Counting People in Groups

Duc Fehr Ravishankar Sivalingam
Vassilios Morellas Nikolaos Papanikolopoulos
University of Minnesota, Twin Cities
Minneapolis, Minnesota, USA
{*fehr, ravi, morellas, npapas*}*@cs.umn.edu*

Osama Lotfallah Youngchoon Park
Johnson Controls, Inc.
Milwaukee, Wisconsin, USA
{*Osama.Lotfallah, Youngchoon.Park*}*@jci.com*

***Abstract*—Cameras are becoming a common tool for automated vision purposes due to their low cost. In an era of growing security concerns, camera surveillance systems have become not only important but also necessary. Algorithms for several tasks such as detecting abandoned objects and tracking people have already been successfully developed. While tracking people is relatively easy, counting people in groups is much more challenging. The mutual occlusions between people in a group make it difficult to provide an exact count. The aim of this work is to present a method of estimating the number of people in group scenarios. Several considerations for counting people are illustrated in this paper, and experimental results of the method are described and discussed.**

***Keywords*-people counting; crowd counting; background subtraction; pixel layering; shadow removal;**

I. Introduction

The ratio of the amount of information that can be collected by a camera to its cost is very large, which supports its use in almost every surveillance and inspection task. In these times of ever–rising security concerns, camera surveillance has become very important. There has been a lot of work done on several very essential tasks, such as abandoned object detection and people tracking.

While tracking people is relatively easy, being able to provide an accurate estimate of the number of people in a scene is extremely challenging, owing to the mutual occlusions between people in groups. Crowded scenes pose an even more daunting problem as the different foreground segments are usually overlapping in the field of view of the camera. Most systems rely on detecting each person individually and then counting these individuals, which cannot be done reasonably in crowded conditions.

There are several applications for counting people in groups. In public places with high crowd densities, statistics about human traffic flow can facilitate security management as well as urban planning. For security–critical areas such as airports and railway stations this is of extreme importance. There is also the possibility of military applications. For instance in urban warfare, soldiers might not be able to check every room of every building. Sending a camera into a room that could autonomously report how many people are present can help soldiers assess threat levels.

The paper presents a method estimating the count of people in groups. The rest of the paper is organized as follows: Section II covers a brief discussion of existing related work. Section III explains the main approach, including the background subtraction techniques used. Section IV presents the experimental results obtained with the implemented system. Section V states the conclusions of the paper and possible future research directions.

II. Related Work

There has been a lot of work in the field of crowd estimation and people counting. Some of the earlier work in this area has relied on heavily confined environments. For instance, Terada *et al.* [1] count people going through a gateway that only allows for a small number of people to go through at the same time. The stereo cameras used in this system are mounted overhead in order to avoid occlusions. Work by Velipasalar *et al.* [2] use a similar setup of ceiling–mounted cameras in order to avoid the problem of occlusions. The camera view is also very narrow and does not cover a large scene, unlike what we are trying to achieve.

Work realized by Zhao and Nevatia [3], [4] segments and tracks humans in crowded scenes using a human model composed of ellipses corresponding to the different parts of the body. This model helps keep track of individuals and is thus capable of giving a count of people in the scene. In [5], Rabaud and Belongie describe a method of counting crowded moving objects. Their counting technique is based on clustering a set of features in a video sequence and estimating the trajectories of these different detected clusters over time. The object counting is then performed based on these trajectories. Kong *et al.* [6] present a viewpoint–invariant way of counting people in a crowd. The key idea in this work is to use feature histograms in conjunction with feature normalization to make the algorithm viewpoint–invariant. In [7], Kilambi *et al.* present a technique to count groups of pedestrians utilizing camera calibration information. They project the foreground blobs onto different planes and use an area–based heuristic to estimate the number of people in a group.

