstruction` ($\mathbf{Z}_h$, **Z**, h);
//for all remaining regions ...
for $i \leftarrow 1$ ***to*** $\|\Lambda\|$ **do**
  //process region A_i only if it is within a certain size
  if $A_{\min} \leq$ `Area`$(A_i) \leq A_{\max}$ **then**
    //accept a detection only if within r pixels from the circumference defined by the contour of A_i there are at least t pixels
    if `EnoughBorderPixels` (A_i, r, t) **then**
      //the center of region A_i is recorded
      $\Omega \leftarrow \Omega \bigcup$ `Center` (A_i)
    end
  end
end

Algorithm 1: Detect heads from depth images

Call : `GetHeadPositionsFromUnit` (κ)

repeat
  //get a depth image **Z**
  Z $\leftarrow$ `CaptureImage` ;
  //detect head positions from depth image **Z**, as in Sect. 3.1
  $\Omega \leftarrow$ `DetectHeads` (**Z**);
  //send the head positions to the counter
  `SendInformationToCounter` (Ω);
until;

Algorithm 2: Each local unit κ job is to detect heads out of the depth image stream. In turn, the detected head positions are sent to the global counter

3.2 People tracking

In one tracklet, we list the frame numbers, the capture time stamps, and the pixel coordinates where a person was detected. A tracklet is terminated when it is not updated with a new observation during a given period, $t_{\max}$, corresponding to a few frames, while a new tracklet is created when a new observation could not be assigned to an existing one. For the tracklet assignment, we use a complete bipartite graph model to solve the problem of assigning detections to tracklets. Assuming that at some frame, n detections were made and there are m tracklets, we create a matrix $\mathbf{D}_{m \times n}$ containing the distances between each detection and the last observation of each tracklet. $\mathbf{D}_{m \times n}$ is used as a cost matrix where values greater than a maximum allowed distance $d_{\max}$ are replaced by ∞ and the minimal cost assignment was computed using the Hungarian algorithm [12]. We should point out that this assignment method may fail in the case where

people exchange places from one frame to the next. In such a case, the Hungarian algorithm will conclude that the people did not move. This may happen specially in systems with a low-processing frame-rate or where people are moving fast. In addition, although the nearest neighbor rule properly solves the assignment problem between detections and tracklets in the vast majority of cases, it may give rise to tough assignment problems. Rauter [23] provides additional association rules that may prove to be very useful for these cases. Algorithm 3 provides another perspective about how this process is carried out.

Call : $\mathcal{T}_\kappa \leftarrow$ `PeopleTracking` $(\mathcal{T}_\kappa, \mathbf{Q})$
input : The head positions in $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_n]$, for $\mathbf{q} \in \mathcal{R}^3$ and the m in tracklets $\mathcal{T}_\kappa = \{\tau_1, \ldots, \tau_m\}$ for unit κ
output: Updated tracklets $\mathcal{T}_\kappa$ for unit κ

//compute the distance from tracklets $\mathcal{T}_\kappa$ end-points to the head positions $\mathbf{Q}$ placing results in $m \times n$ matrix $\mathbf{D}$
$\mathbf{D} \leftarrow$ `Distance` $(\mathcal{T}_\kappa, \mathbf{Q})$;
//distances greater than $d_{\max}$ are considered unfeasible
$\mathbf{D}(\mathbf{D} > d_{\max}) \leftarrow \infty$;
//apply the Hungarian algorithm [12] to solve the assignment of observations to tracklets
$\mathcal{T}_\kappa^+ \leftarrow$ `Hungarian-Algorithm` $(\mathcal{T}_\kappa, \mathbf{D})$;
//update tracked tracklets, add new tracklets and mark unused tracklets from $\mathcal{T}_\kappa$.
$\mathcal{T}_\kappa \leftarrow$ `UpdateTracklets` $(\mathcal{T}_\kappa, \mathcal{T}_\kappa^+, \mathbf{D})$;

Algorithm 3: Track people through time. This process runs in the global counter but works with the data provided by each unit κ

The single counter is useful in spaces where the workspace is entirely visible to the camera, even at the height of very tall people, e.g., narrow corridors. In this work, we extend the usable range of the counter to wide corridors by the use of two or more cameras placed in an horizontal arrangement and with a separation between two adjacent cameras by at least the range of one camera (see Fig. 1). The problem consists in combining the individual counts made with each unit using a global counter for the full array of cameras. From the sum of individual counts, we must subtract a count whenever a person walks below the overlapping fields of view of two cameras and thus avoid that he/she is counted twice.

4 Multiple counters

When using multiple cameras with shared fields of view, a prime problem is to determine whether two tracklets created from adjacent cameras correspond to the same person. In other words, we explain our approach to the problem in terms of a camera pair. We show that the extension to multiple cameras corresponds ultimately to a cascade of transformations between local reference systems. Thus, our strategy is to transform all local tracklets to a global reference system defined, without loss of generality, as centered in one of the cameras of the network. Then, if the distance between a pair of them is less than or equal to a given maximum value, we establish that the tracklets correspond to the same person and she/he is counted once, otherwise, we count two pedestrians. Some aspects need to be accounted for: (1) the cameras' intrinsic parameters must be calibrated to obtain the pixel coordinates of the observations, (2) the cameras' extrinsic parameters, i.e., a rotation $\mathbf{R}$ matrix and a translation $\mathbf{t}$ vector are required to do point transformations in 3D space, and (3) both cameras must be clock synchronized as the image capture time is used to compute the distance between tracklets.

