Fast Crowd Segmentation Using Shape Indexing

Lan Dong*
Dept. of Electrical Engineering
Princeton University

Vasu Parameswaran, Visvanathan Ramesh, Imad Zoghlami
Real-Time Vision and Modeling Department
Siemens Corporate Research

Abstract

This paper presents a fast, accurate, and novel method for the problem of estimating the number of humans and their positions from background differenced images obtained from a single camera where inter-human occlusion is significant. The problem is challenging firstly because the state space formed by the number, positions, and articulations of people is large. Secondly, in spite of many advances in background maintenance and change detection, background differencing remains a noisy and imprecise process, and its output is far from ideal: holes, fill-ins, irregular boundaries etc. pose additional challenges for our "mid-level" problem of segmenting it to localize humans. We propose a novel example-based algorithm which maps the global shape feature by Fourier descriptors to various configurations of humans directly. We use locally weighted averaging to interpolate for the best possible candidate configuration. The inherent ambiguity resulting from the lack of depth and layer information in the background difference images is mitigated by the use of dynamic programming, which finds the trajectory in state space that best explains the evolution of the projected shapes. The key components of our solution are simple and fast. We demonstrate the accuracy and speed of our approach on real image sequences.

1. Introduction

Detecting, localizing, and tracking humans in a scene are important problems that occur in several current applications areas such as security, safety, smart interfaces, video-retrieval, etc. These application domains usually involve fixed cameras, allowing the use of background modelling and change detection algorithms which produce a change measure at each pixel in a video stream. The resulting distance images, along with the original images, can be used to detect humans and their movements in the scene. While isolated humans can of course be trivially detected, groups of people close together and partially occluding each other are more difficult to localize and track. This *crowd segmentation* problem, i.e. localizing individual humans in a crowd, has been the focus of several researchers in the past, and forms the focus of this paper.

1.1. Previous work

Previous approaches for the problem of detecting and localizing multiple occluding people in images and video can generally be grouped into three categories: *appearance based*, *grouping based*, and *generative models based parameter optimization* approaches. Appearance based approaches include the use of "head detection", where the "Ω" shape of head and shoulder contour is important cue to localizing people (e.g. [19, 20, 8]). The head cannot always be detected reliably across different viewing angles and far distances, and can only be used as a supporting cue. Another thread in this category is the use of learned local appearance descriptors and their grouping. For example, Leibe et al [11] describe an interest-point and local feature descriptor based detector, followed by global grouping constraints to detect humans. Wu and Nevatia [18] describe a parts based human detector and extend it to handle multiple humans. Such approaches are complex, computation intensive, and it is not clear how well they can generalize to arbitrary surveillance situations and people appearances. Grouping based approaches typically use motion features to isolate tracks of people, and infer their positions in frames: In [13, 3], trajectories over several frames are clustered in space and time for coherence. This method is best used to count moving objects in dense crowds but is not satisfactory for localization. Generative model based parameter optimization approaches model the image formation process as parameterized by the attributes of humans in the scene and search for the parameter set that best explains the observed image: Rittscher *et al*. [15] have proposed a method based on partitioning a given set of image features using a likelihood function that is parameterized on the shape and location of potential individuals in the scene. They use a variant of the Expectation Maximization algorithm to perform global annealing based optimization and find maximum likelihood estimates of the model parameters and the

*Work performed during author's internship at Siemens Corp. Research

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

grouping. In [7], humans are assumed to be isolated as they enter the scene so that a human specific color model can be initialized for segmentation when occlusion occurs later. This initialization assumption is not necessarily valid in crowded situations. In [10], a generalized cylinder based representation is used to model humans and their appearance, and their number and positions are tracked using a particle filter. Zhao and Nevatia [20] describe a generative process where the parameters include the number of people, their location and their shape. They use Markov Chain Monte Carlo (MCMC) to achieve global optimization by searching for maximum likelihood estimates for the model parameters. The main issue with these approaches is the complicated and high dimensional parameter space and the subsequent slow process of searching for the best parameters.

