Privacy Preserving Crowd Monitoring: Counting People without People Models or Tracking

Antoni B. Chan Zhang-Sheng John Liang Nuno Vasconcelos
Electrical and Computer Engineering Department
University of California, San Diego
abchan@ucsd.edu, jliangzs@stanford.edu, nuno@ece.ucsd.edu

Abstract

We present a privacy-preserving system for estimating the size of inhomogeneous crowds, composed of pedestrians that travel in different directions, without using explicit object segmentation or tracking. First, the crowd is segmented into components of homogeneous motion, using the mixture of dynamic textures motion model. Second, a set of simple holistic features is extracted from each segmented region, and the correspondence between features and the number of people per segment is learned with Gaussian Process regression. We validate both the crowd segmentation algorithm, and the crowd counting system, on a large pedestrian dataset (2000 frames of video, containing 49,885 total pedestrian instances). Finally, we present results of the system running on a full hour of video.

1. Introduction

There is currently a great interest in vision technology for monitoring all types of environments. This could have many goals, *e.g*. security, resource management, or advertising. Yet, the deployment of vision technology is invariably met with skepticism by society at large, given the perception that it could be used to infringe on the individuals' privacy rights. This tension is common in all areas of data-mining [1, 2], but becomes an especially acute problem for computer vision for two reasons: 1) the perception of compromised privacy is particularly strong for technology which, by default, keeps a visual record of people's actions; 2) the current approaches to vision-based monitoring are usually based on object tracking or image primitives, such as object silhouettes or blobs, which imply some attempt to "identify" or "single out" the individual.

From the laymen's point of view, there are many problems in environment monitoring that can be solved without explicit tracking of individuals. These are problems where all the information required to perform the task can be gathered by analyzing the environment *holistically*: *e.g*. monitoring of traffic flows, detection of disturbances in public spaces, detection of speeding on highways, or estimation of the size of moving crowds. By definition, these tasks are based on either properties of 1) the "crowd" as a whole, or 2) an individual's "deviation" from the crowd. In both cases, to accomplish the task it should suffice to build good *models for the patterns of crowd behavior*. Events could then be detected as *variations in these patterns*, and abnormal individual actions could be detected as *outliers* with respect to the crowd behavior. This would preserve the individual's identity until there is good reason to do otherwise.

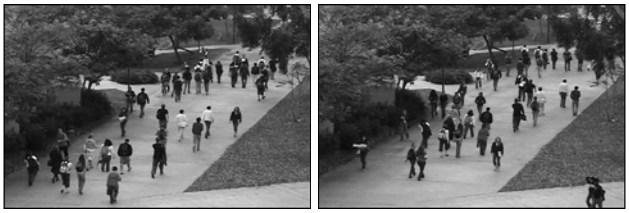

Figure 1. Examples of a low-resolution scene containing a sizable crowd with inhomogeneous dynamics, due to pedestrian motion in different directions.

In this work, we introduce a new formulation for surveillance technology, which is averse to individual tracking and, consequently, privacy preserving. We illustrate this new formulation with the problem of pedestrian counting. This is a canonical example of a problem that vision technology addresses with privacy invasive methods: detect the people in the scene [3, 4, 5, 6, 7], track them over time [8, 9, 10], and count the number of tracks. While a number of methods that do not require explicit detection or tracking have been previously proposed [11, 12, 13, 14, 15, 16, 17], they have not fully established the viability of the privacy-preserving approach. This has a multitude of reasons: from limited applications to indoor environments with controlled lighting (*e.g*. subway platforms) [11, 12, 13, 14, 16]; to ignoring the crowd dynamics (*i.e*. treating people moving in different directions as the same) [11, 12, 13, 14, 15, 17]; to assumptions of homogeneous crowd density (*i.e*. spacing between people) [16]; to measuring a surrogate of the crowd size (*e.g*. crowd density or percent crowding) [11, 12, 16]; to questionable scalability to scenes involving more than a few people [17]; to limited experimental validation of the proposed algorithms [11, 12, 13, 15, 16].

Unlike these proposals, we show that there is in fact no need for pedestrian detection, object tracking, or object-based image primitives to accomplish the pedestrian counting goal, even when the crowd is *sizable and inhomogeneous*, *e.g*. has *sub-components with different dynamics*, as illustrated in Figure 1. In fact, we argue that, when considered under the constraints of privacy-preserving monitoring, the problem actually appears to become simpler. We simply develop methods for segmenting the crowd into the subparts of interest (*e.g*. groups of people moving in different directions) and estimate the number of people by analyzing *holistic* properties of each component. This is shown to be quite robust and accurate.

