

Crowd Flow Estimation Using Multiple Visual Features For Scenes With Changing Crowd Densities

Satyam Srivastava, Ka Ki Ng, and Edward J. Delp
Video and Image Processing Laboratory (VIPER)
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana, USA

Abstract

Crowd estimation and monitoring is an important surveillance task. We address the problem of estimating the “flow,” that is the number of persons passing a designated region in a unit time. We designate an area of the scene as a virtual trip wire and accumulate the total number of foreground pixels (in the trip wire) over a chosen time period. We show that cumulative pixel count is related to the number of persons passing through the trip-wire by a scale factor. This scale factor is highly sensitive to the “crowdedness” (levels of crowd density) of the scene which creates different levels of occlusion of the individuals walking/passing through the trip-wire. We use texture features to determine the crowdedness and choose the most appropriate scaling factor. Our method does not require detection and tracking of individuals and is robust to scene dynamics, background subtraction errors, and different crowd levels.

1. Introduction

Crowd monitoring and behavior understanding using visual methods is important in many surveillance applications. A number of methods for crowd analysis have been proposed in the literature [1]. Some of these methods involve explicit detection, tracking, and monitoring of individuals in the scene such as the use of histogram of oriented gradients (HOG) features for person detection [2]. Indirect methods establish a relation between low level image features and crowd attributes. Marana *et al.* [3] show that different crowd density levels can be represented by different texture features. For example, images of low density crowd tend to resemble coarse textures; likewise, images of high

density crowd tend to resemble fine textures. Ma *et al.* [4] determine the crowd density based on a linear relationship between the number of foreground pixels and number of persons. The method takes into account geometrical distortion from the ground plane to the image plane. It is assumed that the number of foreground pixels is proportional to the number of persons, which is only true when there are not serious occlusions between people. Kong *et al.* [5] use edge orientation and blob size histograms as feature, the relationship between the features and the number of people is related using a linear model. Kim *et al.* [6, 7] use number of foreground pixels and motion vectors to find the number of people passing through a gate. Feature normalization is used to make the method viewpoint invariant and robust to different moving speeds of pedestrians. Chan *et al.* [8] segment crowds with dynamic texture motion models and estimate pedestrian count using several geometric, edge, and texture features. They also point out that such indirect methods are better at preserving privacy of the observed individuals.

In this paper we address the problem of accurate estimation of crowd flow using a single view. This analysis can help determine the average pedestrian traffic at a point of interest, detect important ingress/egress points and possible bottlenecks, and also detect panic situations when the crowd motion patterns become anomalous. Our work extends the related work in crowd density [3] and flow estimation [6, 7]. There are two main new concepts discussed in our method – we use two unrelated features (foreground pixels and texture) to make the analysis more robust and we describe a complete process of training and deploying the system. The process highlights the role of human user input for training which allows it to be flexible and work seamlessly even when crowd densities change significantly. We test our methods with crowd videos from publicly available datasets and obtain good accuracies.

This paper is based upon work supported by the U.S. Department of Homeland Security’s VACCINE Center under Award Number 2009-ST-061-C10001. Address all correspondence to E. J. Delp (ace@ecn.purdue.edu).

2. Flow Estimation by Pixel Accumulation

As stated earlier, our goal is to estimate the number of persons crossing a chosen area in a given time interval. This is referred to as “flow,” and the designated area as a virtual “trip wire.” A direct approach would consist of detecting and tracking every individual in the scene (at least those near the trip wire) and updating the count when any such tracked individual crosses the trip wire. However, robust simultaneous tracking of potentially hundreds of targets is not practical.

We use an indirect approach for flow estimation with the working hypothesis that certain low level image features are closely related to the presence and density of people in the scene. Specifically, our approach accumulates the number of foreground pixels in the trip wire region over a period of time. This cumulative pixel count is then scaled to produce an estimate of the flow. This scaling factor is the typical number of pixels accumulated when one person crosses the trip wire.