4.1 Camera calibration

RGBD camera calibration is an important issue in problems such as measurement, reconstruction, recognition, and others alike. To address it, Herrera et al. [10] introduced a method to obtain simultaneously the calibration of two color cameras, one depth camera, and their relative pose. In addition, they introduced a depth distortion model for the depth sensor. At their end, Zhang and Zhang [30] presented a maximum likelihood approach. Their method naturally provides an assessment of uncertainty in the estimation of the parameters, although it requires a reliable estimation of the depth mapping function and the depth camera noise model. Since the intrinsic parameters for a camera need to be obtained only once, Mikhelson et al. [19] proposed a method to quickly obtain the value for the extrinsic parameters. In their approach, they find a map between the color and the depth cameras. Then, using corners detected in the color image, they find the 3D information for the corresponding points in the depth camera. Still, akin to our work, Macknojia et al. [15] find the relationship between pairs of sensors. To avoid the small holes in the depth map caused by interference, they collect images over different time spans.

We calibrated the cameras using the method of Herrera et al. [10] and their toolbox for Matlab. Using this result, the problem is to find the location $[u^1, v^1]^T$, in the image of the first camera, of points $[u^2, v^2]^T$ taken with the second one. In between, we need to compute the 3D space location $\mathbf{p}^2 = [X^2, Y^2, Z^2]^T$, in the reference frame $\{2\}$ of the second camera, using its intrinsic parameters. Then, the 3D space location of the point $\mathbf{p}^1$, in the first camera reference system $\{1\}$, is given by:

$$\mathbf{p}^1 = \mathbf{R}_2^1 \mathbf{p}^2 + \mathbf{t}^1, \tag{1}$$

where $\mathbf{R}_j^k$ expresses the rotation to align reference systems $\{j\}$ and $\{k\}$, and $\mathbf{t}^k$ is a translation expressed in the reference system $\{k\}$.

4.2 A global reference system

Once we setup the camera array, we determined $\mathbf{R}_2^1$ and $\mathbf{t}^1$ using a planar chessboard pattern with black and white squares. We took color and depth images placing the pattern in the overlapping area of two cameras for different orientations. $\mathbf{R}_2^1$ was computed using the depth images.

In this case, only the planar nature of the pattern is important since the corners are not visible on the depth images and other planar object could be used. For each image, we manually selected an area inside the pattern and determined the coordinates corresponding to the selected points in 3D space. The vector $[a, b, c]^T$ normal to the plane is obtained from the plane equation: $aX + bY + cZ + d = 0$, where $[X, Y, Z]^T$ is the location of a point on the chessboard pattern. The parameters of the plane are obtained using least squares fitting as follows. Let $\mathbf{s}_i^1 = [a_i^1, b_i^1, c_i^1]^T$ and $\mathbf{s}_i^2 = [a_i^2, b_i^2, c_i^2]^T$ be the normals for the ith plane orientation using the reference systems of the first, {1}, and second camera, {2}, respectively. In addition, let $\mathbf{M} = [\mathbf{s}_1^1, \mathbf{s}_2^1, \ldots, \mathbf{s}_n^1]$ and $\mathbf{N} = [\mathbf{s}_1^2, \mathbf{s}_2^2, \ldots, \mathbf{s}_n^2]$ be $\mathcal{R}^{3\times n}$ matrices containing n orientations. Then, there is a 3×3 matrix $\mathbf{Q}$ for which $\mathbf{M} \approx \mathbf{QN}$. The least squares solution for this system is given by $\mathbf{Q} = \mathbf{MN}^T(\mathbf{NN}^T)^{-1}$. $\mathbf{R}_2^1$ is obtained from $\mathbf{Q}$ which, in general, does not satisfy the properties of a rotation matrix, using singular value decomposition [9], i.e., $\mathbf{R}_2^1 = \mathbf{UV}^T$, where matrices $\mathbf{U}$ and $\mathbf{V}$ are the left and right eigenvector matrices of the singular value decomposition of $\mathbf{Q}$.

The translation $\mathbf{t}^1$ was computed using the color images, assuming that the result could be extended to the depth cameras. In this case, the corners of the chessboard pattern are used as reference points. There is an uncertainty associated to this assumption due to the fact that for two particular kinects the relative position and orientation between color and depth cameras may differ somewhat. This is why we preferred to use images captured with the depth cameras to obtain $\mathbf{R}_2^1$, while the use of depth images, in the calculation of $\mathbf{t}^1$, introduces the problem of locating features with precision. For each image, a homograph, $\mathbf{H}$, and the intrinsic parameter matrix, $\mathbf{A}$, were obtained using the method of Zhang [32]. $\mathbf{H}$ is a matrix that relates the location of points in a plane in the 3D space and in the reference system of the plane, with the pixels in the image. Let $\mathbf{R}$ and $\mathbf{t}$ be the extrinsic parameters (rotation and translation) that express points in the plane in the camera reference system and let $\mathbf{h}_i$ and $\mathbf{r}_i$ the ith column vector of $\mathbf{H}$ and $\mathbf{R}$, respectively.

Then, Zhang [32] showed that the extrinsic parameters can be obtained with the expressions

$$\begin{aligned} &\mathbf{r}_1 = \lambda\mathbf{A}^{-1}\mathbf{h}_1, \quad \mathbf{r}_2 = \lambda\mathbf{A}^{-1}\mathbf{h}_2, \quad \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2, \\ &\mathbf{t} = \lambda\mathbf{A}^{-1}\mathbf{h}_3, \end{aligned} \tag{2}$$

with $\lambda = 1/\|\mathbf{A}^{-1}\mathbf{h}_1\| = 1/\|\mathbf{A}^{-1}\mathbf{h}_2\|$. Now, let $\{\mathbf{R}_p^1, \mathbf{t}_p^1\}$ and $\{\mathbf{R}_p^2, \mathbf{t}_p^2\}$ be the extrinsic parameters of the first and second cameras, respectively, relative to the reference system $\{p\}$, defined by a particular position and orientation of the pattern. Also, let $\mathbf{p}^p = [X^p, Y^p, Z^p]^T$, $\mathbf{p}^1 = [X^1, Y^1, Z^1]^T$ and $\mathbf{p}^2 = [X^2, Y^2, Z^2]^T$ be the coordinates of a corner using the reference systems of the chessboard pattern $\{p\}$, first camera {1} and second camera {2}, respectively. The position of the point $\mathbf{p}^p$ in the reference systems, {1} and {2}, of the cameras is given by

$$\mathbf{p}^1 = \mathbf{R}_p^1\mathbf{p}^p + \mathbf{t}_p^1, \quad \text{and} \quad \mathbf{p}^2 = \mathbf{R}_p^2\mathbf{p}^p + \mathbf{t}_p^2. \tag{3}$$