All of the above mentioned counting methods rely on effective background subtraction. In this paper we compare

978-0-7695-3718-4/09 $25.00 © 2009 IEEE
DOI 10.1109/AVSS.2009.55

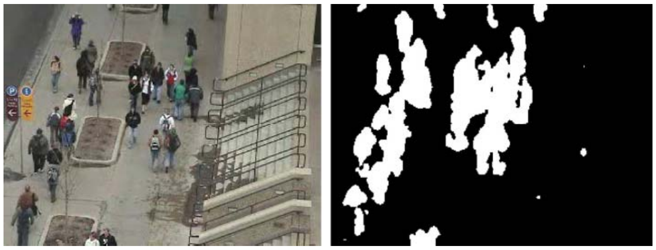


Figure 1. This figure shows a scene from the *Moos Tower* video and the corresponding foreground segmented image (using the *MG* method). The white blobs in the right image represent moving foreground objects.

the results from a people–counting system based on the algorithm from [7] using primarily two different background subtraction methods - a) using the mixture of Gaussians [8] as modified in Atev *et al.* [9], and b) using the pixel layering method for foreground detection by Patwardhan *et al.* [10]. Henceforth Atev *et al.*'s method will be referred to as the *MG* algorithm and Patwardhan *et al.*'s technique, the *layering* algorithm.

III. Group Count Estimation

Given a background segmented image of a scene, the goal is to estimate the count of people in it. Fig. 1 presents one such image from the *Moos Tower* video, which shows the scene next a bus stop in the University campus.

From this figure, the problems arising in the estimation of the crowd count become apparent. When people walk reasonably close to each other, or occlude each other respective to the camera frame, the foreground blobs corresponding to these people merge together to produce a single blob. This overlap of different foreground targets makes an individual count extremely challenging. The effect is even more detrimental when there are large groups of people walking together, which is often the case in crowded public locations. This is the primary motivation for our work. We will now describe the three different background subtraction schemes used in our system, followed by the actual group count estimation technique.

A. Foreground Detection using Mixtures of Gaussians

One of the foreground detection methods used is the mixture of Gaussians approach of Stauffer and Grimson [8], as modified in Atev *et al.* [9], where the authors provide a practical technique of implementing adaptive background subtraction. According to this method, each pixel can be modeled as a weighted combination of Gaussian distributions. The probability of observing a pixel value c is given by

$$P(c) = \sum_{i=1}^{n} P(M_i) P(c|M_i) \tag{1}$$

where n is the number of component Gaussians, and M_i is the i^{th} component, represented by a mean μ_i and covariance Σ_i.

Full color covariances are used for each Gaussian mixture component, and this is shown to perform well within the frame rate requirements of normal video. The learning rate determines how adaptive the system is. The rules for initialization of the Gaussians and parameter selection for color space, learning rates and thresholds, explained in detail in [9], are adhered to in our implementation. Brightness and contrast equalization techniques described in the paper enhance the performance of the algorithm even under varying illumination and scene conditions. The method is extremely fast and can perform at video frame rate without any approximations in the algorithm.

The major drawback of this method is that, although it can adapt to illumination changes, it can cause merging of temporarily static foreground objects into the background. This is not desirable in crowd–counting applications, since in public places like airports or railway stations it is highly likely that there will be people who remain stationary for extended periods of time (for *e.g.*, waiting in queues).

B. Foreground Detection using Pixel Layering

The second foreground detection method used is the pixel layering technique of Patwardhan *et al.* [10]. This consists of coarsely modeling the scene into a set of layers that represent the background of the image. The probability $P_A(\mathbf{y})$ of a pixel $\mathbf{y}$ belonging to a layer A is computed using non–parametric kernel density estimation with a Gaussian kernel,

$$P_A(\mathbf{y}) = \frac{1}{n_A} \sum_{i=1}^{n_A} \frac{1}{\|\mathbf{H}_A\|^{1/2}} \; \mathbb{K}(\mathbf{H}_A^{-1/2}(\mathbf{y} - \mathbf{x}_{Ai})) \tag{2}$$

where $\mathbb{K}$ represents the Gaussian kernel, with bandwidth matrix $\mathbf{H}_A$, and $\{\mathbf{x}_i\}_{i=1}^{n_A}$ are the pixels forming layer A.

The initial layering of the scene is performed offline, using a set of M training frames ($\sim$ 5–10) from the camera. After the training is completed, the layering is performed online on the new incoming frames, and the pixels which do not correspond to any of the layers surrounding it in a local window are assigned as outliers (foreground pixels). The probability estimate of that pixel belonging to its surrounding layers should exceed a minimum threshold τ to be considered to match that layer. If a pixel exceeds the threshold probability for more than one layer, a maximal likelihood assignment is chosen. Based on the *a–contrario* framework [10], the threshold τ can be computed automatically based on a pre–specified number of false alarms (NFA). The mask of outlier pixels constitutes the foreground pixels and is used by the people counting process.