The contributions of this paper are three-fold. First, we present a privacy-preserving vision system for estimating the size of *inhomogeneous* crowds *that does not depend on object detection or feature tracking*. The system is also privacy-preserving in the sense that *it can be implemented with hardware that does not produce a visual record of the people in the scene*, *i.e*. with special-purpose cameras that output low-level features (*e.g*. segmentations, edges, and texture). Second, we validate the system quantitatively on a large dataset of pedestrian video, containing 49,885 pedestrian instances. Third, we demonstrate its robustness by presenting results on an hour of video. To our knowledge, this is the first privacy-preserving pedestrian counting system that accounts for multiple pedestrian flows, and successfully operates continuously in an outdoors, unconstrained, environment for such time periods. The remainder of the paper is organized as follows. In Section 2 we review related work in crowd counting. In Section 3, we introduce a crowd counting system based on motion segmentation and Gaussian processes. Finally, we present the pedestrian database and experimental results in Sections 4 and 5.

2. Related work

The taxonomy of crowd counting algorithms consists of three paradigms: 1) pedestrian detection, 2) visual feature trajectory clustering, and 3) feature-based regression. Pedestrian detection algorithms are based on boosting appearance and motion features [3], Bayesian model-based segmentation [4], or integrated top-down and bottom-up processing [5]. Because they detect whole pedestrians, these methods tend to suffer in very crowded scenes with significant occlusion, which has been addressed to some extent by adopting part-based detectors [6, 7].

The second paradigm counts people by identifying and tracking visual features over time. The feature trajectories that exhibit coherent motion are clustered, and the number of clusters is the estimate of the number of moving people. Examples of this formulation include [8], which uses the KLT tracker and agglomerative clustering, and [9], which takes an unsupervised Bayesian approach.

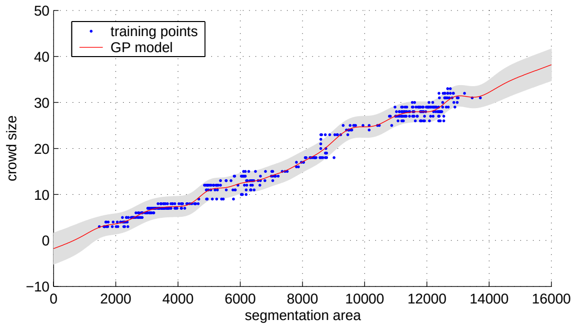

Figure 2. Correspondence between crowd size and segmentation area. The Gaussian process regression is plotted with the two standard-deviations error bars (gray area).

Feature-based regression for crowd counting was first applied to subway platform monitoring. These methods typically work by: 1) subtracting the background; 2) measuring various features of the foreground pixels, such as total area [11, 12, 14], edge count [12, 13, 14], or texture [16]; and 3) estimating the crowd density or crowd count by a regression function, *e.g*. linear [11, 14], piece-wise linear [13], or neural networks [12, 16]. In recent years, feature-based regression has also been applied to outdoor scenes. For example, [15] applies neural networks to the histograms of foreground segment areas and edge orientations. [17] estimates the number of people in each foreground segment by matching its shape to a database containing the silhouettes of possible people configurations, but is only applicable when the number of people in each segment is small (empirically, less than 6).

3. Privacy preserving crowd counting

Figure 1 shows examples of a crowded scene on a pedestrian walkway. The goal of the proposed system is to estimate the number of people moving in each direction, in a privacy-preserving manner. Given a segmentation into the two sub-components of the crowd, it is shown that crowd size can indeed be estimated from low-level features extracted from each crowd segment. For example, as shown in Figure 2, a simple feature such as the segmentation area is approximately linear in the crowd size.

An outline of the crowd counting system appears in Figure 3. The video is segmented into crowd regions moving in different directions, using a mixture of dynamic textures. For each crowd segment, various features are extracted, while applying a perspective map to weight each image location according to its approximate size in the real

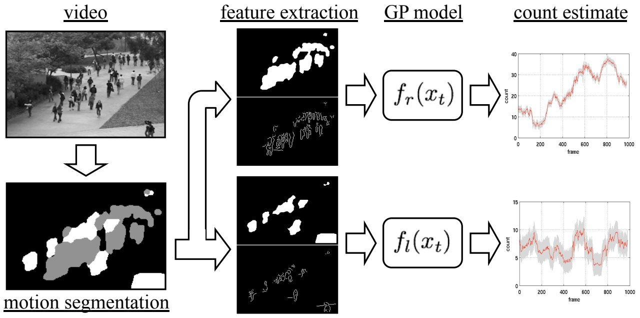

Figure 3. Crowd counting system: the scene is segmented into crowds with different motions. Normalized features that account for perspective are extracted from each segment, and the crowd count for each segment is estimated with a Gaussian process.

scene. Finally, the number of people per segment is estimated with Gaussian process regression. The remainder of this section describes each of these system components.