Solving both equations for $\mathbf{p}^p$ results in

$$\begin{aligned} &\mathbf{p}^p = \mathbf{R}_1^p\left(\mathbf{p}^1 - \mathbf{t}_p^1\right) = \mathbf{R}_2^p\left(\mathbf{p}^2 - \mathbf{t}_p^2\right) \\ &\therefore \mathbf{p}^1 = \mathbf{R}_p^1\mathbf{R}_2^p\mathbf{p}^2 - \mathbf{R}_p^1\mathbf{R}_2^p\mathbf{t}_p^2 + \mathbf{t}_p^1. \end{aligned} \tag{4}$$

Note that we have made use of the notation $\mathbf{R}_i^j = \left(\mathbf{R}_j^i\right)^T = \left(\mathbf{R}_j^i\right)^{-1}$. As a result, we obtain $\mathbf{t}^1$ as the average of $\mathbf{t}_p^1 - \mathbf{R}_p^1\mathbf{R}_2^p\mathbf{t}_p^2$ for all plane orientations. In general, for a larger number of cameras, we just need to propagate the measurements through the different coordinate systems. For instance, suppose there are n cameras, and that the global reference system coincides with the reference system of camera one, then a point $\mathbf{p}^n$ in reference system $\{n\}$ is expressed in the reference system {1} as

$$\mathbf{p}^1 = \sum_{i=1}^{n}\left[\prod_{j=1}^{i-1}\mathbf{R}_{j+1}^j\right]\mathbf{t}^i, \tag{5}$$

and $\mathbf{t}^n = \mathbf{p}^n$. Please note that most of the products can be computed in advance, once the rotations and translations are known.

In practice, as there is no overlapping between cameras that are not adjacent, points are only projected from one camera to its adjacent, using (1). This is possible because our primary interest is counting, not mapping, and because this avoids error propagation, thus promoting scalability.

4.3 Fusing tracklets

We measure the distance between two tracklets created from observations of two cameras made at overlapping time intervals. Assuming that at time t the expected locations of the observations are $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$, then we determine $\delta_t = \|\mathbf{x}_2(t) - \mathbf{x}_1(t)\|$ as the distance between the two tracklets at time t. Since observations with each camera are not done regularly at the same time, we find $\mathbf{x}_1(t)$ by interpolating from two consecutive observations $\mathbf{x}_1(t_i)$ and $\mathbf{x}_1(t_{i+1})$,

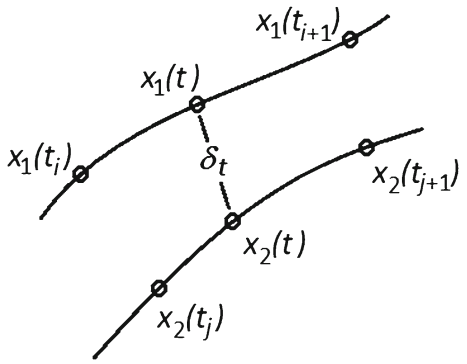

Fig. 3 Distance between two tracklets at time t. Assuming that at time t the expected locations of the observations are $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$, then we determine $\delta_t = \|\mathbf{x}_2(t) - \mathbf{x}_1(t)\|$ as the distance between the two tracklets at time t

and $\mathbf{x}_2(t)$ from $\mathbf{x}_2(t_j)$ and $\mathbf{x}_2(t_{j+1})$, where $t_i \leq t \leq t_{i+1}$ and $t_j \leq t \leq t_{j+1}$ (see Fig. 3). We evenly divide the time interval where the tracklets coincide into a number n of values for t and compute the distance between the tracklets as the minimum value of δ_t.

Clock synchronization between the computers becomes essential. To that effect, we installed a time server in one SBC and clients for the rest of them. We based the synchronization of the SBC network on the precision time protocol (PTP) IEEE 1588 [13] achieving sub-millisecond precision.

4.4 Counting people

A prime assumption is that our method counts people walking in two distinct and roughly straight opposite directions, e.g., persons walking on a straight corridor in two opposite directions, or entering/leaving a room. As a consequence people making a u-turn, or wandering without reaching the other end, are not counted as either entering or leaving an area.

