2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)

Real-time people counting from depth imagery of crowded environments

Enrico Bondi
enrico.bondi88@gmail.com
University of Florence

Lorenzo Seidenari
lorenzo.seidenari@unifi.it
University of Florence

Andrew D. Bagdanov
bagdanov@cvc.uab.es
Computer Vision Center, Barcelona

Alberto Del Bimbo
alberto.delbimbo@unifi.it
University of Florence

Abstract

In this paper we describe a system for automatic people counting in crowded environments. The approach we propose is a counting-by-detection method based on depth imagery. It is designed to be deployed as an autonomous appliance for crowd analysis in video surveillance application scenarios. Our system performs foreground/background segmentation on depth image streams in order to coarsely segment persons, then depth information is used to localize head candidates which are then tracked in time on an automatically estimated ground plane. The system runs in real-time, at a frame-rate of about 20 fps. We collected a dataset of RGB-D sequences representing three typical and challenging surveillance scenarios, including crowds, queuing and groups. An extensive comparative evaluation is given between our system and more complex, Latent SVM-based head localization for person counting applications.

1. Introduction

People counting and crowd analysis are two important and classical computer vision problems in video surveillance. People counting can be loosely understood as the instantaneous estimation of the number of persons present in a scene, and crowd analysis can be understood to be the higher-level analysis of behaviors of groups of people in crowded scenes. In crowded, public environments like airports, shopping malls and railway stations, automatic people counting and crowd analysis can alert operators to potentially dangerous situations without having to dedicate manpower to monitoring many video streams.

Despite the importance of people counting in modern video surveillance contexts, the problem remains a subject of active research. Techniques for people counting can be divided into two, broad categories. The *Detection+Counting* approach uses a pedestrian detector to identify candidate person regions in an image, then typically applies some type of segmentation or disambiguation post-processing to verify person candidates before counting. Recent counting-by-detection approaches use either RGB [4, 2] or depth imagery [14, 7, 9].

The first step for counting objects is usually to apply motion segmentation to detect moving scene elements. Fehr *et al*. detect moving objects using a mixture of Gaussians and project blobs on both ground and "head" planes. The intersection of these two blob projections is considered the area occupied by people. This approach assumes people density to be constant in every moving blob and therefore may be inaccurate in case this assumption is not true. Kong *et al*. rely on histograms of normalized features to take into account perspective [2]. Their method requires some parameter learning.

The simplest way of exploiting depth for counting is to orient cameras perpendicularly to the ground plane [14]. This approach is simple and effective, but poses a strong constraint on system deployment. Moreover, such sensor deployment may strongly reduce the area covered by sensors when the ceiling is not high enough. To address this issue some approaches work with arbitrary camera orientations [7, 9]. Fu *et al*. apply template matching to locate head and shoulder patterns that are used as seeds for segmentation. Hsie *et al*. project depth clouds onto the ground plane and exploit morphology to find people blobs. The method in [7] deals with crowded environments by splitting pairs of incorrectly fused blobs. In case of severe crowding this approach is limited and may not split groups of more than two people for which there is not a single convexity in the extracted hull. Similarly, the approach in [9] can not really handle extremely crowded scenes since the projection of blobs on the ground plane may result in fused blobs for close targets. Chan applies a Gaussian process regressor on segmented blobs to estimate the people count [1]. This hybrid approach needs training data (i.e. annotated people

978-1-4799-4871-0/14/$31.00 ©2014 IEEE

counts for a number of frames).