3.1. Crowd segmentation

We adopt the *mixture of dynamic textures* [18] to segment the crowds moving in different directions. The video is represented as collection of spatio-temporal patches ($7 \times 7 \times 20$ patches in all experiments reported in the paper), which are modeled as independent samples from a mixture of dynamic texture models [19]. The mixture model is learned with the expectation-maximization (EM) algorithm [18]. Video locations are then scanned sequentially, a patch is extracted at each location, and assigned to the mixture component of largest posterior probability. The location is declared to belong to the segmentation region associated with that component. For long sequences, where characteristic motions are not expected to change significantly, the computational cost of the segmentation can be reduced by learning the mixture model from a subset of the video (*e.g.* a representative clip). The remaining video can then be segmented by computing the posterior assignments as before. This procedure tends to work well in practice, and was used in this paper to segment a full hour of video. The resulting segmentations are illustrated in Figures 9 and 11.

3.2. Perspective normalization

Before extracting features from the video segments, it is important to consider the effects of perspective. Because objects closer to the camera appear larger, any feature extracted from a foreground object will account for a smaller portion of the object than one extracted from an object farther away. This makes it important to normalize the features for perspective. One possibility is to weight each pixel according to a perspective normalization map. The pixel weight is based on the expected depth of the object which generated the pixel, with larger weights given to far objects.

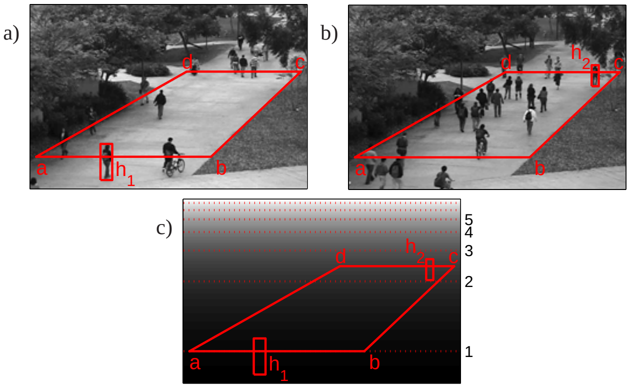

Figure 4. Perspective map: a) reference person at the front of walkway, and b) at the end; c) the perspective map, which scales pixels by their relative size in the true 3d scene.

In this work, we approximate the perspective map by linearly interpolating between the two extremes of the scene. A ground plane is first marked, as in Figure 4a, and the distances $|\overline{ab}|$ and $|\overline{cd}|$ are measured[1]. Next, a reference pedestrian is selected, and the heights h_1 and h_2 are measured when the center of the person is on $\overline{ab}$ and $\overline{cd}$ (see Figures 4a and 4b). The pixels on $\overline{ab}$ are given a weight of 1, and the pixels on $\overline{cd}$ a weight of $\frac{h_1 |\overline{ab}|}{h_2 |\overline{cd}|}$. Finally, the remaining pixel weights are computed by interpolating linearly between the two lines. Figure 4c shows the perspective map of the scene using the above procedure. In this case, objects on the front-line $\overline{ab}$ are approximately 2.4 times bigger than objects on the back-line $\overline{cd}$. Finally, for features based on area (*e.g.* segmentation area), the weights are applied to each pixel. For features based on edges (*e.g.* edge histogram), the square-roots of the weights are used.

3.3. Feature extraction

Ideally, features such as segmentation area or number of edges should vary linearly with the number of people in the scene [14, 11]. Figure 2 plots the segmentation area versus the crowd size. While the overall trend is indeed linear, there exist local non-linearities that arise from a variety of factors, including occlusion, segmentation errors, and pedestrian configuration (*e.g.* spacing within a segment). To model these non-linearities, we extract an additional 28 features from each crowd segment.

Segment features: These features capture segment shape and size.