The layering technique is robust to slight camera motion due to wind or vibrations, and dynamic backgrounds like rippling water and swaying trees. Background memory and temporal persistence of the layers ensure that

a) temporarily static foreground objects are not merged into the background, and b) there are no false alarms when foreground disoccludes the background, as happens with the *MG* method.

However, this method is not adaptive to scene changes such as illumination, compared to the *MG* approach. In our implementation we have modified the algorithm so that it periodically re–learns the background layers after a pre–specified time interval. Since the complete background may not be visible during the re–learning stage, we only renew the knowledge about those regions in the image which are not currently classified as foreground, and which have negligible motion as determined from the optical flow in the set of re–training images.

The continuous learning implemented in the layering technique, although not quite as adaptive as the *MG* approach, provides sufficient stability and robustness for foreground detection in the scene, until the next learning stage. The period of time between successive re–learning steps must be determined by how frequent the scene illumination and other conditions change. For *e.g.*, on a cloudy afternoon in an outdoor scenario, it may be required to perform re–learning much more often than in an indoor scene with controlled artificial lighting.

C. Shadow Removal

Since moving shadows in the image are categorized as foreground by most background subtraction algorithms, the count returned by the system is an overestimate of the actual number of people in the scene. This problem is inherent in any indoor or outdoor scenario. Using the moving shadow detection algorithm of Joshi *et al.* [11], the masks obtained from the foreground detection step can be refined.

In this method, the edge magnitude error E_{mag}, the edge gradient direction error E_{dir}, the intensity ratio I_r, and the color error C_e between the current frame and the background model are used to determine whether a given pixel belongs to the shadow or foreground regions. Given that a pixel belongs to the shadow, the conditional probabilities of these quantities are assumed to be independent.

$$P(E_{mag}, E_{dir}, I_r, C_e|S) = P(E_{mag}|S) \dots P(C_e|S) \quad (3)$$

The component distributions are modeled empirically using exponential and sigmoid functions. Further heuristics are also mentioned in [11] which help in refining the shadow classification procedure.

The improved foreground mask is obtained as those foreground regions which are not classified as shadow by this algorithm. This helps towards obtaining better count estimates from the tracking and counting steps, since the area corresponds more closely to the true count. Shadow regions between blobs which would previously result in merging two individuals are now eliminated. We implement the shadow detection in conjunction with the *MG* method. Fig. 2 shows the shadow detection at work.

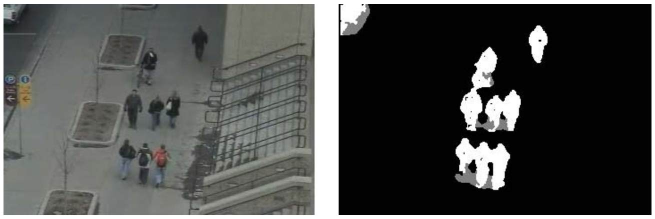


Figure 2. Shadow removal technique from Joshi *et al.* [11] implemented in conjunction with the *MG* method. The grey regions represent those foreground sections determined to be shadows, while the white blobs are actual foreground objects.

D. Estimating Group Counts

The count estimation in this paper has been widely inspired by the work of Kilambi *et al.* [7]. Fig. 3 gives an overview of the system. After obtaining a background–segmented image as in Fig. 1, the different detected blobs are projected onto the ground plane and head plane. The head plane is a horizontal plane roughly at the top of a person's head – it is chosen empirically to be $1.6m$ above the ground plane. This requires that the camera calibration parameters be known and provided as input to the system. The intersection of the two projected surfaces is computed, and an ellipse is fitted to this surface. The projection of the detected foreground onto the different planes, and the intersecting area thus obtained, is depicted in Fig. 4.