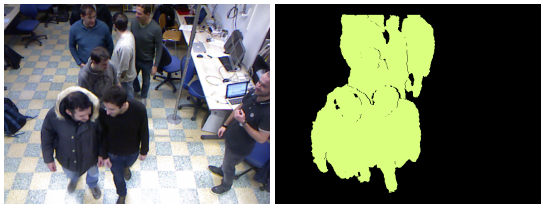
The other category of techniques for people counting is *Feature-Based Counting*. These techniques, rather than relying on an explicit detection phase to identify and delineate persons in an image, exploit image features to estimate the true count [11, 12, 13]. The idea of counting-by-regression was recently exploited in [11, 12]. Feature-based counting has been proposed for extremely crowded images by Idress *et al*. [10]; they propose the exploitation of multiple features that correlate with crowd density such as head detector output and SIFT location. Lempitsky *et al*. apply the idea of learning to count with minimal supervision. They require only a single dot per object to count as annotation and learn a regressor on features correlated with object density. Both these approaches only count the objects in an image and do not provide a true estimate of the number of individuals crossing into and out of an area. To solve this issue in a regression framework, Ma *et al*. proposed a counting-by-regression approach on a time-slice image of the area of surveillance [12].

Counting by regression has the strong disadvantage of requiring training data. So every deployment of such a system will typically need some annotated frames. One of our driving design goals for our counting system is that the deployment require minimal human intervention. We therefore decided to implement a counting-by-detection method based on 3D sensors. Our algorithm runs in real time, requires no training data and can exploit cheap, off-the-shelf sensors. The approach uses depth information to obtain highly accurate foreground segmentation, followed by well-localized head detection and projection to an automatically estimated ground plane. RGB-D cameras like the Microsoft Kinect are becoming commodity devices deployed in many application scenarios. These sensors provide synchronized depth information about the scene in parallel with the RGB video stream.

In the next section we give a brief overview of our entire system. Then in Section 3 we describe our approach to head detection and localization in depth image streams. In Section 4 we describe a multi-target tracking system used to track persons from entry into until exit from the area of surveillance, and in Section 5 we report on a series of experiments performed to evaluate our system.

2. System overview

Our people counting system is designed to provide accurate, real-time performance in crowded environments. We take an appliance approach to the design of our system, envisioning a distributed surveillance system where each appliance communicates with a central surveillance engine implementing logic for global surveillance and resource optimization tasks. As an example, a large department store may wish to monitor people inflow and outflow at given



(a) RGB frame (b) Incorrect blob detection

Figure 2: Example of incorrect blob labeling in crowd.

times of the day. Each door can be equipped with our people counter that communicates how many customers enter or exit the store. Another scenario could be related to public safety, where police agencies may want to monitor the birth of agglomerates of people in sensitive public areas.

In our application scenarios we require simple, effective image processing and computer vision techniques in order to deploy the system on low-end hardware. We show the information flow from sensor to high-level person counting and crowd analysis algorithms in Figure 1. Our design is simple and effective. We assume depth images from a calibrated stereo system as input to our system. We first extract the foreground using the depth stream. Applying simple edge detection with a Sobel operator and combining the foreground edges in the depth map with the foreground map we can separate almost every blob in the scene. We localize heads with the algorithm explained in Section 3 and project the top head point onto the automatically-estimated ground plane. We then use Multi-Target Tracking (MTT) with simple nearest neighbor data association to estimate people flow.

3. Head localization

Since ours is a counting-by-detection approach, the first step of our pipeline is the detection of each subject. Instead of searching for a full pedestrian we focus in the localization of heads in the RGB-D video stream. We begin by removing the background through selective running average background subtraction. Denoting the background pixel model at time t as $B_t(x, y)$ and the new frame as $F_t(x, y)$, a depth pixel is considered foreground if $|B_t(x, y) - F_t(x, y)| > \Delta$. The background at pixel x, y is:

$$B_t(x,y) = \alpha\delta_t(x,y)F_{t-1}(x,y) + (1-\alpha\delta_t(x,y))B_{t-1}(x,y), \quad (1)$$

where

$$\delta_t(x,y) = \begin{cases} 0 & \text{if } |B_t(x,y) - F_t(x,y)| > \Delta \\ 1 & \text{otherwise.} \end{cases} \quad (2)$$