In this step, the program initializes a bidirectional counter to register the number of persons walking below the camera array in two opposite directions: $\mathcal{A} \rightarrow \mathcal{B}$ and $\mathcal{B} \rightarrow \mathcal{A}$. At the beginning, two variables corresponding to these directions are set to zero: $C_{\mathcal{A},\mathcal{B}} = 0$ and $C_{\mathcal{B},\mathcal{A}} = 0$. Let us consider first the simpler case where only one camera is used. The camera is aligned so that the image vertical axis is oriented along the two main directions of pedestrian motion. For instance $\mathcal{A} \rightarrow \mathcal{B}$ would represent a person moving in the image from the top to the bottom and $\mathcal{B} \rightarrow \mathcal{A}$ in the opposite direction. The counter can be updated whenever a tracklet terminates. For instance, in this setting, let v_i and v_f be the vertical coordinates of the first and last observation of the tracklet. Then, if $v_i < v_f$ we will make $C_{\mathcal{A},\mathcal{B}} \leftarrow C_{\mathcal{A},\mathcal{B}} + 1$, and otherwise if $v_i > v_f$ then $C_{\mathcal{B},\mathcal{A}} \leftarrow C_{\mathcal{B},\mathcal{A}} + 1$. Other constraints could be used to update the counter. For instance, the tracklet should have a minimal number of observations to avoid counting sporadic false detections or, if we do not want to count a pedestrian moving perpendicular to the vertical axis, we may establish the constraint $|v_f - v_i| \geq \delta_{\min}$, where $\delta_{\min}$ is the minimum expected distance in the image along the vertical axis that a person should walk to be counted. Algorithm 4 provides another perspective about how this process is carried out.

```
Call : CountPeople
//rotation ^w_κR and translation ^w_κt of the units κ with respect to the
  global reference system {w} are known from calibration, as in
  § 4.1 and § 4.2
// meta-tracklets variable T will contain the tracklets {T_κ} for
  each unit, for κ = 1, ..., u
T ← {};
//init counters variables C_{A,B} and C_{B,A}
C_{A,B} ← 0; C_{B,A} ← 0;
repeat
    //poll data from all units ...
    for κ = 1 : u do
        //get head positions from units
        Ω ← GetHeadPositionsFromUnit (κ);
        //express measurements in the global reference system
        P ← Reference (Ω, ^w_κR, ^w_κt);
        //track people through time
        T_κ ← PeopleTracking (T_κ, P);
    end
    //fuse tracklets separating those T' = {τ_1, ..., τ_{||T'||}} that are
      terminated, as in §4.3
    < T, T' > ← FuseTracklets (T);
    //for all the terminated tracklets ...
    for i = 1 : ||T'|| do
        //process those larger than o_min
        if ||τ_i|| ≥ o_min then
            //get vertical end-points (v_i, v_f)
            < v_i, v_f > ← VerticalEndPoints (τ_i) //if the
              vertical displacement was larger than δv_min
            if ||v_i − v_f|| > δ_min then
                //update the counts depending on the direction of
                  motion
                if v_i < v_f then
                    C_{A,B} ← C_{A,B} + 1;
                end
                else
                    C_{B,A} ← C_{B,A} + 1;
                end
            end
        end
    end
until;
```

Algorithm 4: Counting people using the data provided by local units

Figure 4 shows the case of an unusual path of a person wandering between the field of views covered by different cameras. Tracklets generated by the data provided by each unit are fused into a single tracklet if the distances between successive tracklets are less than a given value δ_m.

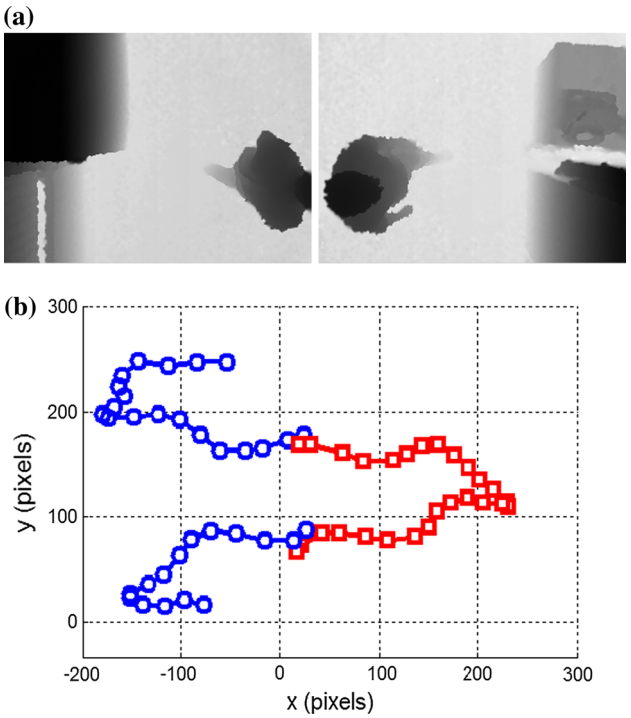

Fig. 4 A person walking more than once in the area covered by different cameras (**a**) generates separate tracklets which can be fused into a single one. In **b**, tracklets are plotted, using the global reference system, with *blue lines with circles* or with the *red line with squares*, whether they are generated with the data provided by one unit or by the other

5 Experimental results

In our experiments, we want to learn both the performance of the people detector in relation to similar methods and how the method to count people compares when two cameras are used instead of one.

Our algorithm was implemented in the Inforce 6410 SBC. These platforms have a 1.7 GHz Quad Core Krait class Snapdragon 600 processor. Using multi-threading, via OpenMP, this SBC is able to sustain processing rates above 10 frames per second on depth images with resolution 240 (rows) × 320 (columns) pixels. As the results below show, we have found this processing speed good enough for the real-time application of the proposed method.

In addition, there is a wealth of research on how to reduce the interference between multiple-structured light depth sensors [14,18]. It has been suggested to apply a small amount of motion to a subset of the sensors so that each unit sees its own projected pattern sharply but sees a blurred version of the patterns of other units. We followed these advices and installed a vibratory motor in some of the Kinects. However, it did not make much of a difference in the performance and ended up not using them at all.

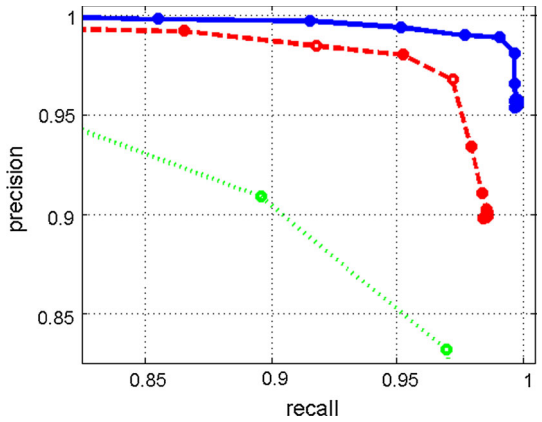

Fig. 5 Precision versus recall for people detection (best seen in color). The performance for Zhang et al. [31] is shown as a *red*, *dashed line*, while Galčík and Gargalík [7] is shown as a *green dotted line*. Our method is shown as a *blue*, *continuous line*. Because the three methods show a high level of performance, we just show the *upper right corner* of the precision–recall curve