1.2. Our Contributions

Our key insight into this problem is that the overall shape of the foreground blob encodes a rough estimate of the number of people and their positions in the blob: indeed, when presented with a foreground blob, we humans, in a lot of cases, are very quick to infer the number and positions of people in it. This observation and the need for real-time operation suggests building an indexing function that maps a suitable representation of the blob shape into the number and position of humans (i.e. the *parameter set*) that generated the shape. Our contributions are the following: (1) Viewing the crowd segmentation problem as one of shape matching and its formulation in an example based framework. Specifically, mappings are learned between a shape descriptor space and the space of parameters generating the shapes, and stored in a look-up table. This avoids expensive searches and returns a rough estimate of candidate parameter sets. (2) Use of *locally weighted regression* (LWR) over the candidate parameter sets to quickly estimate the one that best explains the shape, and (3) When a stream of images is available, the use of dynamic programming to remove inherent ambiguities due to the lack of depth and layering information in the foreground blobs and to further refine the estimate. It is important to note that our indexed look-up of the parameter set cannot be expected to outperform previously explored searching strategies. However, our proposed indexing strategy will produce quick and good initial guess that can, if necessary, be augmented with previously proposed searching algorithms (e.g. [15], [20]), thus making real time operation possible. Further, as we report in section (4.3) there is no need for additional searching in most cases, and where search is necessary, the number of iterations required for convergence is reduced by many hundreds or thousands. This said, the primary focus in this paper is to describe our shape based formulation of the problem and its solution rather than a full-fledged integrated system.

The remainder of this paper is organized as follows. Our example-based estimation framework is explained in section 2. Section 3 explains how the impact of errors in the background estimation process and articulation are mitigated and how temporal information can be incorporated using dynamic programming to yield accurate estimates of the parameter set. We present experiments and validation in section 4 We finally conclude with observations and future work in section 5.

2. Example-based estimation framework

Let $\theta = \{n, \mathbf{b}_1, \ldots, \mathbf{b}_n\}$ be the set of humans and their positions in a given foreground blob: here n is number of people and $\{\mathbf{b}_i, i = 1, \ldots, n\}$ are their relative spatial positions. This set generates an image feature vector $\mathbf{x}$. The underlying generative process is formulated as $\mathbf{x} = f(\theta)$, where f is usually quite complicated. For the particular problem of segmentation of humans in crowds, we are interested in recovering θ from $\mathbf{x}$: $\theta = f^{-1}(\mathbf{x})$. In an example-based approach an explicit form of the inverse generative process f^{-1} is not required. Instead, parameter values for novel input are estimated from known values for similar examples. A training set of labelled examples $(\mathbf{x}_1, \theta_1), \ldots, (\mathbf{x}_N, \theta_N)$ is stored. The training examples should be fairly dense throughout the parameter space so that for novel input there will be at least one example close to it. The goal becomes one of minimizing the residual in terms of the distance, d_θ, in parameter space. Assuming sufficient sampling density, we can achieve this by minimizing the distance $d_{\mathbf{x}}$ in the feature space by similarity search. Example-based approaches have been explored in the areas of object recognition and pose estimation [12, 16, 1] and here we explore their use for fast crowd segmentation. The following questions arise naturally, and are answered in the following sections: (1) What image features, representations and filters provide robustness to inaccuracies in the foreground extraction process and to nuisance perturbations of the parameter set such as body articulations? (2) How should the mapping between the parameter set and the feature representations be learned and stored? (3) How should a feature be mapped to the parameter set? and (4) How can we resolve ambiguities inherent in the mapping process?