- *Area* – total number of pixels in the segment.
- *Perimeter* – total number of pixels on the segment perimeter, computed with morphological operators.
- *Perimeter edge orientation* – orientation histogram of the segment perimeter. The orientations are quantized

[1] Here we assume that the horizontal ground plane is parallel to the horizontal axis of the image, but the procedure can be generalized if not.

into 6 bins, and opposite orientations (180° apart) are considered equal. The orientation of each edge pixel is computed by finding the maximum response to a set of oriented Gaussian filters at that point.

- *Perimeter-area ratio* – ratio between the segment perimeter and area, which measures the complexity of the segment shape. Segments of high ratio contain bumps in their perimeter, which may be indicative of the number of people contained within.

Internal edge features: The edges contained in a crowd segment are a strong clue about the number of people in the segment [15, 14]. A Canny edge detector [20] is applied to the entire image, the edge image is masked by the crowd segmentation, and the following features are computed:

- *Total edge pixels* – total number of edge pixels contained in the segment.
- *Edge orientation* – histogram of the edge orientations in the segment, generated in the same way as the perimeter edge histogram (also using 6 bins).
- *Minkowski dimension* – the Minkowski fractal dimension of the edges in the segment, which estimates their degree of "space-filling" (see [21] for more details).

Texture features: Texture features, based on the gray-level co-occurrence matrix (GLCM), were used in [16] to classify image patches into 5 classes of crowd density. In this work, we adopt a similar set of measurements for *counting* the number of pedestrians in each segment. The image is quantized into 8 gray levels, and the 2nd-order joint conditional probability density function $f(i, j|d, \theta)$ is estimated for distance $d = 1$ and angles $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$. The following texture properties are computed:

- *Homogeneity*: $S_h(d, \theta) = \sum_{i,j} \frac{f(i,j|d,\theta)}{1+(i-j)^2}$
- *Energy*: $S_g(d, \theta) = \sum_{i,j} f(i, j|d, \theta)^2$
- *Entropy*: $S_p(d, \theta) = \sum_{i,j} f(i, j|d, \theta) \log f(i, j|d, \theta)$

resulting in a total of 12 texture features.

3.4. Gaussian process regression

A Gaussian process (GP) [22] is used to regress feature vectors to the number of people per segment. The GP defines a distribution over functions, which is "pinned down" at the training points. The classes of functions that the GP can model is dependent on the kernel function used. For the task of pedestrian counting, we note that the dominant trend of many of the features is linear (*e.g*. segment area), with local non-linearities. To capture both trends, we combine the linear and the squared-exponential (RBF) kernels, *i.e*.

$$k(x_p, x_q) = \alpha_1(x_p^T x_q + 1) + \alpha_2 e^{\frac{-\|x_p - x_q\|^2}{\alpha_3}} + \alpha_4\delta(p, q)$$

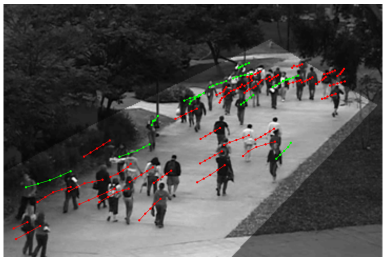

Figure 5. Ground-truth annotations of pedestrian traffic. Red and green tracks are people moving away from, and towards the camera, respectively. The ROI for the experiments is highlighted.

	away	towards	total
# of unique people	108	81	189
max # in frame	33	16	46
min # in frame	3	4	11
total # of people	29,577	20,308	49,885
# of training people	13,047	7,428	20,475
# of test people	16,530	12,880	29,410

Table 1. Properties of the pedestrian database

with hyperparameters $\alpha = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$. The first and second terms of the kernel are the linear and RBF components, while the third term models observation noise.

Figure 2 shows an example of GP regression for segmentation area. Note that the linear component of the kernel captures the dominant trend, while the RBF component models local non-linearities. Finally, while the same feature set is used throughout the system, a different regressor is learned for each direction of crowd motion because the appearance changes with the traveling direction.

4. Pedestrian database

In this section, we describe the pedestrian database used in the experiments. An hour of video was collected from a stationary digital camcorder overlooking a pedestrian walkway at UCSD. The original video was captured at 30 fps with a frame size of 740×480, and was later downsampled to 238×158 and 10 fps. The first 2000 frames (200 seconds) of video were selected for ground-truth annotation, which we will refer to as the pedestrian dataset.

