Pattern Recognition Letters 81 (2016) 41–50

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Counting people by RGB or depth overhead cameras☆

Luca Del Pizzo, Pasquale Foggia, Antonio Greco, Gennaro Percannella*, Mario Vento

University of Salerno, Via Giovanni Paolo II 132, Fisciano, SA I-84084, Italy

ARTICLE INFO

Article history:
Available online 9 June 2016

Keywords:
People counting
Overhead camera
Video analytics
Retail analytics

ABSTRACT

In this paper we present a vision based method for counting the number of persons which cross a virtual line. The method analyzes the video stream acquired by a camera mounted in a zenithal position with respect to the counting line, allowing to determine the number of persons that cross the virtual line and providing the crossing direction for each person. The proposed approach has been specifically designed to achieve high accuracy and computational efficiency, so as to allow its adoption in real scenarios. An extensive evaluation of the method has been carried out taking into account the main factors that may impact on the counting performance and, in particular, the acquisition technology (traditional RGB camera and depth sensor), the installation scenario (indoor and outdoor), the density of the people flow (isolated people and groups of persons), the acquisition frame rate, and the image resolution. We have also analyzed the combination of the outputs obtained from the RGB and depth sensors as a way to improve the counting performance. The experimental results confirm the effectiveness of the proposed method, especially when combining RGB and depth information, and the tests over three different CPU architectures demonstrate the possibility of deploying the method both on high-end servers for processing in parallel a large number of video streams and on low power CPUs as those embedded on commercial smart cameras.

1. Introduction

Systems able to count the number of persons that cross a virtual line provide a valuable information in different real world scenarios. People counting systems are installed at the entrances or at the exits of closed areas and they allow to obtain information regarding the people flow through these points. Some notable locations, where people counting systems are usually installed, are represented by shops, malls, museums, fairs, metros, trains, buses, etc. In all such places, these systems are extremely important for both safety and business intelligence purposes: in the former case, they permit to check in real time that the number of persons present in a premise is not approaching or exceeding the maximum level; in the latter case, they provide statistical information regarding the flow of persons during time. In the last few years we assisted to a growing interest toward the use of people counting systems that adopt computer vision algorithms which process the video stream acquired by cameras mounted in a zenithal position. The retail domain represents one of the most typical scenarios where such systems are used. Information regarding the number of persons that are present in a shop or in a mall is very relevant for several purposes: it allows to better allocate the staff during the day and the week, and the peak hours, to correlate the affluence of customers with the marketing campaigns or with the selling performance of similar shops in the same area, and so on.

Related work: In the literature, several methods for people counting using zenithal cameras have been presented. The main trends that have inspired the researchers in the design of the people counting methods share a common architecture that involves the following steps: (i) performing foreground detection to separate moving pixels from the static part of the image; (ii) detecting each single person in the scene from the foreground mask; (iii) tracking each detected person using the trajectory so as to update the count when the trajectory intersects the virtual line. The main differences among the various proposals lie in the approach used for solving the problems in each of the three previous steps and in the type of the adopted sensor (traditional RGB camera or depth sensor).

The first approaches for people counting that appeared in the literature are designed to process the video stream produced by traditional RGB cameras [1–5]. Most of such methods adopt a combination of a foreground detection method (frame differencing as [1,2] or background subtraction as [3,4]) and tracking techniques (overlap tracking as [1,3,4] or optical flow as [2]), while they differentiate for the way they refine the foreground detection

☆ This paper has been recommended for acceptance by Cosimo Distante.

* Corresponding author. Tel.: +39 089 96 4253; fax: +39 089 96 4218.
E-mail address: pergen@unisa.it (G. Percannella).

http://dx.doi.org/10.1016/j.patrec.2016.05.033
0167-8655/

step with more or less sophisticated people detection phases. In [1–3] tracking is applied on the blob obtained after the application of connected component labeling on the foreground mask. Unfortunately, this approach tends to be error prone in case of passages of more than one person per time. Researchers, over the years, have therefore devised more complex algorithms to strengthen the people detection phase. In [4], a k-means based people segmentation algorithm is adopted for reducing errors of the people detection process. However, this technique allows to improve performance in terms of accuracy, but it solves only partially the issue in crowded scenes. More recently, Mukherjee et al. [5] proposed a head detection method that carries out the edge detection of blobs and the search of the circles through the Hough transform. The people counting operation is then performed by tracking the heads through optical flow. The main limitations of this algorithm are the errors in the circles search due to camouflage and shadows.

The methods described above have major issues in crowded scenes due to changes in lighting, shadows, compound objects. To limit the impact of these problems, in the scientific literature approaches have been proposed that carry out 3D analysis of the scene, making use of the depth information. By exploiting the third dimension it is possible to face the people counting issue in crowded scenes through detection and tracking of the head of the persons. The first approaches in this field were based on the use of stereo cameras as in [6–8], although after the advent of the Microsoft Kinect that provides depth image through structured light, researchers have proposed new approaches that use this type of acquisition device. In particular, Zhang et al. [9] define a method that detects the heads by searching for local minima within the depth map using watershed. Vera et al. [10] propose a method that uses a support vector machine (SVM) classifier to detect people, applying the histograms of oriented gradients descriptor. This method has the disadvantage of being inefficient and dependent on the scenario, due to the use of a classifier. Fu et al. [11] propose a method that replaces the values in the grayscale depth map with color gradients in order to simplify the next step of filtering of the person on the basis of their height from the ground.

Contributions: In this paper we propose a people counting method with the aim of achieving high computational efficiency and high counting accuracy. The proposed method, already presented in a preliminar version in [12], performs only the foreground detection phase, and differently from the common approach usually adopted in the literature it does not carry out the people/head detection and the object tracking steps. Our method is based on a counting sensor specifically designed for the problem at hand, that is able to guarantee a high immunity to the foreground detection errors and to overcome the majority of the difficulties which arise in the application of a person tracking method under crowded conditions.

The main contributions provided with this paper, and in particular with respect to [12], are the following.

Pre-processing stage: we study the impact of the introduction of a noise-removal pre-processing stage on the performance of the original method presented in [12].

Additional technical details: we include an extended description of the method with particular reference to the foreground detection phase, to the newly provided algorithmic formulation that clarifies significantly the overall method and simplifies reproducibility of the implementation, to the procedure to follow for the optimal selection of the algorithm parameters, to the analysis of the computational requirements of the method.

Combination of sensors: we propose a combination of the outputs of the method when both a traditional optical camera and a depth sensing device are used; our results show that the combination improves the performance with respect to each kind of sensor used alone, attaining a more balanced trade-off between false positives and false negatives.

Experimental validation: we study how the resolution of the image and the acquisition frame rate impact on the counting performance of the method. Moreover, we consider an additional dataset acquired in an unconstrained real scenario in order to assess performance in realistic conditions; we also compare the method with respect to both the original version of the approach without the pre-processing stage and a method that exploits an object tracking strategy.

Computational analysis: we reserve a large attention in this paper to the computational issues; the method is tested on three different CPU architectures characterized by increasing computational power (from an embedded single board computer to a high-end workstation) with the aim of verifying the adequacy of the method for field installations where real time analysis is required.