5.1 Head detection

For our experiment, we use the dataset provided by Zhang et al. [31],[1] which has 2384 images. Out of those images, we eliminated those where the head was partially visible and was not annotated in the ground truth. It may be worth noting that the criteria we use may be applied to detect partially occluded heads correctly. Nonetheless, this segregation left 1647 images. Then, we implemented the methods proposed by Zhang et al. [31] and Galčík and Gargalík [7]. Certainly, we implemented the algorithms following the description in the reference but we take full credit if there is a mistake in this step. Based on this comment, we do not compare the execution time for the implementations. Then, for Zhang et al.'s method, we varied the amount of water per pixel and performed a contour analysis similar to ours, where we changed the minimal average of count points that could be fitted to a circumference for detection. As for the method proposed by Galčík and Gargalík, we represented the three criteria they use to identify a head with a logistic function and varied the threshold on the product of the descriptors. We tried different parameters for several logistic functions but the performance did not change much. For our method, we changed the parameter t, i.e., a person is detected when the average of pixels of a contour that can be adjusted to a circumference is greater than or equal to t. Figure 5 shows the resulting precision–recall curve. As expected, the three algorithms show a high level of performance.

[1] http://www.cbsr.ia.ac.cn/users/xczhang/CountPeople.html.

5.2 Pedestrian counter

The images in this experiment were captured at the entrance of a building with a resolution of 240 (rows) × 320 (columns) and at a frequency of about 13 frames per second before storage in hard disk for offline analysis. For the experiments, the cameras were placed at 3.1 m above the floor. A single camera, and considering that the Kinect has a horizontal and vertical field of view of 57° and 43°, respectively, will cover a region of 1.30 m (horizontal) × 0.95 m (vertical) at 1.9 m above the floor. In this case, we captured images during 24 h, which resulted in about 1,123,200 of each color and depth images. Then, we ran the algorithm on the depth images while the color images were used to build ground truth by direct observation. Using our algorithm, we classified direction of motion as either inputs (415) or outputs (323) while for the ground truth we obtained 428 and 313, respectively. We computed false negatives, FN, as the number of persons that were not counted, and false positives, FP, as the number of objects counted as persons or persons counted twice, and obtained: FN = 27 and FP = 24, while the number of persons counted correctly, or true positives, was TP = 714. This gives a precision,

$$P = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}), \tag{6}$$

of $P = 96.8$ % and a recall,

$$R = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}), \tag{7}$$

of $R = 96.4$ %. Then, we set an array of two cameras separated about 0.45 m to cover an horizontal region width of 1.75 m at 1.9 m and captured images during 4 h, which resulted in about 187,200 depth and color images. The cameras were clock synchronized, as previously described.

We ran the algorithm processing a sequence of pairs of images captured at the same time at each iteration. Whenever a tracklet ended, the program tried to combine it with the current tracklets observed by the other camera using a distance measure of 30 pixels. If the tracklet was not combined or if it was observed only by one camera, then an event was counted and classified as input or output. For the ground truth we obtained 128 inputs and 173 outputs, while the program counted 125 inputs and 173 outputs. The number of false detections was FP = 3 and FN = 6 and true positives were TP = 295, giving a precision of $P = 99.0$ % and a recall of $R = 98.0$ %. Then, we run the single camera algorithm using the images captured from each camera separately, and the results were, for one camera, 98 inputs and 152 outputs with FP = 3, FN = 54, TP = 247, $P = 98.8$ % and $R = 82.1$ %, and for the other camera, 76 inputs and 93 outputs with FP = 4, FN = 136, TP = 165, $P = 97.6$ % and $R = 54.8$ %.

5.3 Selecting the value for the parameters

To obtain the value for the parameters, we made an analysis to assess the best performance. Using the images selected from the dataset provided by Zhang et al. [31], we determined the true positives (TP), false positives (FP) and false negatives (FN) by comparing detections made with our method and the ground truth, for different values of the parameters τ, h, $A_{\min}$, $A_{\max}$, r and t (see Sect. 3.1 for the description of these parameters). Fixing a subset of the parameters, we obtained the performance of the system using several combinations of related parameters. In a subsequent cycle, we fixed these new values, and some others in the subset, to evaluate other related parameters in the previously fixed subset. The performance was assessed using the F_1 score, which is defined as

$$F_1 = 2\mathrm{TP}/(2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}). \tag{8}$$

Figure 6 shows the cycle for the best performance, which results in values $h = 25$ cm, $\tau = 15$ cm, $A_{\min} = 70$ pixels^2, $A_{\max} = 350$ pixels^2, $t = 0.8$, and $r = 6$ pixels.

Similarly, we obtained the parameters used by the pedestrian counter for tracking, $t_{\max}$ and $d_{\max}$, and for fusing tracklets, δ_m, as the maximal distance between two tracklets candidates to be fused. We ran our algorithm over the 4 h of video captured by the two cameras using different parameter values and determined the curves of precision, recall and F_1 score. Figure 7 shows the cycle for the best performance, which results in values $t_{\max} = 1$ s, $d_{\max} = 60$ pixels, and $\delta_m = 30$ pixels.