Our approach is based on the weighted pixel counting method described in [6]. In order to estimate the flow (persons per second), we consider a set of consecutive frames of duration T seconds. The frames in this set are represented as

$$F_{n_i} = \{f_{n_i}(x, y) | x = 0, 1, \dots, W - 1 \quad (1)$$

$$\text{and } y = 0, 1, \dots, H - 1\}. \quad (2)$$

Here W and H are the frame width and height, respectively, and n_i is the frame number with $i = 0, 1, \dots, N - 1$ where N is the total number of frames in the segment (related to T through the frame rate). Finally, let the trip wire be represented as a set $\mathfrak{R} = \{f(x, y) | (x, y) \in \text{trip wire}\}$.

To do flow estimation using pixel accumulation, we consider the RGB color of a frame at (x, y) to determine the foreground pixels. Background subtraction is a widely researched topic in image analysis [9]. We use a low complexity adaptive background subtraction technique [10] to classify the pixels in \mathfrak{R} . In this technique the RGB background model is constructed progressively and the decision threshold is determined from the scene statistics. Note that the computational cost of background subtraction in only the trip wire region is much smaller than the cost for the entire frame. We can represent the foreground mask by an indicator function $I(x, y)$. Figure 1 shows an example of a video frame and the associated foreground mask in the trip wire region (indicated by red lines).

The pixel count in region \mathfrak{R} is described as follows:

$$S_T = \sum_{i=0}^{N-1} \sum_{x, y \in \mathfrak{R}} I_{n_i}(x, y). \quad (3)$$

Note that some weighting scheme is needed to account for the effects of perspective, velocities, and direction of mo-

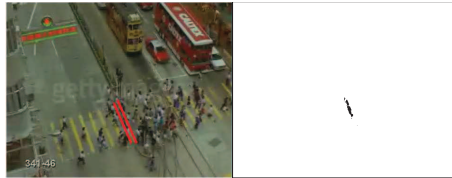


Figure 1. An example of foreground mask generation by background subtraction.

tion with respect to the normal to the trip wire. Methods to compensate for these effects were described in [6]. Here, we assume that the combined effect of these methods can be represented by a single weight factor $w_{n_i}(x, y)$. Thus, the weighted pixel summation is given by:

$$\tilde{S}_T = \sum_{i=0}^{N-1} \sum_{x, y \in \mathfrak{R}} I_{n_i}(x, y) \times w_{n_i}(x, y). \quad (4)$$

The second step in the estimation process involves scaling the *weighted* pixel count \tilde{S}_T by the number of weighted pixels obtained when an individual crosses the trip wire. We represent this quantity by C . The final estimate of the number of persons crossing the trip wire in time T is:

$$\nu_T = \tilde{S}_T / C. \quad (5)$$

It is important to note that the scaling factor C is highly dependent on the sequence (size of individuals) and “crowdedness” of the scene. Crowdedness is related to the crowd density level, that is the number of persons present in the scene. A scene of high crowd density level has more occlusions with people. Thus, the factor C is obtained by a training process which is described in detail in Section 4. Further, multiple such factors are computed for a sequence and the most appropriate is automatically selected based on the estimated crowdedness. In [6] such variation was not considered resulting in a single C for a sequence. Such approach would not be accurate in cases with occlusion. Later, an improvement was suggested to use foreground pixel count (in the full frame) for estimating the crowd density levels [7].

In this paper, we propose the use of texture features to determine the crowd density level. This approach is better than the one using foreground pixel counting [7] because texture features are more robust to small variations in the background and lighting. We estimate the texture features in a subset of the frame known as the region of interest (ROI). We next describe the computation of texture features (Section 3) used for density estimation. These features are associated with different levels of crowd densities through a training process described in Section 4.

3. Crowd Density Level Estimation with Texture Features

Crowd density level is closely related to the amount of occlusion of persons (by each other). In a flow estimation problem this becomes important because the number of pixels accumulated over a period of time does not scale linearly with the number of persons in presence of occlusion. Typically, an individual contributes more foreground pixels in a sparsely crowded scene than in a densely crowded scene. Therefore, we propose using different scaling factors (C in Equation 5) according to crowd density.

Our approach to crowd density estimation is related to the work by Marana *et al.* [3]. This approach is based on the hypothesis that the presence and density of persons in a scene changes the texture of the scene. More specifically, a crowded scene typically results in a finely textured image. In contrast, the image of the same scene with fewer persons would be more coarse-textured or untextured. We characterize the texture content of the image with features extracted from the gray level co-occurrence matrix (GLCM).