A region-of-interest (ROI) was selected on the walkway (see Figure 5), and the traveling direction ("away from" or "towards" the camera) and visible center of each pedestrian[2] was annotated every five frames. Pedestrian locations in the remaining frames were estimated with linear interpolation. An example annotation is shown in Figure 5. Note that the ground-truth pedestrian locations are not required to train the crowd-counting system, but necessary to test performance. The dataset contains a total of 49,885 pedestrian instances (see Table 1 for a summary). Figure 6 presents the ground-truth pedestrian count over time.

[2] Bicyclists and skateboarders were treated as normal pedestrians.

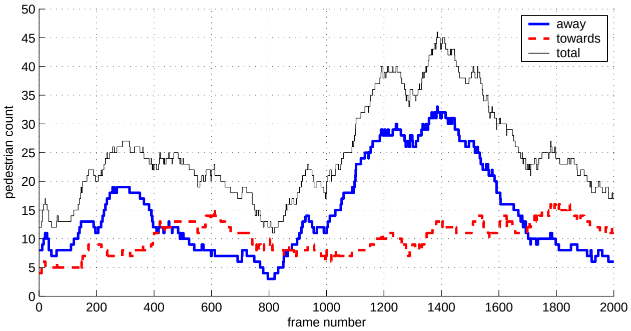

Figure 6. Ground-truth pedestrian count over time. The count is split into pedestrians moving away from, and towards the camera.

The video was split into a training set, for learning the GP, and a test set, for validation. The training set contains 800 frames, between frame 600 and 1399, with the remaining 1200 frames held out for testing. This split tests the ability of the crowd-counting system to *extrapolate* beyond the training set. In contrast, spacing the training set evenly throughout the dataset would only test the ability to *interpolate* between the training points, which provides little insight into generalization ability and robustness. The dataset is available to the vision community [23].

5. Experiments and Discussion

Successful crowd counting depends on effective crowd segmentation. Hence, we first test the segmentation algorithm, and then present crowd counting results.

5.1. Motion segmentation results

The mixture of dynamic textures was used to segment the crowd according to motion: people moving towards the camera, and people moving away. The segmentation was validated with an ROC curve based on the ground-truth pedestrian locations. In each frame, a true positive is recorded if the ground-truth location of a person is within the correct motion segment, and a false positive is recorded otherwise. The true positive and false positive rates (TPR and FPR) are computed over all 2000 frames, and an ROC curve was generated from the TPR and FPR for dilations and erosions of the segmentation with variable size disks.

The ROC curve produced by the mixture of dynamic textures (DTM) is shown in Figure 7. For comparison, the scene was also segmented with normalized cuts and motion-profiles [24], which is denoted by NCuts. At the operating point of the segmentation algorithms (*i.e.* no morphological post-processing), DTM achieves a high TPR of 0.936, at a low FPR of 0.036. NCuts achieves a lower TPR (0.890) at a higher FPR (0.103). In addition, DTM has a larger area under the ROC curve (AROC) than NCuts (0.9727 versus 0.9545). These results validate the DTM as a robust segmentation algorithm for these types of crowded scenes.

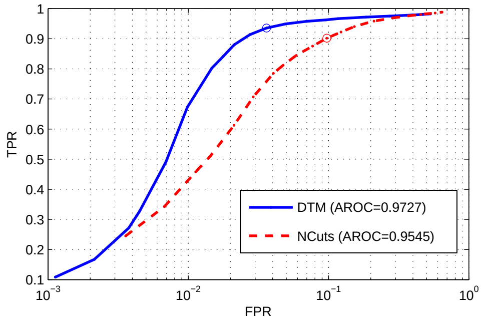

Figure 7. Crowd motion segmentation: ROC curve for the mixture of dynamic textures (DTM) and normalized cuts (NCuts). The circle indicates the operating point of the algorithm.

5.2. Crowd counting results

The crowd counting system was trained on the 800 training frames, and tested on the remaining 1200. The GP output was rounded to the nearest integer to produce a crowd count, and both the mean-squared-error (MSE) and the absolute error between this estimate and the ground-truth were recorded. For comparison, we also trained the system with different subsets of the features: only segment area, the segment features, and the segment and edge features. Finally, we compared performance against the feature sets of [15] (segment size histogram and edge orientation histogram) and [14] (segment area and total edges).

Table 2 shows the error rates for the two crowd directions, under the different feature representations. Using only the area feature performs the worst, with an MSE of $8.155/2.663$ (away/towards), and performance improves steadily as other features are added. Using all the segment features improves the MSE to $6.163/2.147$, and using both the segment and edge features further improves to $5.153/1.987$. Finally, using all features (*i.e.* adding the texture features) performs best, with an MSE of $4.181/1.291$. This demonstrates the informativeness of the different feature subsets: the segment features provide a coarse linear estimate, which is refined by the edge and texture features accounting for various non-linearities. The full feature set also performs better than those of [15, 14]. Finally, using features that do not accounting for perspective, the system performance drops significantly (MSE $6.869/2.541$), indicating that normalization is indeed required.