5.4 Precision in the global reference system

We used a rectangular, planar, chessboard-like pattern with 9 × 4 internal corners (i.e., those at the intersection of two black and two white squares), with a separation of 3.0 cm between adjacent corners, to compute the rotations and translations between the local counters and a global reference system. To study this aspect of the problem, we first considered the case of a two-camera counter, where a camera reference system is used as global reference and the detections on the other camera are transformed into it. We selected, at random, 207 pairs of depth images from an image dataset where a person's head was entirely visible from both cameras and then we made head detections and applied the transformations. The mean deviation of projecting a point from one image to the other was 7.71 pixels and the standard deviation was 2.85 pixels. Note that a factor affecting the precision is that both cameras detect the same person from different angles. For a counter with more than two cameras, assuming that the separation between cameras is enough to reduce

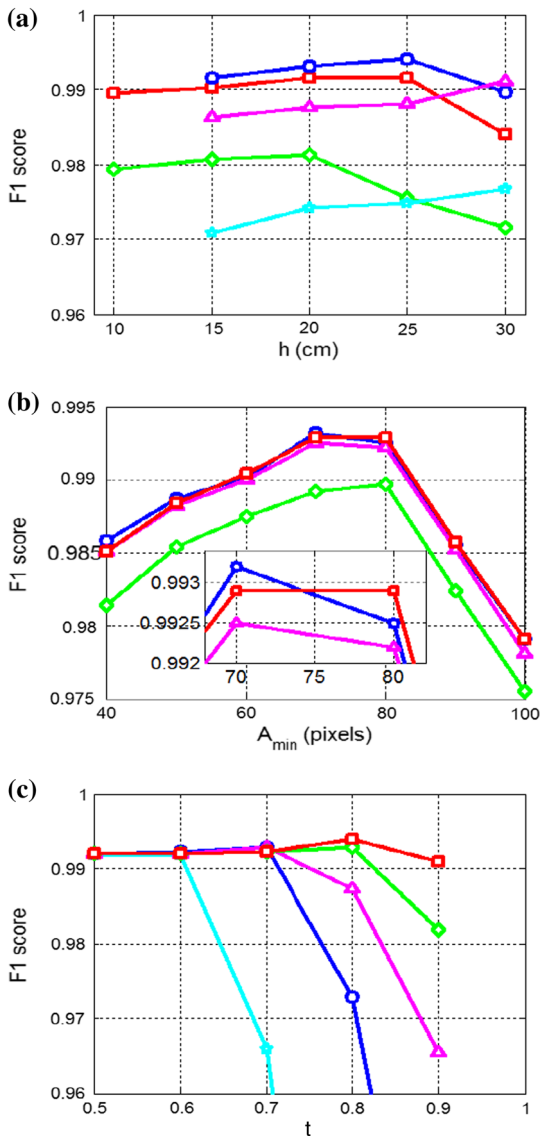

Fig. 6 F_1-score as a function of head detection parameters, h, τ, $A_{\min}$, $A_{\max}$, r and t (see Sect. 3.1 for an explanation of these parameters). This figure is best seen in color. **a** h versus F1-score for $\tau = h - 6$ cm (*green line with diamonds*), $\tau = h - 8$ cm (*red line with squares*), $\tau = h - 10$ cm (*blue line with circles*), $\tau = h - 12$ cm (*magenta line with triangles*), $\tau = h - 14$ cm (*cyan line with pentagrams*), and $A_{\min} = 70$ pixels, $A_{\max} = 350$ pixels, $r = 6$ pixels and $t = 0.8$. Best performance: $h = 25$ cm and $\tau = 15$cm. **b** $A_{\min}$ versus F1-score for $A_{\max} = 300$ pixels (*green line with diamonds*), $A_{\max} = 350$ pixels (*blue line with circles*), $A_{\max} = 400$ pixels (*magenta line with triangles*), $A_{\max} = 450$ pixels (*red line with squares*), and $\tau = 15$ cm, $h = 25$ cm, $r = 6$ pixels and $t = 0.8$. Best performance: $A_{\min} = 70$ pixels $A_{\max} = 350$. **c** t versus F1-score for $r = 2$ pixels *cyan line with pentagrams*), $r = 3$ pixels (*blue line with circles*), $r = 4$ pixels (*magenta line with triangles*), $r = 5$ pixels (*green line with diamonds*), $r = 6$ pixels (*red line with squares*), and $\tau = 15$ cm, $h = 25$ cm, $A_{\min} = 70$ pixels and $A_{\max} = 350$ pixels. Best performance: $t = 0.8$; $r = 6$ pixels