The organization of the paper is the following: in Section 2 we describe the proposed method; in Section 3 we detail the dataset used for the experimental validation; in Section 4 we report and comment the results of the tests; finally, in Section 5 we draw final conclusions and delineate future work.

2. Proposed method

In this section we describe the proposed people counting method, introducing the rationale and providing the technical details. We also focus on the issues related to the computational complexity and describe some optimizations we carried out in order to allow the algorithm to run on both general purpose CPU platforms and on embedded systems (as those typically employed on surveillance cameras).

An object that crosses an area monitored by a generic sensor (an optical camera, a 3D imaging device, a time of flight or a passive infrared sensor, just to name a few) installed in a zenithal position with respect to the area determines a perturbation of the signal generated by the sensor with respect to the steady state when no objects are into the area. In the ideal case, for the problem of the people counting, the variations of the signal with respect to the steady state should be significant only in presence of persons that cross the area. The processing and the interpretation of the signal produced by the sensor should allow to count the exact number of persons crossing the virtual line traced in the monitored area, discriminating also the crossing direction. Given the constraints on the computational power imposed by the need of running the algorithm on embedded systems, we have excluded the approach based on people detection (e.g. recognition of the shape of the head and/or of the shoulders), since they are quite expensive. For the same reason, we have excluded methods based on the optical flow. Moreover, while the use of tracking algorithms on foreground blobs can be reasonably fast, it can have some problems when foreground detection artifacts induce splits or merges of the blobs; also, tracking may become unreliable when the objects are at the boundaries of the image, and this is a condition that may happen quite frequently when the scene is acquired by a zenithal camera that is positioned at a reduced height (e.g. on the ceiling of an aisle). So, in order to have a robust method even in these situations, we have chosen to avoid tracking; and since connected component detection is a costly processing that is mainly needed for tracking, we can also avoid this step.

From these observations it derives that the architecture of a generic people counting method using an overhead sensor might comprise a first foreground detection phase aimed at detecting the objects into the area and a second phase that exploits foreground information for the counting. The people counting method proposed in this paper adopts a similar system architecture composed by two stages (see Fig. 1): (i) the foreground detection stage that

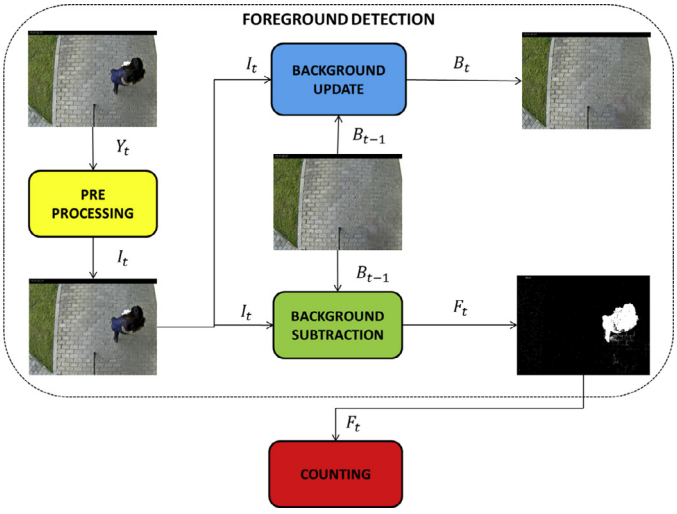

Fig. 1. System architecture of the proposed method.

allows to identify the pixels in motion within the scene by adopting a background subtraction methodology; (ii) the counting sensor that interprets the results of the foreground detection step and provides the final output of the counting.

2.1. Foreground detection

Foreground detection from optical cameras is a widely discussed problem in the scientific literature and plenty of papers have been published on this topic in the last 15 years, regarding both methodologies for detecting changes into the scene as in [13–17], and algorithms for removing artifacts as shadows [18,19] or reflections [20].

The accuracy of the output provided by the foreground detection stage is a key issue for achieving high counting performance, as errors done at this stage may impact significantly on the successive counting stage. The largest source of errors arising in typical people counting scenarios is represented by the changes of the illumination, especially when the system is installed outdoor. In order to cope with this problem we adopt a method that obtains the foreground per difference from the background that is dynamically updated. Furthermore, we also introduce image processing procedures for reducing the noise in the image. Thus the foreground detection stage used in the proposed approach involves the following steps: pre-processing, background subtraction and background update.

2.1.1. Pre-processing

The images captured by the cameras in indoor and outdoor environments are affected by noise, which can be caused by vibrations, heat or interference. In order to reduce such noise we perform a Gaussian smoothing, which consists of the convolution of the image with kernel of Gaussian values. We have experimented different sizes of the kernel, and noticed that a 3×3 kernel constitutes the best choice when operating on videos at 640×480 resolution.

2.1.2. Background subtraction

The foreground is obtained per difference from the background as in [21]. The background image is used as a reference to compute, for each pixel, the absolute color distance between the current frame and the background model: $D_t(x,y) = |I_t(x,y) - B_{t-1}(x,y)|$. Thus, the foreground mask F_t is obtained as follows:

$$F_t(x,y) = \begin{cases} 1 & if\ D_t(x,y) \geq \tau_F \\ 0 & otherwise \end{cases} \quad (1)$$

The method above can be applied without modifications also to the gray level image provided by a depth sensor. It is interesting to note that in the latter case, it is possible to choose the threshold τ_F so as to ensure that only the pixels at a certain height from the ground are classified as foreground pixels.

2.1.3. Background update

The background model is updated through an exponential smoothing (or equivalently a first-order infinite impulse response, IIR filter), according to which the background image B_t at time t is obtained by blending the current frame I_t with the background image B_{t-1} calculated at time $t-1$, using the smoothing factor α_B:

$$B_t(x,y) = \alpha_B \cdot I_t(x,y) + (1-\alpha_B) \cdot B_{t-1}(x,y) \quad (2)$$

The weight α_B allows to control the time interval required by a static foreground object to become part of the background image, as it is related to the time constant τ:

$$\alpha_B = 1 - e^{\frac{-\Delta T}{\tau}} \quad (3)$$

where ΔT is the sampling time interval of the discrete time implementation that can be approximated to $\alpha \approx \frac{\Delta T}{\tau}$, when $\Delta T << \tau$ (in our experimentations we set $\tau \approx 120$ s while ΔT, being the interframe period, is typically of the order of 40–100 ms).

It has to be noted that although it would be possible to selectively update the background using a different smoothing factors α for the pixels that at time t belongs to foreground or to background, in the proposed method we do not adopt such approach for computational reasons, as the choice of adopting separated smoothing factors for the foreground and the background would hinder the possibility of using SIMD instructions for the background update phase (as explained in Section 2.3), thus with a very negative impact on the execution time.