The GLCM is a statistical technique for characterizing the texture of an image by counting the number of times spatially adjacent pixels occur in the image with specified intensity values [11]. We construct the GLCM by considering pixel pairs differing by 1 intensity level. Further, we repeat the process for four spatial orientations corresponding to right (0°), top-right (45°), top (90°), and top-left (135°) neighbors. For each of the four matrices M_1, M_2, M_3, M_4 we obtain four statistical features defined in [11] – energy, entropy, homogeneity, and contrast. We concatenate these features to represent the texture by a single vector τ in a 16 dimensional feature space.

Consider a video sequence in which crowd densities occur at k levels such as “sparse,” “low,” and “very high.” We obtain the feature vectors $\tau_1, \tau_2, \dots, \tau_k$ through a training process detailed in Section 4. While classifying the density level for a test frame, we estimate the distance between the test frame’s feature vector τ_{test} and the k reference vectors. The density level is then determined by a nearest neighbor rule. Let l represent the crowd density level for a given frame. Then,

$$l = \arg \min_i d(\tau_{test}, \tau_i), \quad (6)$$

where $d(\cdot)$ is the distance function and $i \in \{1, 2, \dots, k\}$. We use the Euclidean distance after normalizing the values of each feature to approximately¹ $[0.0, 1.0]$. The normalization is achieved using the maximum and minimum values of the features from the training frames. Since the scaling factor depends primarily on the crowd density near the trip

¹The goal of this scaling process is to make the 16 dimensions comparable rather than enforce probability-like constraints. Thus, some values may lie outside of the range which is acceptable.



Figure 2. An example of the trip-wire and ROI.

wire, we estimate the texture features only in a subset G of the frame such that $G = \{f(x, y) | (x, y) \in \text{ROI}\}$. This region of interest (ROI) is specified during the training process and contains the trip wire ($\mathfrak{R} \subset G$).

Note that for a sequence with k crowd density levels, there should be k scaling factors C_1, C_2, \dots, C_k to be used for flow estimation. If the density level for frame n_i is $l(n_i)$, then the final flow estimation is obtained by combining the steps in Equation 4 and Equation 5 as follows:

$$\nu_T = \sum_{i=0}^{N-1} \sum_{x, y \in \mathfrak{R}} \frac{I_{n_i}(x, y) \times w_{n_i}(x, y)}{C_{l(n_i)}}. \quad (7)$$

4. Experimental Setup

For each test sequence, a trip-wire \mathfrak{R} and a region of interest G are specified with graphical user input. Training with user input is required for two purposes – 1) to associate the texture features extracted from the GLCM of the ROI with different levels of crowd density and 2) to associate the number of foreground pixels on the trip-wire with the number of persons for every crowd density level. The ROI is chosen such that the trip-wire is bounded by it. An example of the trip-wire and the ROI is shown in Figure 2.

4.1. Training of Texture Features

To associate the texture features extracted from the GLCM of the ROI with different levels of crowd density, a training sequence is used and T frames are randomly selected from it. The ROI from each of the T frames is displayed to the user who then assigns a label l to the frame which serves as the ground truth. The ROI of each labeled frame is used to determine the GLCM and the 16-dimensional texture feature vectors t_{n_i} (where n_i is the index of the i^{th} training frame and $i = 1, 2, \dots, T$) are extracted from each GLCM. After all T feature vectors are determined, the average of the set of feature vectors associated with the j^{th} texture class is used as the “model” of the

j^{th} density level:

$$\tau_j = \frac{\sum_{i=1}^T t_{n_i} \times \delta[l(n_i) - j]}{\sum_{i=1}^T \delta[l(n_i) - j]}. \quad (8)$$

Here $\delta(x)$ represents an indicator function (or a Kronecker delta function) which is zero everywhere except at $x = 0$ where it takes unit value.

4.2. Training of Foreground Pixel Counts

After constructing the texture models for different crowd density levels, the crowd density level of each frame in the entire testing sequence is determined using the models according to Equation 6. A plot of the result of this step is shown in Figure 3. This plot also serves as an illustration for the training of foreground pixel counts. A “stable” period of the plot is chosen for foreground pixel counts training of each texture class. A set of consecutive frames is considered “stable” if they all have the same texture classification.