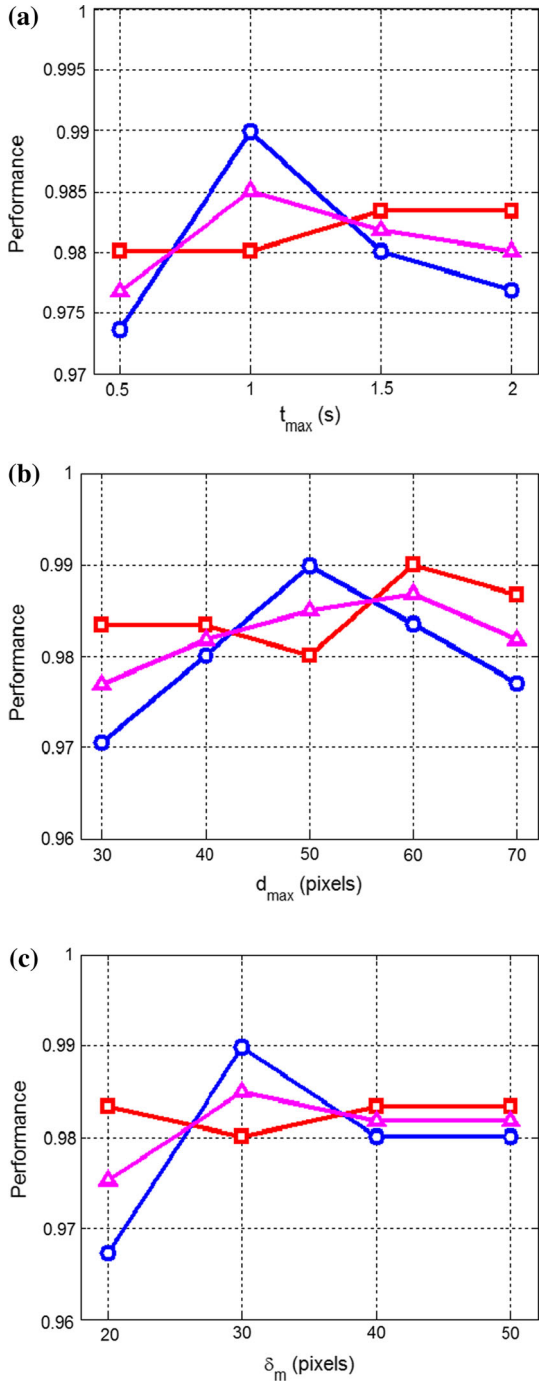

Fig. 7 Selection of parameters for tracking. Three measures of performance: precision (*blue line with circles*), recall (*red line with squares*) and F_1-score (*magenta line with triangles*), as a functions of tracking parameters. This figure is best seen in color. **a** Performance analysis for $t_{\max}$ with $d_{\max} = 60$ pixels and $\delta_m = 30$ pixels. Best performance is for $t_{\max} = 1$ s. **b** Performance analysis for $d_{\max}$ with $t_{\max} = 1$ s pixels and $\delta_m = 30$ pixels. Best performance is for $d_{\max} = 60$ pixels. **c** Performance analysis for δ_m with $t_{\max} = 1$ s and $d_{\max} = 60$ pixels. Best performance is for $\delta_m = 30$ pixels

overlapping areas only to adjacent cameras, although the deviation would increase as it is propagated from one camera to another, the relative deviation between two adjacent cameras would be the same as for the case of the two-camera system and it can be used the same parameter value to fuse tracklets, so the scalability of the system is not compromised.

Fig. 8 Illustrative examples. Examples of cases that generated false negatives: a person wearing a hat (**a**), a man carrying a big box above his head (**b**), and a woman raising her hand (**c**); false positives: a *red* container with a contour shape similar to a person's head (**d**); and true positives without false positives or negatives: a child (**e**), persons walking with their heads very close (**f**), arm in arm (**g**), arm in shoulder (**h**), a man wearing a cap (**i**), a man wearing a helmet (**j**), a man carrying a backpack (**k**), and a man carrying packages (**l**)

Interference between kinect cameras produces dots with missing depth values on the images. Maimon and Fuchs [17] found an increase up to 14.1 % of pixels without depth information values for a near worst case overlap scenario for two kinect cameras placed very close to each other. In our case, to determine a rotation matrix, we use pairs of images of the plane pattern at different orientations and compute the plane normals by selecting an area corresponding to the pattern. Pixels with missing depth values do not affect the accuracy, since they are not included into the least squares analysis, so they only reduce the amount of information available to form the overdetermined system. This could be compensated either by selecting a larger area or using more plane orientations. If a pair of images are too noisy, they can be discarded as well.

5.5 Special cases

During the counter operation, there are some special cases where the algorithm is susceptible to fail generating false detections or false counts. In the previously described evaluation of performance, we only filtered out images where no motion was detected, so these special cases are included in the evaluation. From a depth and color image dataset, we selected short sequences containing some of these cases and run the algorithm. Figure 8 shows some of these cases. They include false-negatives cases, generated by persons wearing a hat, lifting a big box above their heads and raising a hand; a false-positive case, generated by a small rounded container; and some true-positives cases, where counting was accurate, generated by children, people with their heads very close, people walking arm in arm, arm in shoulder, wearing a cap or helmet, and carrying backpacks or other objects.

Conclusion

In this paper, we presented a method for detecting and counting people using an array of Kinect sensors placed in zenithal position. We showed that the method is sound and allows to extend the narrow range of a single sensor to wider scenarios. The counting process was performed by tracking each person to obtain tracklets. Then, we solved the problem of the overlapped area between two adjacent cameras by projecting all measurements to a global frame of reference and then fusing corresponding tracklets using a distance measure. Our detector performs well compared to similar methods. Nonetheless, it may be worth to mention that the performance exhibited by state-of-the-art methods is considerably high. Our main contribution is a method that counts pedestrians from a network of zenithally placed depth cameras. For some scenarios, such as wide corridors, this configuration may provide to be a valuable solution.

Our solution propose to take advantage of a simple yet somewhat general characteristic of the human body, when observed from a zenithal position. In broad terms and for a large number of cases, the solution seems sound because it generates good levels of detection and is fast to compute. Overall, it represents a practical solution to the problem. Yet models fail when reality departs from the assumptions made. In the future, we will implement the described counter and feed a model of service in a metropolitan subway system using the information the counter provides.

Acknowledgments This work was partially supported by the FOMIX GDF-CONACYT under Grant No. 189005, IPN-SIP under Grant No. 20140325. We thank Multilink Traductores for their comments to the document and the Facultad de Ingeniería at UAQ for providing a warm environment for the development of this work. Finally, we warmly thank the reviewers for their comments, which resulted in a much better paper than the original.