2.2. Counting

Many of the people counting algorithms presented in the literature (see Section 1) use some form of tracking. Actually, a tracking algorithm provides far more information than what is needed to perform the counting: the tracking gives the detailed trajectories of the objects, while what is needed is only an indication of the direction of crossing the counting region. However, this detail is paid with a reduction of the robustness of the algorithm, since the tracking results can be affected by problems in the foreground detection (such as the splitting of the foreground region associated to a single object) that make the algorithm lose the continuity of the tracks. Furthermore, many tracking algorithms work reliably only if the objects to be tracked are completely contained within the image. However, this might be difficult to achieve if the camera is positioned indoor and the ceiling is not very high (of course a fish-eye camera could be used, but then the distortion would entail other kinds of problems).

For this reason, we have chosen a simpler, but more robust approach: the foreground mask obtained from the foreground detection phase is directly fed to a “virtual sensor”, which is based on integral operations that are both robust with respect to errors in the foreground mask, and very efficient to compute. Also, for estimating the motion direction of the foreground objects, we use a technique that is similar to the working of incremental rotary encoders, see [22]: while it yields less information than what can be obtained from a tracker (for instance, there is no speed indication;

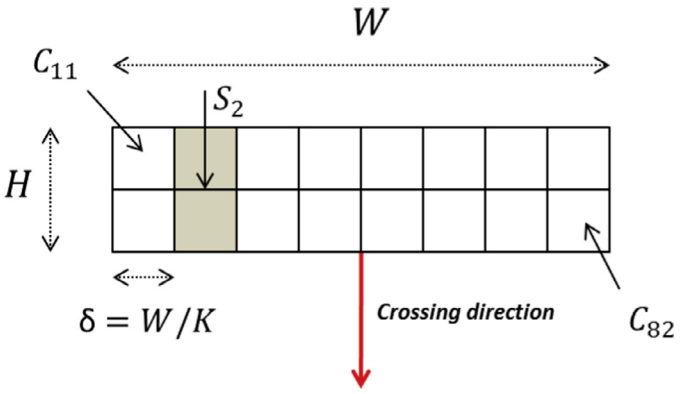

Fig. 2. Architecture of the proposed counting sensor.

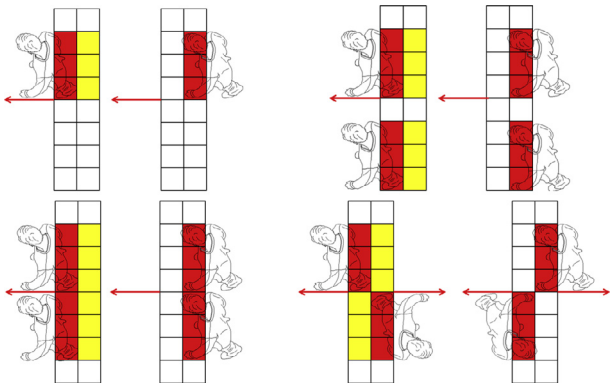

Fig. 3. Examples of the behavior of the sensor with one or more crossings. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

it can only sense if motion is happening along a predefined direction), it is robust and efficient.

The proposed sensor is geometrically characterized by a rectangular area, with a crossing direction associated to it, as shown in Fig. 2.

We define the width of the sensor W, as the side of the rectangle perpendicular to the crossing direction and the height of the sensor H, as the side parallel to the crossing direction. The rectangle is divided widthwise into K stripes with a width $\delta = W/K$. Each stripe is divided heightwise in two cells with a height $H/2$. We denote with S_i $(i = 1, \ldots, K)$ the generic stripe of the sensor and with C_{ij} $(j = 1, 2)$ the two cells belonging to the stripe S_i. At the time t a cell C_{ij} is active if the number of foreground pixels within the cell is higher than the total number of pixels in the cell $Area(C_{ij})$ multiplied by a threshold $\theta_C \in [0, 1]$.

$$A_{ij} = \begin{cases} 1, & \text{if } F_t(C_{ij}) \geq \theta_C \cdot Area(C_{ij}) \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

The algorithm stores the current activation value A_{ij} and the previous activation value A'_{ij} for each cell, while for each stripe it maintains an activation value B_i, initialized to 0. The value of B_i changes from 0 to 1 when the following condition holds:

$$B_i \leftarrow 1 \text{ if } A_{i2} = 1 \wedge A'_{i1} = 1 \wedge A'_{i2} = 0 \tag{5}$$

The evaluation of the activation sequence is used to detect separately, with a good reliability, people walking with carts or in a queue in both directions. This single condition is not sufficient to properly solve the problem of counting people walking nearby in the same direction. For this reason, the algorithm searches contiguous sequences of cells that have an activation value equal to 1. In other words, it find all the pairs of indices p, q, with $p \leq q$, such that $B_p = B_{p+1} = \cdots = B_q = 1$ and $B_{p-1} = B_{q+1} = 0$ (managing borderline cases with the assumption $B_0 = B_{K+1} = 0$).

```
IF A_{p,1} = A_{p+1,1} = ... = A_{q,1} = 0 THEN
    B_i ← 0 FOR i = p, ..., q
    IF q − p + 1 ≥ θ_K THEN
        L = q − p + 1
        Increase counting by floor(L/θ_K)
    END IF
END IF
```

The algorithm disables a stripe only when also its adjacent are deactivated and counts a number of persons which corresponds, for each group of stripes with a width $L = (q - p + 1)$ greater than a threshold θ_K, to the floor of the ratio L/θ_K. Thus, the algorithm is able to solve also the problem of counting more persons crossing the gate very close together.

The setup of the system requires to define the parameters W, H, θ_C, θ_K, while the remaining parameters K and δ can be straighforwardly derived from them. In particular, W, being the width of the gate, is an external parameter determined by the location where the system is installed. H has to be configured by looking at the best trade-off between two opposite requirements. On one side the higher is the value of H the lower is the requested frame rate of the video, that is an important requirement when the processing is done on embedded low-end devices or when there is the need of processing as much as possible streams in parallel on a single server. On the other side the lower H, the higher is the ability of the sensor to correctly detect the passage of two persons walking close to each other in a row. We noticed that setting H as twice the depth of an average sized person represents a good compromise between the above two opposite requirements. Finally, in the experiments described in the following Section, the values of $\theta_C = 0.2$ and $\theta_K = 3$ were determined as those able to guarantee the highest performance on a sequence of frames not included in the test dataset.

Fig. 3 shows four examples of the behavior of the sensor under typical situations with one or more persons passing through the virtual line. In the example shown in the figure, the counting sensor is composed of 8 horizontal stripes, while the minimum number of consecutive stripes for the activation of the sensor is set to $\theta_K = 3$. In each row we show typical cases of crossings over the counting sensor. Each example is composed by a pair of images; the right image shows the activation of the sensor at time $t - 1$, while the left image the activation at time t. The cells active at time $t - 1$ are yellow, while the cells active at time t are red. In the first figure, one person activates three adjacent stripes, while, in the second figure, two persons activate two different groups of three adjacent stripes. In the third figure, the two persons walking together activate only one group of six consecutive stripes. In the fourth figure, the two persons walking in opposite directions activate at time $t - 1$ two groups of three consecutive cells in two different sides of the sensor; at time t the sensor evaluates the activation sequence of the stripes and is able to detect the crossing direction of the person. In all these cases the proposed method is able to properly count the number of people crossing the sensor.