2.1. Image Features

Background estimation and change detection are fairly developed areas in visual surveillance and a number of sophisticated algorithms exist to compensate for various extraneous effects such as flickering, lighting changes, weather, motion etc., and the change detection process is typically insensitive to variations in surfaces due to color and texture. The downside is that effects such as strong reflections, shadows, and regions where foreground and back-

ground have similar colors, cause perturbations in the final output. A recent survey can be found in [14]. These techniques typically return distance images which can be thresholded for acceptable false alarm rates to yield binary images depicting changes as blobs. We take as input to our system, the silhouette of a connected blob and note that the silhouette encodes a lot of information about the number and positions of people in it (other work, e.g. [15] has also used the same kind of input). A number of choices exist to represent the silhouette shape: shape context [2], Hu moments [9], Fourier descriptors, etc. We choose Fourier descriptors for allowing the ability to represent shape to a desired level of detail by filtering, compactness in representation, and fast computation. Further, as reported in [12], Fourier descriptors (FD) and shape context outperformed Hu moments for pose recovery. We found that seven Fourier coefficients are all it takes to recover the parameter set with good accuracy.

We sample the silhouette using a fixed number of points M: $\{(x_1, y_1), \ldots, (x_M, y_M)\}$ on the external boundary. We use equidistant sampling to make it more uniform. The sample points are transformed into complex coordinates $\{z_1, \ldots, z_M\}$ with $z_j = x_j + iy_j$ where $i^2 = -1$ and are further transformed to the frequency domain using a Discrete Fourier Transform (DFT). The result is Fourier coefficients $\mathbf{F} = [F^{(1)}, \ldots, F^{(M)}]$, which are complex numbers. These coefficients can be used as descriptors of the silhouette. Figure 1 (a) to (d) illustrate the transformation steps from input image to Fourier descriptors. To achieve position invariance, the first coefficient is set to 0. In contrast to [12], we do not seek rotation invariance: Humans are assumed to be upright, and rotation invariance will in fact result in some information loss. The FD coefficients at low frequency contain information on the general characteristics of the shape and the ones at high frequency contain information on finer details. Also, the coefficients at high frequency are much smaller in magnitude. A natural choice is to use only the coefficients at low frequency as the shape signature. Two advantages of doing so are that we achieve robustness to fine local variations and, at the same time, speed up similarity searching. Figure 1(e) shows the reconstructed shape of the image by using the first and last seven coefficients only (the DFT here ranges from 0 to 2π, and so the first and last seven coefficients are correlated, both corresponding to low frequencies). Experimental results show that the number of coefficients used can be significantly less than the number of sample points on the boundary. Given two shapes with Fourier Descriptors $\mathbf{F}_1$ and $\mathbf{F}_2$, we use the Euclidean distance as a similarity measure between the two shapes:

$$d_{\mathbf{x}} = \sqrt{\sum_j \left\| F_1^{(j)} - F_2^{(j)} \right\|^2} \qquad (1)$$

$d_{\mathbf{x}}$ is the distance of the two shapes in feature space $\mathbf{x}$ and is used as similarity measure. Its relation to their distance in parameter space is described below.

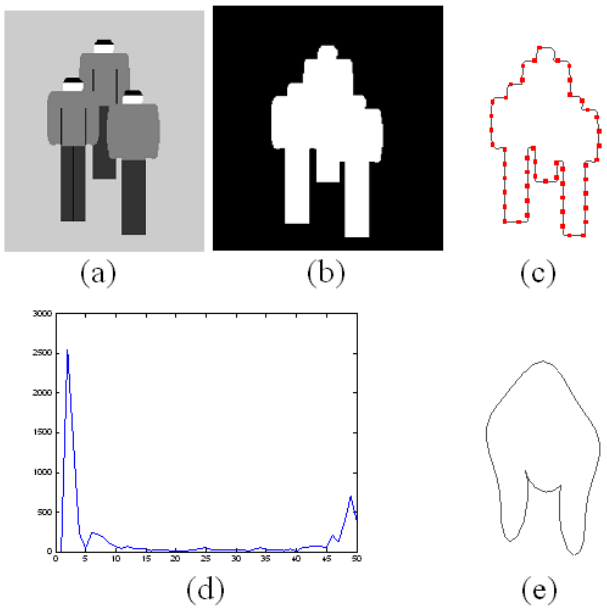

Figure 1. From input image to Fourier Descriptor (a) input image (b) foreground blob (c) sampling points on the boundary ($M = 50$) (d) magnitudes of Fourier Descriptor (e) reconstructed shape from 14 Fourier coefficients