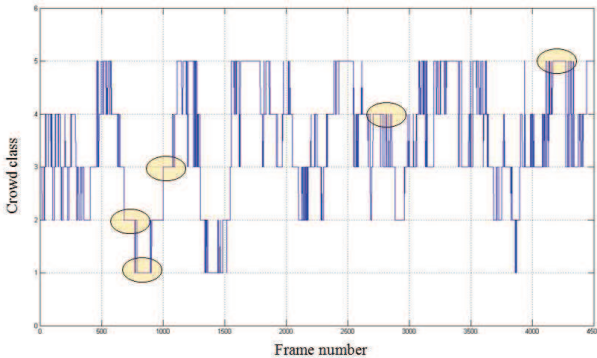


Figure 3. A plot of a five-class texture classification on a test sequence. The “stable” periods are circled.

Background subtraction is used to determine the foreground regions and this process is done in the trip-wire region only. Hence the method has a very low computational cost. The accumulative foreground pixels \tilde{S}_T are counted for the entire “stable” period. Simultaneously, the frames are displayed to the user who counts the number of persons ν_T crossing the trip wire (which can be fractional). These values are used to estimate the scaling constant C_l for each texture class (or crowd density level) l .

4.3. Testing: Finding the number of Persons

Given a frame of a testing sequence the GLCM of the ROI is first determined and the texture features are extracted from it. The crowd density level is then determined using the models of different texture classes from the training process. Given a crowd density level, the number of foreground

pixels on the trip-wire is determined using background subtraction. Finally, the number of persons is determined according to Equation 7. This process is illustrated in Figure 4.

5. Results and Discussion

The methods described above were tested surveillance videos most of which are obtained from publicly available datasets [12, 13]. This UCF data was used by Ali and Shah to test their methods for tracking subjects in crowded scenes [14] while the UCSD dataset is designed for detecting anomalies in pedestrian crowds [15]. We present the results in two parts. First, we show the effectiveness of texture based classification for crowd density level estimation. We then present the final output of our flow estimation system for several sequences.

Figure 5 shows examples of video frames of different crowd density levels from two test sequences. The density levels (estimated and ground truth) are provided in the captions. Note that bigger labels are used to represent higher density crowds. Also note that the crowdedness levels are decided relative to each sequence. Thus, the class 2 frames of the two sequences have very different density of persons.

The results of final flow estimation are presented in Table 1. We provide the length of the test sequence, the number of persons crossing the trip wire (estimated and ground truth), and the number of crowd density levels seen in the duration of the test. In most cases the estimate is close to the true flow irrespective of the change in crowd density levels. Further, there is no systematic over- or under-estimation in our method.

Table 1. Testing results for crowd flow estimation.

Seq	Frames	Estim Flow	True Flow	Density Levels
1	200	16	19	1,2,3,4,5
2	100	1.0	1	1,2
3	100	7.9	6.5	1,2,3
4	200	25.5	24	1,2,3,4
5	110	9.1	8.5	2,3,4
6	110	4.7	5	1,2

The results above are tested on sequences between 4 and 20 seconds in length. While our method would work equally well on longer sequences, it has been designed to be particularly useful in real-time surveillance situations where instantaneous flow might be more important than an average estimated over a long time. Our test sequences also capture the difficult cases where crowd density fluctuates heavily (as shown in column 5 of Table 1).

It should be noted that the accuracy of the methods is sensitive to the training of texture models. More specif-

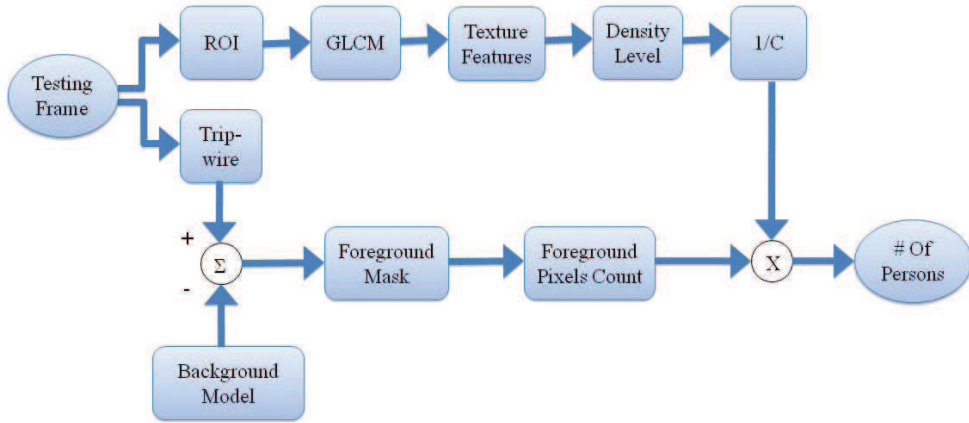


Figure 4. A block diagram illustrating the testing process.