Figures 8a and 8b show the crowd count estimates (using all features) as a function of time, for the two crowd directions. The estimates track the ground-truth well in most of the test frames. The overestimate of the size of the “away” crowd in frames 180-300 is caused by two bicyclists traveling quickly through the scene, as shown in the second image of Figure 9. Figures 8c and 8d show the cumulative error, *i.e.* the frequency with which the counting error is below a particular number of people. The count is within 3 people of the ground-truth 91% of the time for the “away”

	Away		Towards	
Feature Set	MSE	|error|	MSE	|error|
all features	**4.181**	**1.621**	**1.291**	**0.869**
segm+edge	5.153	1.767	1.987	1.122
segm	6.163	1.894	2.147	1.172
only area	8.155	2.037	2.663	1.307
all (no norm.)	6.869	2.191	2.541	1.321
[15]	5.438	1.808	2.871	1.343
[14]	6.953	1.995	2.131	1.108

Table 2. Crowd counting results: MSE and absolute error on the test set using different feature sets.

crowd, and within 2 people 98% of the time for the "towards" crowd. On average, the count estimate is within 1.62 and 0.87 ("away" and "towards", respectively) of the ground-truth. This suggests that the system is robust and accurate enough for monitoring pedestrian traffic over long time-periods. Finally, Figure 9 shows the original image, segmentation, and crowd estimates for several frames in the test set. A video is also available from [23].

Finally, we trained the GP on the full 2000 frames of the pedestrian dataset, and ran the system on the remaining 50 minutes of captured video. The resulting crowd estimates are shown in Figure 10, while Figure 11 shows several example outputs of the system (the video is also available from [23]). Qualitatively, the system tracks the changes in pedestrian traffic fairly well. Most errors tend to occur when there are very few people (less than two) in the scene. These errors are reasonable, considering that there are no training examples with such few people, and the problem could be fixed by simply adding training examples of such cases. Note that the GP signals its lack of confidence in these estimates, by assigning them larger error bars.

A more challenging set of errors occur when bicycles, skateboarders, and golf carts travel quickly on the walkway. Again, these errors are reasonable, since there are very few examples of fast moving bicycles and no examples of golf carts in the training set. These cases could be handled by adding more mixture components to the segmentation algorithm, which would label fast moving objects as different classes. Another GP could then be trained to count the number of fast moving vehicles in the scene. Another possibility would be to simply identify these objects as outliers, based on the posterior assignment probabilities of the segmentation stage. Any of these possibilities would require larger training sets, with a richer representation of the outliers. We intend to analyze them in future work.

Acknowledgments

The authors thank Jeffrey Cuenco for providing some of the ground-truth data. This work was partially funded by NSF award IIS-0534985 and NSF IGERT award DGE-0333451.

References

[1] J. Vaidya, M. Zhu, and C. W. Clifton, *Privacy Preserving Data Mining*. Springer, 2006.

[2] V. Verykios, E. Bertino, I. Fovino, L. Provenza, Y. Saygin, and Y. Theodoridis, "State-of-the-art in privacy preserving data mining," *ACM SIGMOD Record*, vol. 33, no. 1, pp. 50–57, March 2004.

[3] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *IJCV*, vol. 63(2), pp. 153–61, 2005.

[4] T. Zhao and R. Nevatia, "Bayesian human segmentation in crowded situations," in *CVPR*, vol. 2, 2003, pp. 459–66.

[5] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *CVPR*, vol. 1, 2005, pp. 875–85.

[6] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors," in *ICCV*, vol. 1, 2005, pp. 90–7.

[7] S.-F. Lin, J.-Y. Chen, and H.-X. Chao, "Estimation of number of people in crowded scenes using perspective transformation," *IEEE Trans. System, Man, and Cybernetics*, vol. 31, no. 6, 2001.

[8] V. Rabaud and S. J. Belongie, "Counting crowded moving objects," in *CVPR*, 2006.

[9] G. J. Brostow and R. Cipolla, "Unsupervised bayesian detection of independent motion in crowds," in *CVPR*, vol. 1, 2006, pp. 594–601.