To identify the heads of each target we first label connected components in the foreground image. Unfortunately,

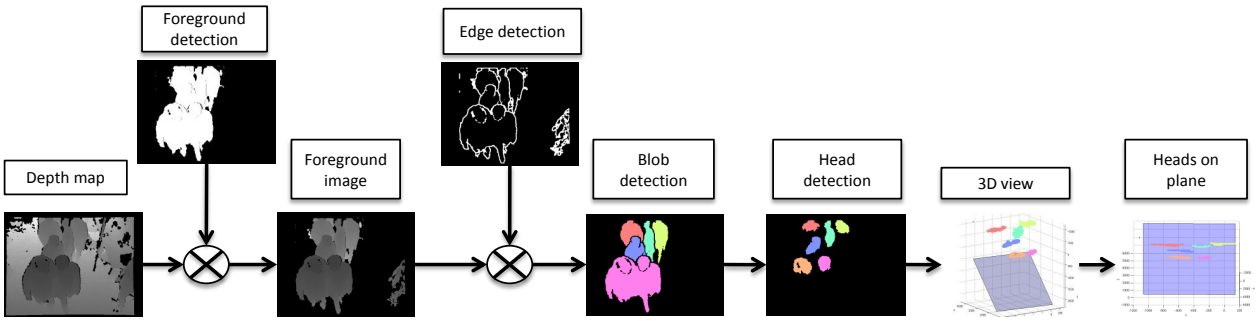


Figure 1: A diagram of the principal modules in our system. The depth maps of frames are processed to detect moving objects. Heads are then located in the foreground-segmented depth map, and the highest point is projected onto the ground plane. These positions are then used as detections for input to a multi-target tracking system and the crowd analysis algorithm.

when crowding occurs all foreground pixels can be incorrectly grouped into a blob as shown in Figure 2. To avoid connecting all foreground pixels we first apply a mask computed by binarizing the response of the Sobel operator on the depth image. Using this mask we separate blobs at different distances from the sensor. As shown in Figure 1, blob detection is significantly improved by this simple step: the edges in the depth map provide essential cues required for segmenting people moving in unison an who are thus not well-segmented by the background model alone.

For each blob B we apply the following procedure to localize the corresponding head. Consider the set of pixels $P = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\} \in B$. We compute $\hat{d} = \min_p D(x, y)$, where $D(x, y)$ is the depth value at location (x, y). We retain pixels (x, y) of B for which $D(x, y) \in [\hat{d}, \hat{d} + \epsilon]$. After processing each blob in this way only the head pixels remain in the foreground mask and relabeling this image yields accurate head localization in the image plane. Blobs with an area of less than 300 pixels are discarded.

4. People tracking

Head localization as described in the previous section allows frame-wise people counting. This is already a useful feature that has many security and public safety applications. To count the exact number of people passing from a controlled area we need to detect entering and exiting events. This is a problem that can be solved only with multiple target tracking. Tracking on the ground plane is also beneficial in case of multiple devices controlling partially overlapped areas.

Using the internal camera parameters:

$$K = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

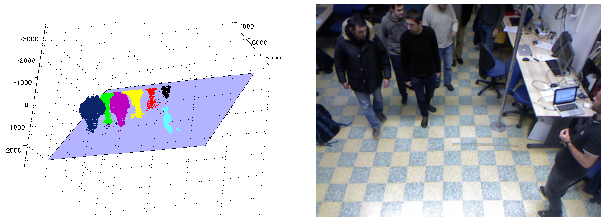


Figure 3: Example of ground plane estimation (left) in a crowded scene (right). Each person point cloud has a different color.

we estimate the 3D location of points using:

$$\begin{aligned} x_{3D} &= [(x_{2D} - x_0) \cdot D(x_{2D}, y_{2D})]/\alpha_x \\ y_{3D} &= [(y_{2D} - y_0) \cdot D(x_{2D}, y_{2D})]/\alpha_y \\ z_{3D} &= D(x_{2D}, y_{2D}). \end{aligned}$$

To extract the ground plane we run RANSAC to fit a plane on the entire point cloud [6, 8]. This is a simple setup step that need be run only once or in cases the camera orientation is changed. As shown in Figure 3, the ground plane is accurately estimated. To count people an area of analysis must be defined. The user can easily select a polygon on the ground in the RGB frame that we then project on the ground plane using the stereo calibration.