2.3. Computational issues

A crucial issue that needs to be taken into account during the design and the implementation phases of the method is the computational efficiency so as to allow the algorithm to run in real time on embedded systems and on devices with low computational power. In order to achieve the result of real time processing, two main types of optimizations are considered here. The first type of optimization is inspired to that described in [23], according to

which the method does not perform floating point operations and largely uses SIMD instructions, thus exploiting the parallelism of the vector processors so as to simultaneously process more data. The second type of optimization starts from the results in [23], showing that the background update phase is a rather costly procedure when compared against the remaining phases of the foreground detection stage. Thus, in order to reduce the computational load of the background update phase, we perform this phase at frequency that is 10% of the frequency at which the method operates, as we have experimentally verified that this does not introduce a significant performance degradation while it allows to dramatically reduce the computational burden associated to such phase.

3. Dataset description

The experimental validation of the method has been carried out on two datasets that we have specifically collected for this aim (the dataset is made publicly available and can be downloaded at http://mivia.unisa.it/datasets-request). The adopted datasets respond to two different needs which arise in the experimentation of a pattern recognition system. On one side, we intend to analyze the behavior of the proposed method under a controlled environment in order to determine its response to the variations of the main variables that may have a potential impact on the performance. To this aim, while building the dataset we took into account the acquisition technology (optical camera or depth camera), the source of the illumination (indoor and outdoor) and the flow density (increasing number of passages of persons in one or both directions). Furthermore, we are also interested to analyze the impact on the performance of the image resolution and of the frame rate of the videos, as in the typical application on the field we are interested to achieve optimal performance with minimum computational load on the CPU. As a matter of fact, we are interested to process more streams in parallel on the same CPU, or conversely, to use cheaper systems for processing the image sequences.

All the variables mentioned above were specifically modeled for the construction of the first of the two used datasets. Hereinafter, we will denote this dataset with $Dataset_C$, where the subscript C stands for the term *Controlled*, meaning that the dataset was acquired in a controlled environment where each person who crossed the counting line had preliminarily received specific instructions regarding how to transit. On the other side, in order to assess performance of the method in a real scenario, we collected a second dataset where persons freely flow across the virtual line. We will refer to the second dataset as $Dataset_U$, where subscript U stands for *Uncontrolled*. In the following we provide a general description of both datasets.

3.1. Description of $Dataset_C$

The $Dataset_C$ has been firstly presented in [12]. For the sake of completeness, here we report the relevant elements of the dataset. The dataset comprises video sequences captured by two imaging devices at the same time. Specifically we used a traditional optical camera operating in the visible range and acquiring RGB images; the other adopted acquisition device is the depth imaging system of a Microsoft Kinect sensor that provides a gray level image where the intensity of the pixel is linearly related to the distance from the camera of that part of the framed object which the pixel belongs to. As in [12], for the sake of brevity hereinafter we will refer to the optical camera and the depth Kinect sensor as *RGB* and *DEPTH*, respectively.

The two devices were mounted on top of a crane, very close to each other in order to maximize the overlap of the framed areas. The cameras are located at a height of 3.2 m from the ground in a zenithal position so as to frame the persons from overhead. Videos from both devices were originally captured at 30 frames per second with a resolution of 640 × 480 pixels. Then, in order to study the dependence of performance on the image resolution and the frame rate, from the original version we derived new sequences scaled in resolution (320 × 240 and 160 × 120) at 30 frames per second and in frames per second (25, 20, 15, 10 and 5) while maintaining the initial 640 × 480 resolution.

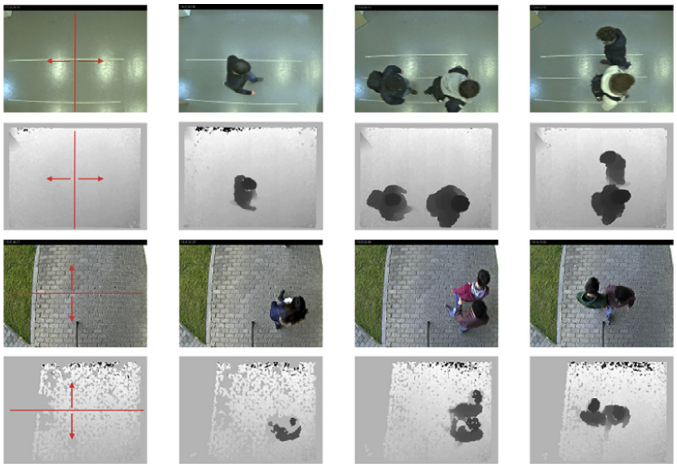

Fig. 4. Example images extracted from the $Dataset_C$: images in the first two rows refer to the INDOOR scenario, while the remaining images are from the OUTDOOR scenario, alternating RGB and then DEPTH; images in the leftmost column show the virtual crossing line, while in the remaining columns from left to right there are examples of images with increasing number of persons.

The dataset contains sequences acquired in two different scenarios that account for the situations when a natural illumination is prevalent (we name it *OUTDOOR*) or when the source of illumination is exclusively artificial (*INDOOR*) as shown in Fig. 4. It is important to note that the two scenarios above were specifically devised to resemble typical real world environments where people counting systems are adopted, paying attention to the characteristics of the flooring, too. In fact, the INDOOR scenario accounts for an installation inside a closed area (a shop, a bank, a shopping mall, etc.) where the aim is to count the number of entering/exiting persons; in such cases the illumination is assured by a distributed artificial lighting system with a relevant reflection from the flooring. On the contrary, the OUTDOOR scenario accounts for a typical installation in front of the entrance of a premise where the people counting system is adopted with the aim of getting statistics about the people flows on that area; these situations are generally characterized by the presence of a matte pavement (thus, with no reflections) and indirect but prevalent sunlighting.

$Dataset_C$ comprises sequences with an increasing number of persons that flow within the area of interest in the same direction and/or in the opposite directions. Such sequences allow to evaluate the people counting performance of the analyzed methods under different crowding conditions. In the simplest case, there is a single person that crosses the area framed by the camera, while in the most two complex cases there is a group of six persons which cross the area proceeding either in the same direction or in the two opposite directions (three persons in a direction and the remaining three in the other one).

In Table 1, information regarding the number of samples of the $Dataset_C$ are shown. Data are reported separately with respect to the scenario and the density of the flow distinguishing the cases of the transits of isolated individuals and of the groups of persons. In Fig. 4 example images extracted from the $Dataset_C$ are shown.

Table 1
Number of samples of the $Dataset_C$ used for the benchmarking in a controlled environment, expressed in terms of the persons which cross the virtual line.

Scenario	Flow density	Number of transits
INDOOR	Isolated transits	212
	Group of persons	203
OUTDOOR	Isolated transits	236
	Group of persons	317

3.2. Description of $Dataset_U$

The $Dataset_U$ has been recorded at the entrance of a corridor in uncontrolled way, thus allowing the free flow of the persons under the camera in both directions. This allowed to collect several complex situations which are not present in the first dataset and may typically arise in real world scenarios, where people may cross the virtual line in unconventional ways. Some situations which are present in the $Dataset_U$ are the following: one or more persons standing in the area of interest or waving over the virtual line, persons that diagonally cross the virtual line, persons that suddenly stop and re-start in proximity of the virtual line or change their travelling direction, persons crossing with medium/large-sized objects (e.g. cart, stepladder) or with open arms, very dense groups of persons passing through the passage.