[10] B. Leibe, K. Schindler, and L. Van Gool, "Coupled detection and trajectory estimation for multi-object tracking," in *ICCV*, 2007.

[11] N. Paragios and V. Ramesh, "A mrf-based approach for real-time subway monitoring," in *CVPR*, vol. 1, 2001, pp. 1034–40.

[12] S.-Y. Cho, T. W. S. Chow, and C.-T. Leung, "A neural-based crowd estimation by hybrid global learning algorithm," *IEEE Trans. Syst, Man, Cybern.*, vol. 29, pp. 535–41, 1999.

[13] C. S. Regazzoni and A. Tesei, "Distributed data fusion for real-time crowding estimation," *Signal Process.*, vol. 53, pp. 47–63, 1996.

[14] A. C. Davies, J. H. Yin, and S. A. Velastin, "Crowd monitoring using image processing," *Electron. Comm. Eng. J.*, vol. 7, pp. 37–47, 1995.

[15] D. Kong, D. Gray, and H. Tao, "Counting pedestrians in crowds using viewpoint invariant training," in *British Machine Vision Conf.*, 2005.

[16] A. N. Marana, L. F. Costa, R. A. Lotufo, and S. A. Velastin, "On the efficacy of texture analysis for crowd monitoring," in *Proc. Computer Graphics, Image Processing, and Vision*, 1998, pp. 354–61.

[17] L. Dong, V. Parameswaran, V. Ramesh, and I. Zoghlami, "Fast crowd segmentation using shape indexing," in *ICCV*, 2007.

[18] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30(5), pp. 909–26, 2008.

[19] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *Intl. J. Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.

[20] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intellig.*, vol. 8, pp. 679–714, 1986.

[21] A. N. Marana, L. F. Costa, R. A. Lotufo, and S. A. Velastin, "Estimating crowd density with minkoski fractal dimension," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, vol. 6, 1999, pp. 3521–4.

[22] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[23] "Privacy preserving crowd monitoring: Counting people without people models or tracking." [Online]. Available: http://www.svcl.ucsd.edu/projects/peoplecnt

[24] J. Shi and J. Malik, "Motion segmentation and tracking using normalized cuts," in *ICCV*, 1999, pp. 1154–60.

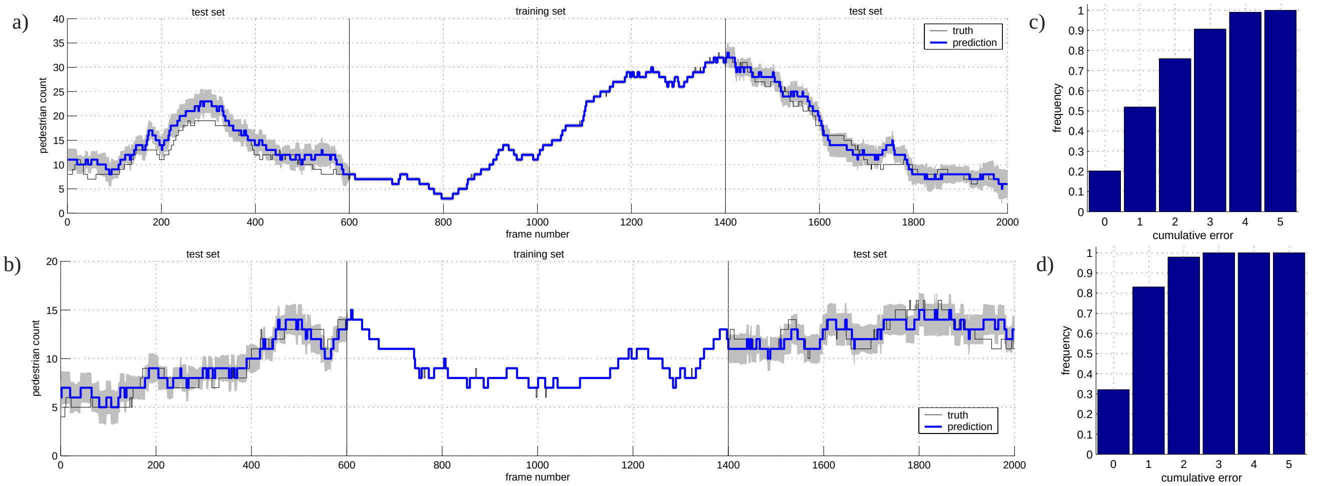

Figure 8. Crowd counting results over both the training and test sets for: (a) people moving away and (b) people moving towards the camera; and (c, d) the corresponding cumulative error on the test set. The gray bars show the two standard-deviations error bars of the GP.