2.2. Local Regression

Given a parameter set θ, keeping the number of people n fixed, if one were to vary the locations of the people or their articulations infinitesimally, one would expect the image feature $\mathbf{F}$ to also change infinitesimally. The training process samples points in the two spaces while the testing process results in a "query point" in the image feature space. The query point will typically not coincide with a sample point but be close to several sample points (assuming sampling is sufficiently dense). The problem now becomes that of "interpolation" among the neighbors of the query points. One possible solution is to use the closest match returned as the initialization point of a search based strategy. Doing this alone speeds up the crowd segmentation process proposed in [20] as we show in section 4.3. Nevertheless, motivated by the need for real time operation, we seek to eliminate or postpone the need for searching using local regression. The classic K nearest neighbors technique (K-NN) takes the sample mean among the closest K sample points and is known to be consistent and to achieve Bayes-optimal risk for many loss functions [6]:

$$\hat{\theta}_{\text{NN}} = \frac{1}{K} \sum_{\mathbf{x}_i \in \text{K-NN}} \theta_i \qquad (2)$$

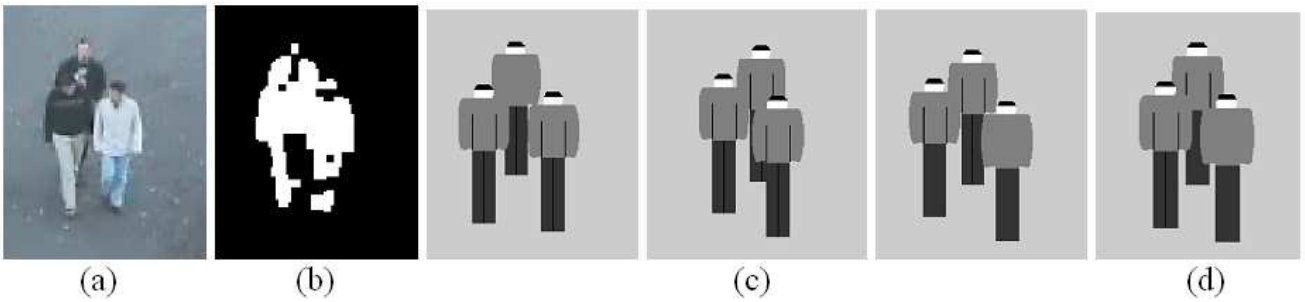

Figure 2. A sample of example-based segmentation algorithm (a) input image (b) foreground blob (c) top 3 matched examples (d) Interpolated result.

$K = 1$ takes the closest sample point. A natural extension to *K-NN* is locally-weighted regression (LWR) [5], where each sample point is given a weight. The procedure fits a *local* function to the neighborhood around the query point and evaluates the function at the query point. The function can be fit using weighted least squares, giving more weight to sample points near the query point and less weight to those farther away. The local approximating function can be chosen from a particular class $g(\mathbf{x};\beta)$ which is typically a low-order polynomial, often of degree one or even zero (higher degrees prone to over-fitting) The parameters β are chosen to minimize the weighted square errors in the test input $\mathbf{x}_0$:

$$\beta^* = \arg\min_{\beta} \sum_{\mathbf{x}_i \in \text{K-NN}} (g(\mathbf{x}_i;\beta) - \theta_i)^2 \mathcal{K}(d_{\mathbf{x}}(\mathbf{x}_i, \mathbf{x}_0)) \quad (3)$$

where $\mathcal{K}$ is a kernel function that assigns lower weight to larger distances from the query point. For low-order polynomial functions $g(\mathbf{x};\beta)$, a closed form solution for β^* can be obtained by taking derivatives. The parameter set at the query point is then given by:

$$\theta_0 = g(\mathbf{x}_0;\beta^*) \quad (4)$$

For an example of this, see figure 2.