The $Dataset_U$ has been obtained in an INDOOR scenario and contains 264 isolated transits and 673 passages from persons in groups.

4. Performance indices and experimental results

In this section, we report the results of the performance assessment of the proposed people counting method on the benchmark datasets previously described. We describe the figures of merit used for quantifying the performance of the method and the adopted experimental protocol; then we report and comment the results obtained by the method over the two adopted datasets; we analyze the performance achieved when combining the output of the DEPTH and the RGB sensor; we also compare the method with the original version proposed in [12], which does not use the pre-processing stage, and with respect to an approach based on the adoption of the object tracking strategy; finally we study the computational requirements of the method over different processing platforms.

4.1. Experimental protocol

We use $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$ and $f\text{-}index = \frac{2 \cdot Prec \cdot Recall}{Prec+Recall}$ as figures of merit for measuring performance, with TP the number of true positives, i.e. transits of persons that are correctly detected by the method, FP the number of false positives, i.e. falsely detected passages of persons, and FN the number of false negatives, i.e. passages of persons missed by the method. In order to account for the possible delays that may occur between the time instants of each passage across the virtual line as reported into the ground truth and as detected by the method, in the computation of the aforementioned performance indices we declare a passage as correctly detected if such delay is below a threshold that in our tests has been fixed to 1 s.

Moreover, it has to be noted that the tuning of the parameters of all the considered methods analyzed in this section was carried out on separate sequences not used for testing.

4.2. Analysis of the results

Performance of the proposed method has been assessed separately on the $Dataset_C$ and the $Dataset_U$; in the first case, we are interested to characterize the impact of the crowding level (isolated transit vs group of persons), of the scenario (INDOOR vs OUTDOOR) and of the used sensor (DEPTH vs RGB) on the overall counting performance; in the second case, we analyze performance in an unconstrained scenario when people flow freely so as to resemble a real situation. Furthermore, we also report the performance that can be achieved by combining the output of the method using the RGB and the DEPTH sensors.

As for the analysis on the first dataset, we report the disaggregated results in Table 2 which allow the reader to find the result for any combination of sensor, scenario and crowding level. In this table data are also reported when the video stream is processed at different resolutions. Moreover, in order to study the performance of the approach only with respect to a single dimension, flow density and scenario, respectively, in the Tables 3 and 4 data aggregated over the other dimension.

4.2.1. Analysis of results on disaggregated data

In Table 2, the performance achieved by the proposed method (including the pre-processing phase) on the $Dataset_C$, are expressed in terms of the figures of merit previously defined, and are calculated at the original 640 × 480 resolution and at half (320 × 240) and a quarter (160 × 120) rescaled resolutions. As a first observation, we note a performance degradation when the resolution is reduced. When the images are rescaled at half resolution, the *f-index*

Table 2
Performance of the proposed method on the $Dataset_C$ at three different resolutions: the original capture resolution of 640 × 480 and the scaled ones, 320 × 240 and 160 × 120. The frame rate of all the considered sequences is fixed to 30 fps.

			640 × 480			320 × 240			160 × 120		
Sensor	Scenario	Flow density	Re	Pr	*f*	Re	Pr	*f*	Re	Pr	*f*
RGB	INDOOR	Isolated transits	0.967	0.990	0.979	0.934	1.000	0.966	0.849	1.000	0.918
		Group of persons	0.803	0.982	0.883	0.685	1.000	0.813	0.665	1.000	0.799
	OUTDOOR	Isolated transits	1.000	1.000	1.000	0.992	1.000	0.996	0.919	1.000	0.958
		Group of persons	0.868	0.986	0.923	0.852	0.975	0.909	0.864	0.986	0.921
DEPTH	INDOOR	Isolated transits	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
		Group of persons	0.975	1.000	0.988	0.961	1.000	0.980	0.956	1.000	0.977
	OUTDOOR	Isolated transits	0.962	1.000	0.981	0.911	1.000	0.953	0.754	1.000	0.860
		Group of persons	0.940	1.000	0.969	0.924	1.000	0.961	0.858	0.996	0.922
RGB∨DEPTH	INDOOR	Isolated transits	1.000	0.991	0.995	1.000	1.000	1.000	1.000	1.000	1.000
		Group of persons	0.975	0.985	0.980	0.970	1.000	0.985	0.956	1.000	0.977
	OUTDOOR	Isolated transits	1.000	1.000	1.000	1.000	1.000	1.000	0.975	1.000	0.987
		Group of persons	0.984	0.994	0.989	0.968	0.978	0.973	0.953	0.984	0.968

Table 3
Performance of the people counting method with different flow densities when using the RGB or the DEPTH imaging sensors on the $Dataset_C$. Image resolution is 640 × 480, while frame rate is 30 fps.

Sensor	Flow density	Recall	Precision	*f*-index
RGB	Isolated transits	0.984	0.995	0.990
	Group of persons	0.842	0.984	0.908
DEPTH	Isolated transits	0.980	1.000	0.990
	Group of persons	0.954	1.000	0.976
RGB∨DEPTH	Isolated transits	1.000	0.995	0.998
	Group of persons	0.980	0.989	0.985

Table 4
Performance of the people counting method in the INDOOR and in the OUTDOOR scenarios when using the RGB or the DEPTH imaging sensor with reference to $Dataset_C$. Image resolution is 640 × 480, while frame rate is 30 fps.

Sensor	Scenario	Recall	Precision	*f*-index
RGB	INDOOR	0.887	0.987	0.934
	OUTDOOR	0.924	0.992	0.957
DEPTH	INDOOR	0.988	1.000	0.994
	OUTDOOR	0.949	1.000	0.974
RGB∨DEPTH	INDOOR	0.988	0.988	0.988
	OUTDOOR	0.992	0.997	0.994

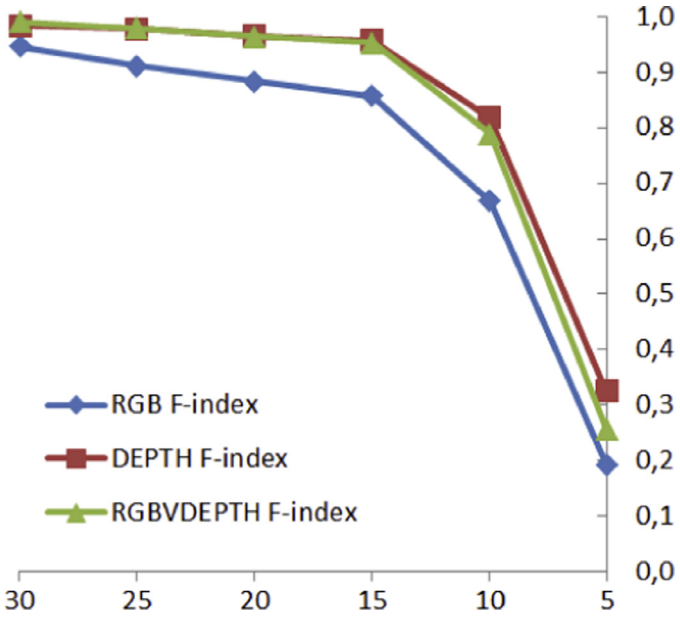

Fig. 5. Performance of the method evaluated on the $Dataset_C$ for different values of the acquisition frame rate.