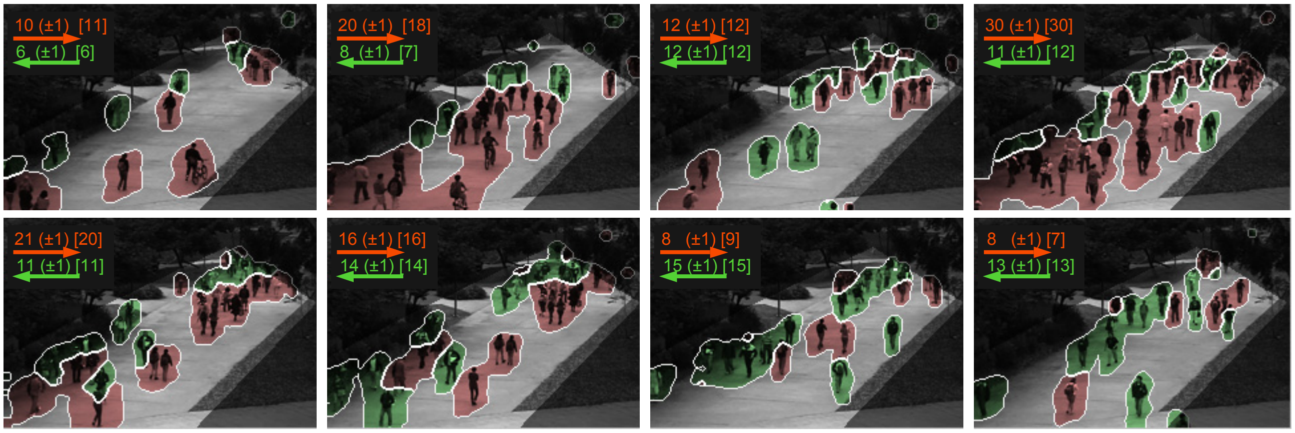

Figure 9. Crowd counting results: The red and green segments are the “away” and “towards” crowds. The estimated crowd count for each segment is in the top-left, with the (rounded standard-deviation of the GP) and the [ground-truth]. The ROI is also highlighted.

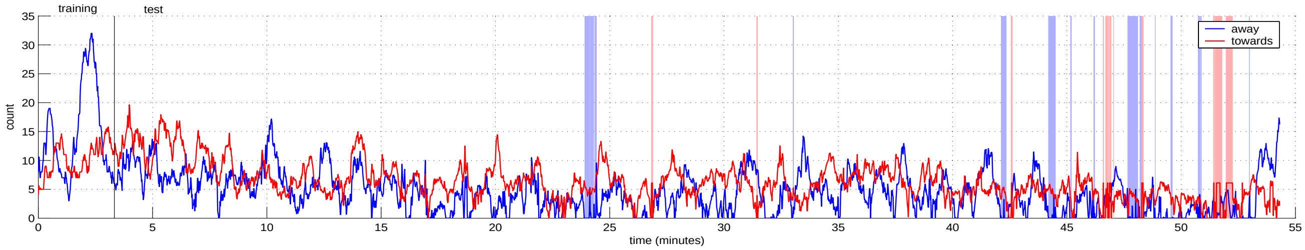

Figure 10. Count estimates on 55 minutes of video. The shaded bars indicate periods when the GP model had low confidence ($\sigma > 3$).

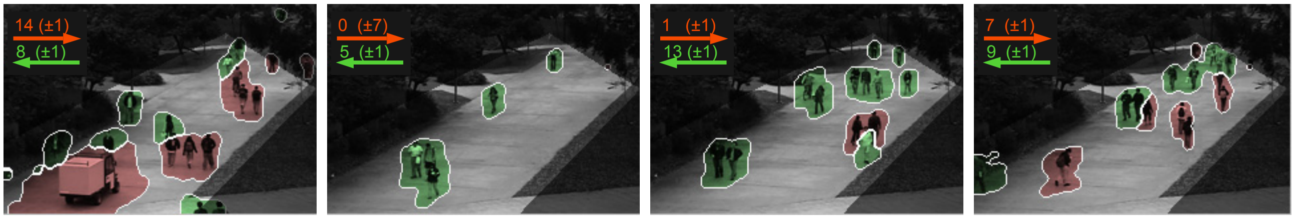

Figure 11. Example counting results on 55 minutes of video. The counts are in the top-left, with the (rounded standard-deviation).