To perform multiple target tracking on the ground plane we use a very simple yet effective approach. After detecting objects (i.e. head projections on the plane), we perform data association. At each frame we build a distance matrix:

$$C = \begin{bmatrix} c_{11} & c_{12} & ... & c_{1m} \\ c_{21} & c_{22} & ... & c_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ c_{n1} & c_{n2} & ... & c_{nm} \end{bmatrix} \quad c_{ij} = d(t_i, p_j). \quad (4)$$

that maps detections p_j to existing tracks t_i. We use a greedy approach that iteratively associates track/detection

Seq	Frames	Persons	Persons/frame	Person flow
FLOW	1260	3542	2.80	28
QUEUE	918	5031	5.48	8
GROUPS	1180	9057	7.68	0

Table 1: Dataset statistics. "Persons" is the total number of detectable persons, and "Person flow" is the number of persons entering and exiting the scene.

pairs with the smallest distance. We do not associate pairs for which c_{ij} is more than 30 cm. Every unassociated track is temporarily disabled. For each unassociated detection a new track is created. We attempt to re-associate disabled tracks in subsequent frames until their time-to-live expires (10 frames).

Counting people is now trivial: for a specified direction of counting, everytime a target inside the controlled area exits the area a counter is incremented. The instantaneous count per frame is given by the number of detections obtained with the method in Section 3.

5. Experiments

Here we report on a number of experiments performed to evaluate the performance of our approach in a number of challenging and realistic situations. We also describe a publicly available dataset we developed of RGB-D sequences designed specifically for evaluation of people counting applications.[1] We believe this to be one of the most complete RGB-D dataset resources for people counting and crowd analysis applications.

5.1. Dataset

We acquired a dataset of RGB-D imagery under typical conditions for visual surveillance applications in crowded environmnents. We recorded three video sequences that encompass three common people grouping and motion scenarios:

- In the **FLOW** sequence we asked the participants to walk straight from one point to another of the room. This sequence is useful to evaluate people counting systems in scenarios such as retail access or underground/train station pedestrian walkways.
- In the **QUEUE** sequence we asked participants to act as if waiting in line. People move slowly forward as people ahead are served. This sequence is challenging since pedestrians can be absorbed by the background while waiting and is useful to test the robustness of the background modeling under real conditions.
- Finally, in the **GROUPS** sequence we asked participants to split into two groups and talk to each other without exiting the controlled area. This sequence represents scenarios related to public safety in open and closed spaces.

In all of the three sequences people are highly occluded as seen in in Figure 4. Table 1 reports some statistics on the sequences in our dataset.

5.2. Experimental results

We performed a comparison of our counting by detection method with the state of the art Latent SVM (LSVM) pedestrian detector [5]. We used the model trained on the INRIA dataset since on our dataset it gave the best performances. We localize heads in frames using LSVM by running the part-based detector on the frame. Then, after non-maximum suppression we extract all the head parts and then further suppress overlapping heads, which we found removed some false positive detections. We also report results using an improved version of this technique that exploits the segmented foreground mask; after heads are localized, we remove all boxes that do not contain at least 30% foreground pixels. We refer to this segmentation-enhanced version as "LSVM+Segm" below.

We leave the 3D localization of targets and tracking the same, and therefore project the lowest point of the detected head onto the ground plane and use this data as detections for input to greedy, nearest neighbor data association scheme. The lowest point is the most likely to fall on the person point cloud.