References

1. Chan, A., Vasconcelos, N.: Counting people with low-level features and Bayesian regression. IEEE Trans. Image Process. **21**(4), 2160–2177 (2012)
2. Chen, K., Kamarainen, J.-K.: Learning to count with back-propagated information. In: International Conference on Pattern Recognition, pp. 4672–4677. IEEE (2014)
3. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. **34**(4), 743–761 (2012)
4. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
5. Ferryman, J., Ellis, A.-L.: Performance evaluation of crowd image analysis using the PETS2009 dataset. Pattern Recognit. Lett. **44**, 3–15 (2014)
6. Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
7. Galčík, F., Gargalík, R.: Real-time depth map based people counting. In: International Conference on Advanced Concepts for Intelligent Vision Systems, vol. 8192, p. 330. Springer (2013)
8. Gao, K.: An emergency evacuation model based on computer vision smart inducing in hotel stampede environment. In: Applied Mechanics and Materials, vol. 556, pp. 5736–5739. Trans Tech Publ (2014)
9. Golub, G., Van Loan, C.: Matrix Computations, vol. 3. JHU Press, Baltimore (2012)
10. Herrera, C., Kannala, J., Heikkilä, J.: Joint depth and color camera calibration with distortion correction. IEEE Trans. Pattern Anal. Mach. Intell. **34**(10), 2058–2064 (2012)
11. Kong, D., Gray, D., Tao, H.: A viewpoint invariant approach for crowd counting. In: International Conference on Pattern Recognition, vol. 3, pp. 1187–1190. IEEE (2006)
12. Kuhn, H.W.: The Hungarian Method for the Assignment Problem. Naval Res. Logist. Q. **2**(1–2), 83–97 (1955)
13. Lee, K., Eidson, J., Weibel, H., Mohl, D.: IEEE 1588-standard for a precision clock synchronization protocol for networked measurement and control systems. In: Conference on IEEE, vol. 1588, p. 2 (2005)
14. Lemkens, W., Kaur, P., Buys, K., Slaets, P., Tuytelaars, T., De Schutter, J.: Multi RGB-D camera setup for generating large 3D point clouds. In: International Conference on Intelligent Robots and Systems, pp. 1092–1099. IEEE (2013)
15. Macknojia, R., Chávez-Aragón, A., Payeur, P., Laganiere, R.: Calibration of a network of kinect sensors for robotic inspection over a large workspace. In: IEEE Workshop on Robot Vision, pp. 184–190. IEEE (2013)
16. Maddalena, L., Petrosino, A., Russo, F.: People counting by learning their appearance in a multi-view camera environment. Pattern Recognit. Lett. **36**, 125–134 (2014)
17. Maimone, A., Fuchs, H.: Reducing interference between multiple structured light depth sensors using motion. In: IEEE Virtual Reality, pp. 51–54. IEEE (2012)
18. Martin Martin, R., Lorbach, M., Brock, O.: Deterioration of depth measurements due to interference of multiple RGB-D sensors. In: International Conference on Intelligent Robots and Systems, pp. 4205–4212. IEEE (2014)
19. Mikhelson, I.V., Lee, P.G., Sahakian, A.V., Wu, Y., Katsaggelos, A.K.: Automatic, fast, online calibration between depth and color cameras. J. Vis. Commun. Image Rep. **25**(1), 218–226 (2014)
20. Najman, L., Schmitt, M.: Geodesic saliency of watershed contours and hierarchical segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **18**(12), 1163–1173 (1996)

21. Nam, W., Dollár, P., Han, J.H.: Local decorrelation for improved pedestrian detection. In: Advances in Neural Information Processing Systems, pp. 424–432 (2014)
22. Porzycki, J., Lubaś, R., Mycek, M., Wkas, J.: Dynamic data–driven simulation of pedestrian movement with automatic validation. In: Traffic and Granular Flow, pp. 129–136. Springer (2015)
23. Rauter, M.: Reliable human detection and tracking in top-view depth images. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 529–534 (2013)
24. Ryan, D., Denman, S., Fookes, C., Sridharan, S.: Crowd counting using multiple local features. In: Digital Image Computing: Techniques and Applications, pp. 81–88. IEEE (2009)
25. Ryan, D., Denman, S., Sridharan, S., Fookes, C.: An evaluation of crowd counting methods, features and regression models. Comput. Vis. Image Underst. **130**, 1–17 (2015)
26. Spinello, L., Arras, K.: People detection in RGB-D data. In: IEEE International Conference on Intelligent Robots and Systems, pp. 3838–3843 (2011)
27. Wang, Y., Lian, H., Chen, P., Lu, Z.: Counting people with support vector regression. In: International Conference on Natural Computation, pp. 139–143. IEEE (2014)
28. Yan-Yan, C., Ning, C., Yu-Yang, Z., Ke-Han, W., Wei-Wei, Z.: Pedestrian detection and tracking for counting applications in metro station. Discrete Dyn. Nat. Soc. **2014**, 1–11 (2014)
29. Yu, S., Wu, S., Wang, L.: SLTP: a fast descriptor for people detection in depth images. In: IEEE International Conference on Advanced Video and Signal-Based Surveillance, pp. 43–47 (2012)
30. Zhang, C., Zhang, Z.: Calibration between depth and color sensors for commodity depth cameras. In: Computer Vision and Machine Learning with RGB-D Sensors, pp. 47–64. Springer (2014)
31. Zhang, X., Yan, J., Feng, S., Lei, Z., Yi, D., Li, S.Z.: Water filling: unsupervised people counting via vertical kinect sensor. In: International Conference on Advanced Video and Signal-Based Surveillance, pp. 215–220. IEEE (2012)
32. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000)
33. Zhu, L., Wong, K.-H.: Human tracking and counting using the kinect range sensor based on adaboost and kalman filter. In: Advances in Visual Computing, pp. 582–591. Springer (2013)
34. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: International Conference on Pattern Recognition, vol. 2, pp. 28–31. IEEE (2004)

Pablo Vera received a BS. in Communications and Electronics and an MSc. degree in Advanced Technology in 2007, all of them from Instituto Politécnico Nacional (IPN), Mexico. Currently, he is working at IPN, as an associate professor in the area of Image Analysis. His areas of research interest include computer vision and pattern recognition.

Sergio Monjaraz graduated from *Instituto Tecnológico de Morelia* as a computer systems engineer, specializing in security and information management in 2015. Currently is a Master student in the Instituto Politécnico Nacional (IPN), Mexico, in the area of image analysis.

Joaquín Salas is a professor in the area of Computer Vision at Instituto Politécnico Nacional. His research interests include the development of assistive technology for the visually impaired and visual interpretation of human activity. Salas received a PhD in computer science from Monterrey Institute of Technology and Higher Studies (ITESM), Mexico. Contact him at jsalasr@ipn.mx.