Figure 5. Examples of crowd density estimation on three video sequences. The estimated and true (in parentheses) density levels are (left to right) – Top: 3(3), 1(1), 2(2), Middle: 1(1), 3(3), 4(4), Bottom: 2(2), 3(3), 1(1).

ically, the random selection of frames affects how well the texture models represent each class. We observe that choosing more number of crowd density levels can result in higher accuracy in flow estimation but the gain in accuracy becomes smaller with very large number of levels. Also the ROI is selected such that it can represent the texture of the local density near the trip wire. It should be large enough to contain multiple persons around the trip wire. However, a very large ROI would incur unwanted influence from persons far from the trip wire and would increase computa-

tional cost.

6. Conclusion

In this paper, we present a robust method for crowd flow estimation that counts the number of persons passing through a trip-wire. The method relates the cumulative foreground pixel counts to the number of persons in a crowd scene through a scaling factor. This scaling factor depends on the local texture features that takes into account the level of occlusions. We test our method using publicly available

dataset and experimental results show the effectiveness of our method even when the crowd density levels vary frequently. Future extensions of our work include finding the optimal size of the trip-wire and ROI, the number of texture levels, and using more robust background subtraction methods.

References

- [1] B. Zhan, D. Monekosso, P. Remagnino, S. Velastin, and L. Xu, "Crowd analysis: A survey," *Machine Vision and Applications*, vol. 19, no. 5, pp. 345–357, 2008.
- [2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, California, June 2005, pp. 886–893.
- [3] A. N. Marana, S. A. Velastin, L. F. Costa, and R. A. Lotufo, "Automatic estimation of crowd density using texture," *Proceedings of the International Workshop on Systems and Image Processing*, Poland, May 1997.
- [4] R. Ma, L. Li, W. Huang, and Q. Tian, "On pixel count based crowd density estimation for visual surveillance," *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, Singapore, December 2004, pp. 170–173.
- [5] D. Kong, D. Gray, and H. Tao, "Counting pedestrians in crowds using viewpoint invariant training," *Proceedings of the British Machine Vision Conference*, Oxford, UK, September 2005.
- [6] G. G. Lee, B. S. Kim, and W. Y. Kim, "Automatic estimation of pedestrian flow," *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*, Vienna, Austria, September 2007, pp. 291–296.
- [7] B. S. Kim, G. G. Lee, J. Y. Yoon, J. J. Kim, and W. Y. Kim, "A method for counting pedestrians in crowded scenes," *Proceedings of the International Conference on Intelligent Computing*, Shanghai, China, September 2008, pp. 1117–1126.
- [8] A. B. Chan, Z. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008, pp. 1–7.
- [9] M. Piccardi, "Background subtraction techniques: A review," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, The Hague, Netherlands, October 2004, pp. 3099–3104.
- [10] K. K. Ng and E. J. Delp, "Object tracking initialization using automatic moving object detection," *Proceedings of SPIE/IS&T Electronic Imaging: Visual Information Processing and Communication*, vol. 7543, San Jose, California, January 2010.
- [11] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, May 1979.
- [12] University of Central Florida, Computer Vision Lab, "Data Sets and Test Data," Online: <http://server.cs.ucf.edu/~vision/data.html>, Accessed: February 2011.
- [13] University of California, San Diego, Statistical Visual Computing Lab, "UCSD Anomaly Detection Dataset," Online: <http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm>, Accessed: February 2011.
- [14] S. Ali and M. Shah, "Floor fields for tracking in high density crowd scenes," *Proceedings of the European Conference on Computer Vision*, Marseille, France, October 2008, pp. 1–14.
- [15] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, California, June 2010, pp. 1975–1981.