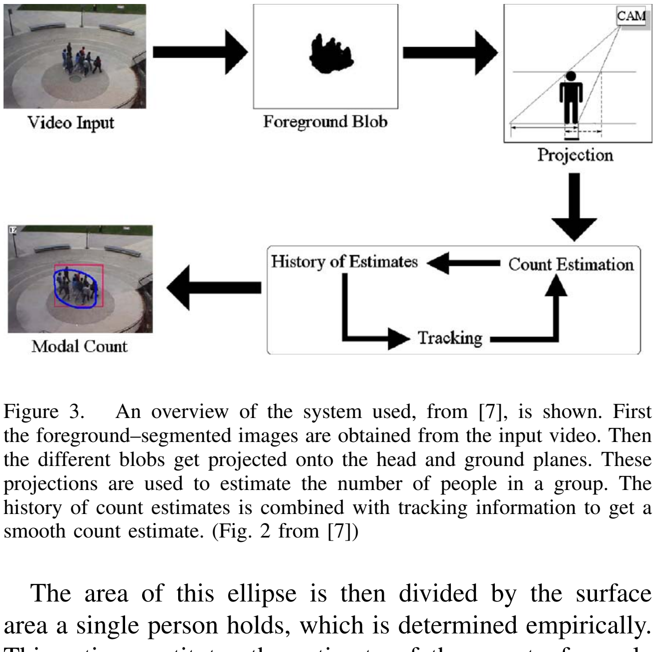


Figure 3. An overview of the system used, from [7], is shown. First the foreground–segmented images are obtained from the input video. Then the different blobs get projected onto the head and ground planes. These projections are used to estimate the number of people in a group. The history of count estimates is combined with tracking information to get a smooth count estimate. (Fig. 2 from [7])

The area of this ellipse is then divided by the surface area a single person holds, which is determined empirically. This ratio constitutes the estimate of the count of people in that group. The Extended Kalman Filter (EKF) is used for tracking of the different individual and group blobs.

If the velocities of any of the blobs exceeds and stays above a certain threshold, they are classified as vehicles and not included in the count estimate of the scene. This is especially important when trying to count pedestrians in outdoor environments (for *e.g.*, bus stops).

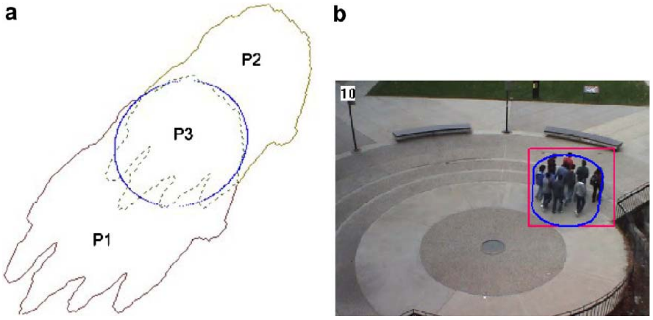


Figure 4. The projections of the blob onto the ground plane (P1) and the head plane (P2) are shown. An ellipse is fit to the region of intersection (P3) of these projected surfaces, the area of which gives the estimate of the number of people in the group. (Fig. 6 from [7])

For the EKF tracker associated with a blob representing a group of people, the count of people in that blob is also added as auxiliary information to the tracker. The history of estimates maintained by the tracker is useful for obtaining a smooth count estimate over the period of time the blob is visible. The mode of history of estimates for a blob is used as the smoothed estimate.