We measure precision and recall for people detection on the ground plane. We map the ground truth and detected head points onto an area of 0.5×0.5 meters around the projected point and consider a detection correct if the VOC overlap score is above 0.5 [3]. As in the PASCAL VOC challenge, duplicate detections are considered false positives and are not associated even if they have a VOC score higher than 0.5.

We also measure the counting accuracy with the mean absolute error:

$$\mathrm{MAE} = \frac{1}{N}\sum_{f=1}^{N} |\mathrm{count}(f) - \mathrm{gt}(f)|, \qquad (5)$$

over all frames $f = 1, \ldots, N$. This averages the per frame discrepancies between predicted count $\mathrm{count}(f)$ and the ground truth $\mathrm{gt}(f)$. With this value we have a more interpretable datum to assess system performance and compare its performance with competing ones. This measure, however, is not rigorous in a scientific sense since it can be influenced by lucky guesses. As an example, a frame with three persons and three detections is considered correct even if the bounding boxes are completely decorrelated. Our system runs in real time (20fps), while running the LSVM de-

[1] http://www.micc.unifi.it/vim/datasets/micc-people-counting/

(a) Our method (b) LSVM (c) LSVM+Segm.

Figure 4: Example detections for our method, Latent SVM and Latent SVM+Segm.

	Precision	**Recall**	**MAE**
Our Method	**0.9753**	**0.8926**	**0.4138**
Latent SVM+Segm [5]	0.9082	0.5955	1.2502
Latent SVM [5]	0.8478	0.6054	1.0302

Table 2: Comparison of our method with LSVM and LSVM+Segm on the FLOW sequence.

	Precision	**Recall**	**MAE**
Our Method	**0.9534**	**0.9169**	**0.6521**
Latent SVM+Segm [5]	0.8516	0.5538	2.6314
Latent SVM [5]	0.7915	0.5673	2.3097

Table 3: Comparison of our method with LSVM and LSVM+Segm on the QUEUE sequence.

	Precision	**Recall**	**MAE**
Our Method	**0.9795**	**0.9453**	**0.5861**
Latent SVM+Segm [5]	0.8725	0.5633	3.1781
Latent SVM [5]	0.8236	0.5798	2.8448

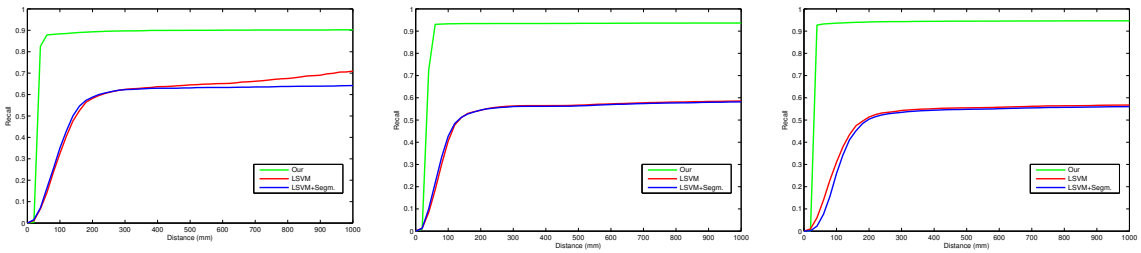
Table 4: Comparison of our method with LSVM and LSVM+Segm on the GROUPS sequence.

tector on our frames which are 640x480 takes around 6 seconds per frame.

As can be seen in Tables 2, 3 and 4 our technique greatly improves over LSVM in frame-by-frame people counting for the basic LSVM and the version using the depth image for background modeling. In Table 2 we report results on the FLOW sequence. On this sequence the relatively high speed of people renders the background model accurate, although the fast motion can also result in missed head detections, affecting recall. Recall is also affected by the high occlusions rate in this sequence.