decreases of about 2% for the RGB images, and 1% for the DEPTH images, while in case of 160 × 120 resolution the performance loss is about 4–5% independently of the sensor. The performance in Table 2 confirms the main findings in [12]. A higher counting accuracy is achieved when exploiting the depth information provided by the Kinect sensor in comparison to the adoption of a traditional camera. Depth information allows to detect more accurately than the RGB camera the foreground, being insensitive to problems related to illumination changes, object camouflage with the background, artifacts due to cast shadows or reflections. Furthermore, we also note that the main limitation of the algorithm is represented by the high false negative rate especially in the case of the adoption of the RGB sensor when one out of ten passages are missed. On the contrary, the false positive rate is zero when the DEPTH sensor is used, and allows to achieve 0.990 for the Precision index on the RGB camera.

We also study the performance achieved by properly combining the outputs using the two sensors. The combination is done using the *max* rule, i.e. for each passage, the output of the combination (RGB∨DEPTH) is given by the maximum counting value proposed by each sensor. In fact, since the proposed method tends to have in most configurations a higher *Precision* than a *Recall* independently of the type of used sensor, the proposed combination of the two sensors is effective in improving the *Recall*, while preserving a high value of the *Precision*, as confirmed by the results in the last three rows of Table 2.

4.2.2. Analysis of results with respect to the flow density

The Table 3 highlights performance under different conditions of the flow density (isolated transits or group of persons). In case of RGB, crowding level significantly affects accuracy causing a decrease of the *f-index* of 0.082 from the isolated transits case to the group of persons case. Differently, the performance achieved by the considered method when using the DEPTH sensor appear very stable as in this case the variation is only 0.014. As it can be noted, the combination of the two sensors attains a higher performance with respect to each single sensor, and, perhaps more important, also yields a more balanced trade-off between *Precision* and *Recall*.

4.2.3. Analysis of results with respect to the scenario

Table 4 provides a view of the results aggregated over the type of scenario (INDOOR or OUTDOOR). We note that the method, when applied to the images provided by the RGB sensor, behaves slightly better in the OUTDOOR scenario, although the difference of the *f-index* in the two cases is only 0.023. A similar variation can be noticed with the DEPTH sensor, but in this case the highest performance is achieved in the INDOOR scenario. The latter result is motivated by the fact that the sunlight affects negatively the quality of the depth image provided by the Kinect sensor, by increasing the noise in the input image that reflects in the reliability of the foreground detection process. Nevertheless, it has to be noted that in both scenarios, the method built on top of the DEPTH sensor behaves better than the same method fed with traditional RGB images. We note again that the combination of the two sensors through the *max* rule allows to achieve higher and more balanced performance with respect to each single sensor.

4.2.4. Analysis of results with respect to the frame rate

We also study how the frame rate of the video affects the overall performance of the people counting method. In Fig. 5 we show the *f-index* achieved by our method when processing the videos in the $Dataset_C$ at the 640 × 480 resolution at different frame rates from 30 fps to 5 fps with step 5. Not surprisingly, we note that the lower is the frame rate, the lower is the accuracy. Performance is quite stable from 30 to 15 fps, while it dramatically decreases for lower fps. This is motivated by the fact that at low frame rates the crossing speed becomes too high and does not allow to activate both cells of the sensor in the proper sequence.

4.2.5. Analysis of results on the $Dataset_U$

The previous observations are still more evident when analyzing the results in Table 5: in the challenging scenario proposed by the video sequences contained in the $Dataset_U$, the exploitation of the depth information for the proposed method allows to largely outperform the approach using RGB images. We also note a general degradation of the performance of the approach with respect to results achieved on the $Dataset_C$. From a detailed analysis of the errors we found different causes discussed below.

Table 5
Performance achieved by the proposed method on the $Dataset_U$.

Sensor	Flow density	Recall	Precision	f-index
RGB	Isolated transits	0.947	0.958	0.952
	Group of persons	0.762	0.864	0.810
DEPTH	Isolated transits	0.973	0.973	0.973
	Group of persons	0.875	0.983	0.926
RGB∨DEPTH	Isolated transits	0.981	0.963	0.972
	Group of persons	0.933	0.904	0.918

*Non uniform illumination:*The considered scene is illuminated with a focused light from different lighting points which cast shadows all around the persons. This typically has a very negative impact on the performance of the method based on RGB sensor as it is the cause of both false positive (the persons may occupy more stripes) and false negatives (persons in a row may appear as a single object); due to the immunity of the used DEPTH sensor to the illumination, this aspect has no effect on the performance of this type of sensor.

*Highly dense groups of persons:*This database comprises roughly fifteen occurrences of passages of groups composed by eight to twelve persons crossing the sensor in opposite directions or in the same direction. In such cases both the DEPTH and RGB sensors tends to underestimate the counting. This phenomenon motivates the lower *Recall* of the system on this dataset with respect to the controlled scenario.

*Passages with objects:*There are also some residual situations (roughly a dozen over the whole dataset) when a person crosses the virtual line by carrying an object as a chair, a stepladder, a desktop PC, a trolley, a large bag, or with open arms. In such cases the method (independently of the used sensor) generates a false positive if the person carries the object laterally so as to activate more than $2*\theta_K$ stripes. Conversely, if the object extends perpendicularly with respect to the crossing line no false positive is generated. Generally speaking, since the a priori probability of occurrence of such cases in real installations is typically very low, their impact on the overall performance is negligible.

4.3. Comparison with other methods

In this subsection we compare the proposed method with two other methods available in the scientific literature. Specifically, we are interested to analyze the performance improvement ascribable to the pre-processing stage newly introduced with respect to our previous approach in [12]; furthermore, we are also interested to compare our method with respect to a people counting approach based on tracking. To this aim, we implemented a people counting method that uses a tracking method inspired to that described in [1] and generates a counting event when the centroid of the tracked blob crosses the virtual line. In Table 6 it is reported the comparison between the performance of the proposed method both with and without the pre-processing stage and the tracking-based method. The experiments were carried out on the videos of the $Dataset_C$, at 30 fps and 640 × 480 resolution.

The proposed method achieves the best performance in terms of Recall and Precision. The comparison with [12] method shows that, by applying the pre-processing step, we obtain an improvement of the accuracy on either RGB or DEPTH images. The Gaussian filter allows, on one hand, to reduce the false positives due to the presence of noise in the image and, on the other hand, to decrease the false negatives caused by the incorrect evaluation of the activation sequence of the sensor. The performance achieved by the tracking-based method is significantly lower and the results confirm the validity of our choice to avoid tracking. In fact, the use of a tracking algorithm can cause many false positives and false negatives when foreground detection artifacts induce splits or merges of the blobs. The main problem in the RGB INDOOR scenario is the presence of reflections, while in the RGB OUTDOOR scenario the shadows cause most of errors. Using the DEPTH sensor, there are no troubles with reflections and shadows, but the splits or merges of the blobs and the lack of a counting algorithm depending on the size of the crossing objects produce many errors.