IV. Experimental Results

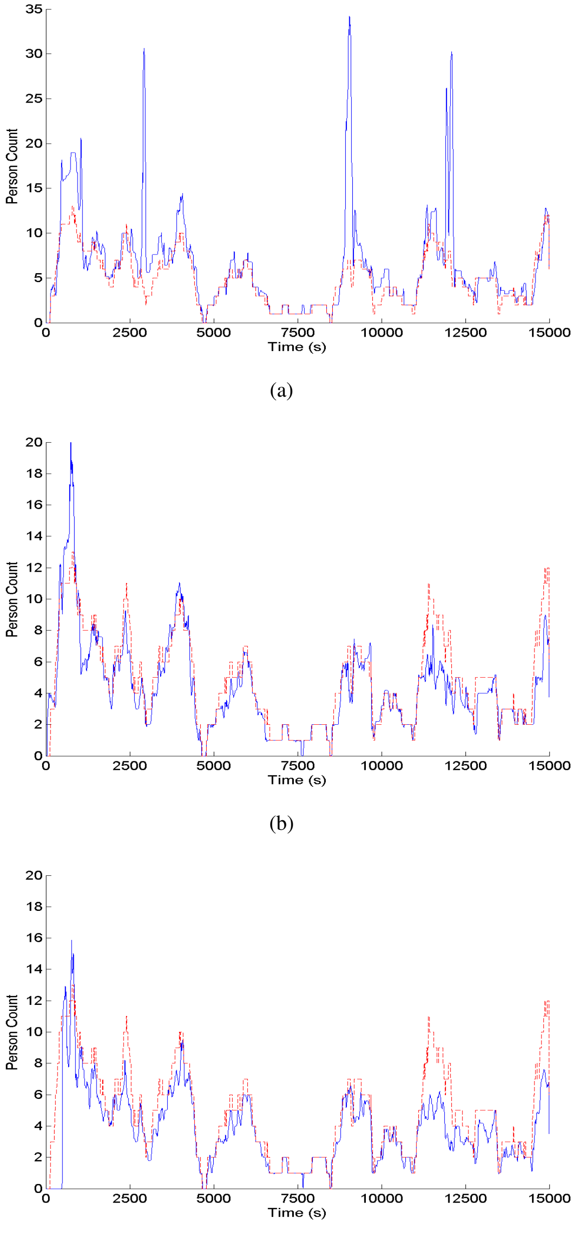
We present a comparison of the group count estimates using three different background subtraction techniques – *a)* the *MG* method without shadow removal, *b)* the *MG* method with shadow removal, and *c)* the *layering* method – on the *Moos Tower* video and two videos from the PETS 2007 benchmark database (Dataset S0) [12]. The error in the group count estimate is computed as a *root–mean–squared* (RMS) error between the estimate and the ground truth, as given by:

$$E_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(t_i - c_i)^2} \qquad (4)$$

where N is the total number of frames, and t_i and c_i are respectively the ground truth and estimated count at frame i.

The group count estimates for the *Moos Tower* video are presented in Fig. 5. Since there is a continuous flow of people in this scene (*i.e.* without too many instances of people remaining stationary for long periods of time), the *MG* method with shadow removal produces the best results in terms of the RMS error with respect to ground truth.

The sudden spikes in the count with the *MG* method without shadow removal in Fig. 5(a), is due to the fact that the scene is right next to a bus stop, and therefore there are slow–moving buses (and their huge shadows) which could not be discarded by the velocity thresholding of the tracker. In a practical outdoor set–up, one could always mark the region of interest where people are expected to be present, cutting off the interference due to on–road vehicles.



(a)

(b)

(c)

Figure 5. Estimated person count for the entire *Moos Tower* video (500 seconds at a frame rate of 30 fps). (a) *MG* background subtraction method without shadow removal – $E_{rms} = 4.14$. (b) *MG* background subtraction method with shadow removal – $E_{rms} = 1.51$. (c) *layering* approach – $E_{rms} = 1.88$. The dashed red line shows the ground truth, while the solid blue line shows the estimate.

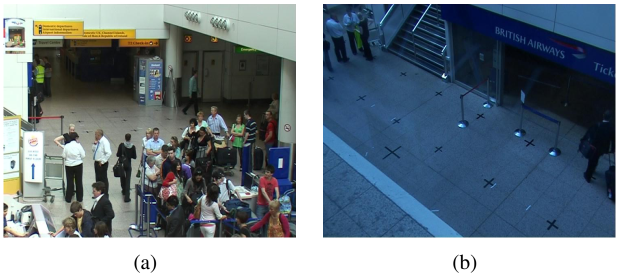


Figure 6. First and third camera views of the airport scene from the PETS 2007 database (Dataset S0) [12].

Method	*Moos Tower*	PETS View 1	PETS View 3
MG1	4.14	23.09	2.61
MG2	1.51	20.16	1.67
Layering	1.88	32.88	2.09

Table I
COMPARISON OF RMS ERROR FOR THE THREE VIDEOS AND THREE DIFFERENT BACKGROUND SUBTRACTION SCHEMES. *MG1* REFERS TO THE *MG* METHOD WITHOUT SHADOW REMOVAL AND *MG2* REFERS TO THE *MG* METHOD WITH SHADOW REMOVAL.

The *layering* method is pretty robust to shadow effects, and therefore performs very close to the *MG* method with explicit shadow removal.