In Table 3 we report results of our approach and LSVM on the QUEUE sequence from our dataset. In this sequence precision is much lower than on the other two. Finally, in Table 4 we report our results compared to LSVM and LSVM+Segm on the GROUPS sequence. Our approach performs best on this sequence, likely due to the lower occlusion rate.

In every sequence our algorithm has a substantially better performance with respect to LSVM and its improved version LSVM+Segm. In terms of MAE our method has an average error of less than one person per frame, while LSVM and LSVM+Segm produce counts off by 2-3 persons on GROUPS and QUEUE and by 1 person on FLOW.



(a) FLOW (b) QUEUE (c) CROWD

Figure 5: Recall distance curves on the three sequences from our dataset.

In terms of precision and recall, we obtain a similar behavior on the three video sequences, with the segmentation step making LSVM more selective increases precision significantly. Our method is still the best with a precision always higher then 0.95 and a recall of at least 0.89.

Requiring a VOC score of 0.5 can be considered very strict for assessing performance. For this reason we also add an evaluation of the three methods with less strict criteria. For each dataset we compute a recall-distance curve similar to [12]. For each frame in every sequence we associate ground truth detections with predicted ones using the data association in Section 4. In Figure 5 we plot recall as a function of distance threshold for considering detections correct. To compute these curves we run the association algorithm for a set of distance thresholds and compute the recall for each threshold over the whole sequence.

In all experiments our curve is much steeper than the others. We also see qualitatively from the frames in Figure 4 that our method extracts heads at more precise locations than Latent SVM and its version exploiting foreground segmentation. Recall that LSVM-based methods reach a value close to 0.5 when the distance threshold is more than 300 mm. This means that LSVM-extracted heads are correlated with the head locations. The lower maximum recall indicates that LSVM labels fewer heads than our approach, and thus suffers more in terms of recall than precision.

6. Discussion

In this paper we described a robust, real-time system for people counting and crowd analysis. The main application scenario envisaged is the control of access to designated areas of surveillance. Our approach uses RGB-D imagery, exploiting depth information to accurately segment the foreground from the background and to segment persons from each other even in crowded sequences. A multi-target tracker using greedy data association is used to track entrance and exit of people from a designated area. Experimental results show our approach to significantly outperform more complex detection techniques such as Latent SVM for head localization.

References

[1] A. B. Chan, Z.-S. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Proc. of CVPR*, pages 1–7. IEEE, 2008.

[2] D. G. Dan Kong and H. Tao. A viewpoint invariant approach for crowd counting. In *Proc. of ICPR*. 2006.

[3] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[4] D. Fehr, R. Sivalingam, V. Morellas, N. Papanikolopoulos, O. Lotfallah, and Y. Park. Counting people in groups. In *Proc. of AVSS*. IEEE, 2009.

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

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

[7] H. Fu, H. Ma, and H. Xiao. Real-time accurate crowd counting based on rgb-d information. In *Proc. of ICIP*, pages 2685–2688. IEEE, 2012.

[8] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[9] C.-T. Hsieh, H.-C. Wang, Y.-K. Wu, L.-C. Chang, and T.-K. Kuo. A kinect-based people-flow counting system. In *Proc. of (ISPACS)*. IEEE, 2012.

[10] H. Idrees, I. Saleemi, C. Seibert, and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. In *Proc. of CVPR*. 2013.

[11] V. S. Lempitsky and A. Zisserman. Learning to count objects in images. In *Proc. of NIPS*, 2010.

[12] Z. Ma and A. B. Chan. Crossing the line: Crowd counting by integer programming with local features. In *Proc. of CVPR*. IEEE, 2013.

[13] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Crowd counting using multiple local features. In *Proc of DICTA*, pages 81–88. IEEE, 2009.

[14] X. Zhang, J. Yan, S. Feng, Z. Lei, D. Yi, and S. Z. Li. Water filling: Unsupervised people counting via vertical kinect sensor. In *Proc. of AVSS*, 2012.