4.4. Analysis of the computational efficiency

We analyze the computational requirements of the method on the following three processing platforms:

1. Intel(R) Core(TM) i7-3770S CPU @ 3.10 GHz with 4GB of RAM (hereinafter called PC platform)
2. 400˜MHz Multi-Thread RISC CPU with 0.5GB of RAM (hereinafter MIPS platform)
3. ARMv7 Processor rev 5 (v7l) with 1CPU of 600˜MHz with 0.8GB of RAM (hereinafter ARM platform)

These three platforms were chosen as they can be considered as representative of the computational architectures adopted in real world installations. As a matter of fact, the PC platform represents the most common situation when a large number of cameras are connected to a central workstation and several instances of the people counting algorithm run for processing the video streams in real time (*server side* execution). The analysis on the second considered platform, namely the MIPS one, allows to consider the so called *edge side* execution, where the people counting is performed directly on board of the acquisition device; there is a large interest toward the implementation of such type of distributed architecture where the processing is performed on the periphery and the camera just becomes a sensor providing counting information to the central data collection workstation. There are two great advantages when performing the counting on board of the camera. First, there is no need to send the whole video stream over the network to the central server, as just few bytes per each person passage are to be communicated to the server thus without impacting on the network. Second, the distributed architecture allows to overcome the problem of the single point of failure that conversely afflicts centralized installations realized using PC platforms. On the other hand it has to be noted that the edge side

Table 6
Overall performance comparison between the proposed method with the pre-processing stage, without the pre-processing stage as described in [12] and the tracking line sensor on the $Dataset_C$.

Sensor	Method	TP	FN	FP	Recall	Precision	f-index
RGB	Proposed method with pre-processing	879	89	9	0.908	0.990	0.947
	Proposed method without pre-processing	874	94	20	0.903	0.978	0.939
	Tracking method	793	175	177	0.819	0.818	0.818
DEPTH	Proposed method with pre-processing	935	33	0	0.966	1.000	0.983
	Proposed method without pre-processing	928	40	0	0.959	1.000	0.979
	Tracking method	852	116	77	0.880	0.917	0.898
RGB∨DEPTH	Proposed method with pre-processing	958	10	7	0.990	0.993	0.991

Table 7
Frame processing time in microseconds and profiling of the proposed method on the $Dataset_C$ at different resolutions.

	640 × 480			320 × 240			160 × 120		
	PC	ARM	MIPS	PC	ARM	MIPS	PC	ARM	MIPS
Pre-processing	551	6623	22,386	136	1551	4155	33	364	709
	10.3%	10.9%	10.8%	9.7%	10.0%	9.4%	9.1%	9.1%	7.7%
Foreground detection	3241	41,377	1,599,332	821	10,424	34,344	209	2628	7099
	60.4%	68.1%	76.8%	58.6%	67.1%	77.5%	58.3%	65.4%	77.4%
Background update	3	54	4295	1	17	1086	1	8	298
	0.1%	0.1%	2.1%	0.1%	0.1%	2.5%	0.2%	0.2%	3.3%
Counting	1574	12,968	21,347	445	3542	4713	116	1016	1062
	29.3%	20.9%	10.3%	31.7%	22.8%	10.6%	32.4%	25.3%	11.6%
Total FPS	189	17	5	728	65	23	2855	254	109

execution raises the issue of a proper engineering of the software in order to run on the very low processing power CPU typically available on the cameras. The third architecture is roughly halfway between the previous ones (we will refer to this scenario as *server to the edge* execution) in the sense that it is realized using low profile single board computers whose processing power is higher than that available on a camera but less than a standard PC. This situation may arise in those installations where a large number of streams from cameras not supporting embedded execution have to be elaborated, and the cameras are in small local groups (typically less than a dozen devices) but with the groups distributed on a geographic network (we can think to a large retail chain with several shops distributed over the world, and in each shop few counting systems are deployed to monitor the entrances and the exits).

In Table 7 we report for each processing stage of the proposed method, for each platform and for each of the three considered resolutions, the processing times expressed both as absolute values in microseconds and as percentage over a frame. Data represent the average values of the processing time over all the videos of the $Dataset_C$. Foreground detection is the most computationally intensive step of the whole processing requiring on each platform at least 60% of the time, while the optimizations done on the background update phase allow to make negligible the impact of this phase over the processing times. We also note that the preprocessing phase has a reduced although not negligible impact on the overall computation times contributing for roughly 10%. Moreover, there are very large differences among the processing times over the different platforms; in fact, there is an increase of roughly an order of magnitude going from the *server side* execution (PC platform) to the *server to the edge* execution (ARM platform), and almost another order of magnitude increase with the MIPS platform (*edge side* execution).

It is also interesting to consider in Table 7 the average processed frames per second (total FPS) for each platform using the proposed people counting method over the $Dataset_C$ at the original 640 × 480 resolution and at the scaled 320 × 240 and 160 × 120 resolutions. Not surprisingly, the average value of the processed frames per second quadruples after halving the horizontal and the vertical resolution. With 640 × 480 resolution, the method is able to process 189 fps on a single core of a server, over 700 fps at a 320 × 240 resolution and even 2855 fps at a 160 × 120 resolution. With ARM platform the method is able to reach 17 fps, that is compatible for its adoption in practical scenarios (see Section 4.2 and Fig. 5). Conversely, the processing fps on the MIPS platform at 640 × 480 resolution is only 5 that is far below the minimum suggested fps required to achieve reliable counting. However at a 320 × 240 resolution this platform is able to process 23 fps and over 100 fps at a 160 × 120 resolution. As shown in Section 4.2, the rescaling of the resolution at half or at a quarter causes a loss of accuracy between 2% and 5%. The MIPS platform can then ensure reliable counting with a resolution less than or equal to 320 × 240, accepting a small degradation of accuracy.

It is also important to recall that the software implementation of the proposed approach does not exploit any parallelization of the code, thus it does not benefit of the availability of multiple cores on some platforms, as for instance the ARM and the PC. For the sake of comparability of the results, we run the tests for each platform on a single video stream; however, from experiments not reported in this paper we verified that the number of processed streams at a given fps scales linearly with the number of cores.

5. Conclusions

In this paper we have proposed a new method for counting people from overhead cameras. The method has been specifically designed in order to take into account two main issues that may hinder the adoption in real world scenarios: the counting performance and the computational efficiency. The first goal has been achieved by defining a method based on foreground detection and background update for the detection of the persons crossing the virtual counting line and a sensor that adopts a finite state automaton for determining the crossing direction; such approach demonstrated to be more robust than an object tracking based approach. The computational efficiency goal has been achieved by a careful implementation of the software that eliminates floating point operations, uses SIMD instructions of the processors and reduces the update rate of the background.