We also present the group counting estimates using the first and third camera views from the PETS 2007 benchmark database [12], shown in Figs. 6(a) and 6(b) respectively. Due to lack of space, we show only the plots for the *MG* method with shadow removal in each view. Table I shows the *RMS* errors for all three different background subtraction techniques used, for each of the three videos. All results are averaged over 5 trials.

In the first camera view Fig. 6(a), the system performed poorly due to the following reasons. Since the camera is looking at a crowd of people waiting in an stationary queue, these people simply "merge" into the background when using the *MG* technique. Also, there is no suitable training frame that allows for layering the background in the video - in fact, throughout the video the complete background is never visible. Hence the layering mechanism is unable to obtain a layered background estimate. Despite this setback that a major portion of the people who are stationary in the scene are discounted as background, the moving population in the scene is captured well. This difference is reflected in the count estimate in Fig. 7(a) as well as the RMS error shown in Table I.

The system performs very well in the third camera view PETS video Fig. 6(b), except for the fact that there are two or three people standing in a near–stationary fashion for a major portion of the video. Hence the system had assumed them to be background, as they exhibit very less motion.

It is important to note here the effects of calibration and intelligent camera placement in surveillance scenarios. Consider the scenes shown in Figs. 8(a) and 8(b). In these videos the camera is placed at a very shallow angle, and therefore there a lot of occlusions. In such scenarios even a human observer will find it difficult to come up with an accurate estimate. Reflections on the ground in both scenes constitute a challenge for any of the methods used, and shadow removal does not help either.

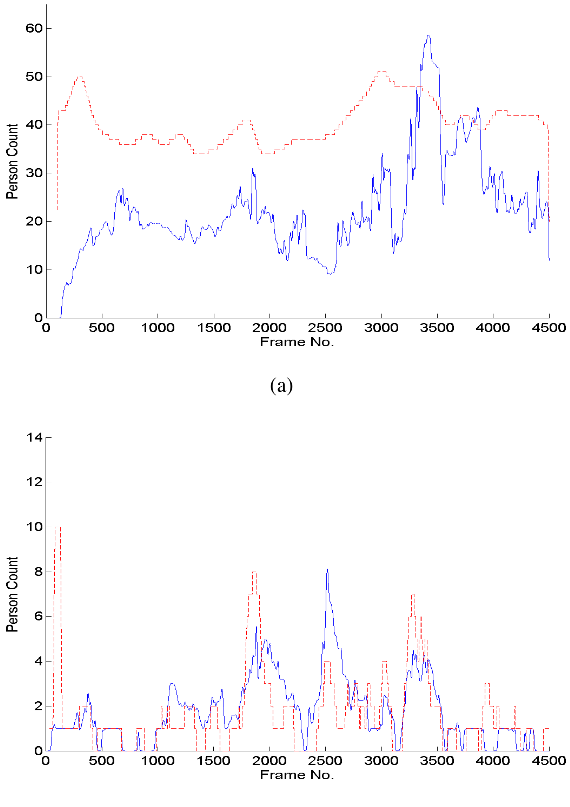
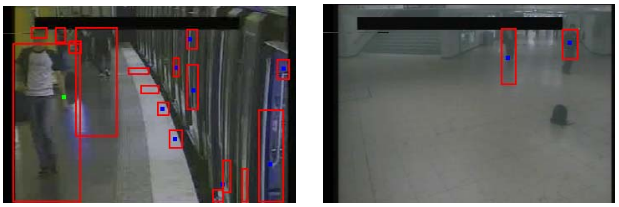


Figure 7. Estimated person count using the *MG* background subtraction method with shadow removal for the (a) first ($E_{rms} = 20.16$) and (b) third ($E_{rms} = 1.67$) camera views of the PETS 2007 videos. Each video is 150 seconds long with a frame rate of 30 fps. The dashed red line shows the ground truth, while the solid blue line shows the estimate.

Another major issue with shallow cameras is the fact that their calibration plays a major role in the system. Since the vanishing point comes very close to the head plane, the projection of the blob onto this plane is very big, and a tiny variation in the blob's size will cause a drastic change in the projected area and thus in the count estimate. However well the background subtraction performs in these scenarios, the count estimation will fail.