The method has been validated on a dataset of images that has been specifically devised and collected in order to account for the main issues that arise in typical installations which may affect the counting performance: the acquisition technology (traditional RGB camera and depth sensor), the installation scenario (indoor and outdoor), the density of the people flow (isolated people and groups of persons), the acquisition frame rate (from 5 to 30 fps with step 5), the image resolution (160 × 120, 320 × 240 and 640 × 480). The results demonstrate that passages of isolated persons can be detected with the same performance ($f\text{-}index = 0.99$) with both traditional RGB cameras and depth sensors. A similar level of performance can be achieved when using the depth sensor also in the case of higher density passages, while a significant performance drop is revealed in case of passages of groups of persons when RGB cameras are employed. The results also show that performance is improved by the combined adoption of the DEPTH and RGB sensors, which allows the system to reach a more balanced trade-off between false positives and false negatives. Image resolution also affects performance of the method as the performance smoothly decreases as far as we decrease the resolution. On the contrary, we notice an hard performance drop when the image acquisition rate goes below a cut-off value that for a typical people counting set-up, as the one used for the tests in this paper, is around 15 fps. The method has been also tested on three

different computer architectures which are very different with respect to their computational power, and the results demonstrated that the provided implementation of the method can process in real time at 320 × 240 resolution on low power devices as those typically embedded on commercial smart cameras, while it can process simultaneously more than ten 640 × 480 streams per CPU core on a high-end workstation.

Future research efforts in this area will include: the validation of the method on a larger dataset that should comprise sequences taken from field installations coming from applicative domains characterized by different installation requirements, as the transport domain where the cameras are typically installed much closer to the heads of the persons (at the doors of buses, trains, metros); the validation of the method when other depth sensors are used (as stereo cameras or time of flight devices) or thermal cameras.

Acknowledgments

This work has been partially supported by Regione Campania in the framework of the "Embedded Systems in Critical Domain" project, founded by POR Campania FSE 2007/2013, and by A.I. Tech srl, a spin-off company of the University of Salerno.

References

[1] C.-H. Chen, T.-Y. Chen, D.-J. Wang, T.-J. Chen, A cost-effective people-counter for a crowd of moving people based on two-stage segmentation, J. Inf. Hiding Multimed. Signal Process. 3 (1) (2012) 2073–4212.
[2] J. Barandiaran, B. Murguia, F. Boto, Real-time people counting using multiple lines, in: Proceedings of the Ninth International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'08, IEEE, 2008, pp. 159–162.
[3] T.-H. Chen, T.-Y. Chen, Z.-X. Chen, An intelligent people-flow counting method for passing through a gate, in: Proceedings of 2006 IEEE Conference on Robotics, Automation and Mechatronics, IEEE, 2006, pp. 1–6.
[4] B. Antic, D. Letic, D. Culibrk, V. Crnojevic, K-means based segmentation for real-time zenithal people counting, in: Proceedings of 2009 16th IEEE International Conference on Image Processing, ICIP, IEEE, 2009, pp. 2565–2568.
[5] S. Mukherjee, B. Saha, I. Jamal, R. Leclerc, N. Ray, Anovel framework for automatic passenger counting, in: Proceedings of 2011 18th IEEE International Conference on Image Processing, ICIP, IEEE, 2011, pp. 2969–2972.
[6] K. Terada, D. Yoshida, S. Oe, J. Yamaguchi, A method of counting the passing people by using the stereo images, in: Proceedings of 1999 International Conference on Image Processing, ICIP'99, vol. 2, IEEE, 1999, pp. 338–342.
[7] T. Yahiaoui, C. Meurie, L. Khoudour, F. Cabestaing, A people counting system based on dense and close stereovision, in: Image and Signal Processing, Springer, 2008, pp. 59–66.
[8] T. Van Oosterhout, S. Bakkes, B.J. Kröse, Head detection in stereo data for people counting and segmentation., in: Proceedings of International Conference on Computer Vision Theory and Applications, VISAPP, 2011, pp. 620–625.
[9] X. Zhang, J. Yan, S. Feng, Z. Lei, D. Yi, S.Z. Li, Water filling: Unsupervised people counting via vertical kinect sensor, in: Proceedings of 2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance, AVSS, IEEE, 2012, pp. 215–220.
[10] P. Vera, D. Zenteno, J. Salas, Counting pedestrians in bidirectional scenarios using zenithal depth images, in: Pattern Recognition, Springer, 2013, pp. 84–93.
[11] H. Fu, H. Ma, H. Xiao, Scene-adaptive accurate and fast vertical crowd counting via joint using depth and color information, Multimed. Tools Appl. 73 (1) (2014) 273–289.
[12] L. Del Pizzo, P. Foggia, A. Greco, G. Percannella, M. Vento, A versatile and effective method for counting people on either RGB or depth overhead cameras, in: Proceedings of 2015 IEEE International Conference on Multimedia Expo Workshops, ICMEW, 2015, pp. 1–6, doi:10.1109/ICMEW.2015.7169795.
[13] R.J. Radke, S. Andra, O. Al-Kofahi, B. Roysam, Image change detection algorithms: a systematic survey, IEEE Trans. Image Process. 14 (3) (2005) 294–307.
[14] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, C. Rosenberger, Review and evaluation of commonly-implemented background subtraction algorithms, in: Proceedings of the 19th International Conference on Pattern Recognition, ICPR 2008, IEEE, 2008, pp. 1–4.
[15] T. Bouwmans, F. El Baf, B. Vachon, Background modeling using mixture of Gaussians for foreground detection-a survey, Recent Pat. Comput. Sci. 1 (3) (2008) 219–237.
[16] S. Brutzer, B. Höferlin, G. Heidemann, Evaluation of background subtraction techniques for video surveillance, in: Proceedings of 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2011, pp. 1937–1944.
[17] A. Sobral, A. Vacavant, A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos, Comput. Vis. Image Underst. 122 (2014) 4–21.
[18] A. Sanin, C. Sanderson, B.C. Lovell, Shadow detection: A survey and comparative evaluation of recent methods, Pattern Recognit. 45 (4) (2012) 1684–1695.
[19] A. Prati, I. Mikic, M.M. Trivedi, R. Cucchiara, Detecting moving shadows: algorithms and evaluation, IEEE Trans. Pattern Anal. Mach. Intell. 25 (7) (2003) 918–923.
[20] D. Conte, P. Foggia, G. Percannella, M. Vento, Removing object reflections in videos by global optimization, IEEE Trans. Circuits Syst. Video Technol. 22 (11) (2012) 1623–1633.
[21] D. Conte, P. Foggia, G. Percannella, F. Tufano, M. Vento, An experimental evaluation of foreground detection algorithms in real scenes, EURASIP J. Adv. Signal Process. 2010 (2010) 1–11, no. 373941, doi:10.1155/2010/373941.
[22] A. Ellin, G. Dolsak, The design and application of rotary encoders, Sens. Rev. 28 (2) (2008) 150–158, doi:10.1108/02602280810856723.
[23] V. Carletti, L. Del Pizzo, G. Percannella, M. Vento, Foreground detection optimization for SoCs embedded on smart cameras, in: Proceedings of the International Conference on Distributed Smart Cameras, ICDSC, ACM, New York, NY, USA, 2014, Article 31, 5 pages, doi:10.1145/2659021.2659060.