V. CONCLUSIONS & FUTURE WORK

We have presented a practical system for the problem of estimating the count of people, especially suited to crowd



(a) (b)

Figure 8. Videos from the ETISEO database [13]. (a) shows a poor view taken from the ETISEO database shot in a subway station. (b) shows a similar view.

counting in group scenarios. The group count estimation greatly depends on the quality of the background subtraction. The better the foreground masks returned, the more accurate the count estimates are. The experimental results obtained clearly support this argument. If the foreground blobs represent the foreground objects of interest well, the system returns a much better count estimate of the people in the image. When we are unable to give a good background subtraction (due to the nature of the scene) as input to the count estimation, it is difficult to obtain a good count estimate.

Most of our future work will revolve on enhancing the background subtraction. The major issue faced in the *MG* method, where temporarily static objects "disappear" into the background, is absent when using the layering technique. Shadow removal improves quality of the foreground masks returned. The continuous re–learning of the background by the layering algorithm makes it adaptive to the scene conditions such as illumination changes. However it still needs an empty frame during initialization for identifying the background, but this is the case for almost any background subtraction algorithm.

Combining the shadow removal algorithm with the *layering* technique will be looked into. We will be working toward developing a hybrid approach combining the best of qualities in both these techniques, which will greatly help not only in people–counting applications but also in many computer vision tasks. The use of convex hulls in the count estimation step will be investigated as to its validity in providing an accurate person count from the blobs. Future work will also revolve around developing more noise–tolerant crowd–counting techniques.

ACKNOWLEDGMENTS

This work was supported by the U.S. ARMY (ARO) through contract #W911NF-08-1-0463 (Proposal 55111-CI), Johnson Controls Inc., the National Science Foundation through Grants #IIS-0219863 and #IIP-0443945, and the Minnesota Department of Transportation.

REFERENCES

[1] K. Terada, D. Yoshida, S. Oe, and J. Yamaguchi, "A method of counting the passing people by using the stereo images," *Proceedings of the International Conference on Image Processing, 1999. ICIP 99.*, vol. 2, pp. 338–342 vol.2, 1999.

[2] S. Velipasalar, Y.-L. Tian, and A. Hampapur, "Automatic counting of interacting people by using a single uncalibrated camera," *IEEE International Conference on Multimedia and Expo, 2006*, pp. 1265–1268, July 2006.

[3] T. Zhao and R. Nevatia, "Bayesian human segmentation in crowded situations," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003.*, vol. 2, pp. II–459–66 vol.2, June 2003.

[4] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004.*, vol. 2, pp. II–406–II–413 Vol.2, June-2 July 2004.

[5] V. Rabaud and S. Belongie, "Counting crowded moving objects," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006*, vol. 1, pp. 705–711, June 2006.

[6] D. Kong, D. Gray, and H. Tao, "Counting pedestrians in crowds using viewpoint invariant training," in *18th International Conference on Pattern Recognition, ICPR*, pp. 1187–1190, 2006.

[7] P. Kilambi, E. Ribnick, A. J. Joshi, O. Masoud, and N. Papanikolopoulos, "Estimating pedestrian counts in groups," *Computer Vision and Image Understanding*, vol. 110, no. 1, pp. 43 – 59, 2008.

[8] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.*, vol. 2, pp. –252 Vol. 2, 1999.

[9] S. Atev, O. Masoud, and N. Papanikolopoulos, "Practical mixtures of Gaussians with brightness monitoring," *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems, 2004.*, pp. 423–428, Oct. 2004.

[10] K. Patwardhan, G. Sapiro, and V. Morellas, "Robust foreground detection in video using pixel layers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 746–751, April 2008.

[11] A. Joshi, S. Atev, O. Masoud, and N. Papanikolopoulos, "Moving shadow detection with low- and mid-level reasoning," *IEEE International Conference on Robotics and Automation, 2007*, pp. 4827–4832, April 2007.

[12] J. M. Ferryman, "PETS 2007 Video Database," in *Proceedings of the Tenth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance PETS*, pp. 1–103, Oct 2007.

[13] A.-T. Nghiem, F. Bremond, M. Thonnat, and V. Valentin, "ETISEO, Performance Evaluation for Video Surveillance Systems," in *Proceedings of AVSS*, Sep 2007.