We experimented with a robust version of LWR (see [4]) which reduces the influence of outliers via an iterative process where each iteration re-calculates the weights. However, we did not find it much better than classic LWR and in our experiments, we used classic LWR to interpolate for the best location of the humans in the blob *assuming* a correct n. How n itself is estimated is described in section 3.

3. Image Space and Time Filtering

Current background differencing algorithms are not always robust to camouflage (similar colors between foreground and backgrond) resulting in gaps, holes, irregular boundaries and shape distortions. Figure 2(b) shows a real-world example of background differencing where some parts of the bodies are missing and parts of the bodies are not all connected. Problem also arise due to viewpoint and body articulations. The two images in figure 4(a) show one person with different views and articulations where the boundaries are quite significantly different from one another. While low pass filtering via Fourier descriptors eliminates finer distortions we used morphological filtering using a rectangular structuring element over the foreground image to reduce the impact of blob fragmentation and articulation. Figure 3 shows an example of two input images having the same parameter set (one real image and another simulated), and their filtered images. The filtering process helps connect fragmented blobs and forms a better external boundary, making the filtered shapes very similar. The beneficial effect of the filtering process on articulation is shown in figure 4(b).

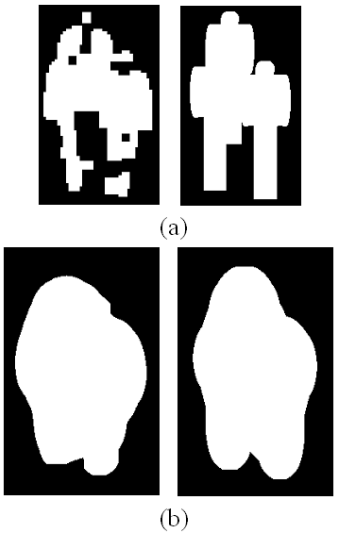

Figure 3. Comparison of two input images of same parameter. (a) input image: left is from real video as in figure 2 (b) right is from simulation. (b) filtered output.

The problem of determining the number of people n

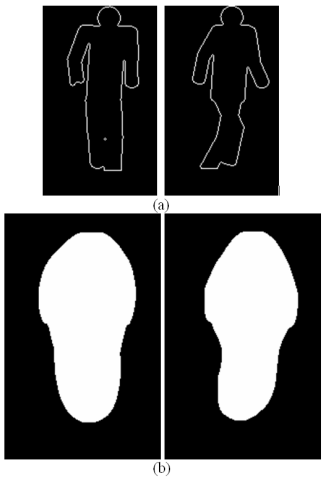

Figure 4. Comparison of two input images of same parameter. (a) input image (b) filtered output.

from a foreground blob is inherently ill-posed: For certain configurations, consisting of a given number of people (e.g. three or more), it is possible to insert another person into the mix and end up with the exact same foreground blob. Hence, while the mapping process can disambiguate between some configurations there will be a significant number of configurations where the top match will be incorrect. Nevertheless, if we consider all of the examples which are within a small distance to the query point, they almost surely contain the correct configuration. This is demonstrated in table 1, section 4 that shows the confusion matrix for random configurations of 1 to 6 overlapping people with random articulations where the configuration with the smallest Euclidean distance to the examples is considered to be the final answer. While there is confusion between some "neighboring" number of people such as between 3 and 4 people and between 5 and 6 people, the correct answer is almost always found as one within a given distance to the query point, the distance calculable by simulation (see table 2, section 4). Our solution for disambiguation is to use temporal coherence, by observing that as people move about in the blob, their number will not change and there is a possibility to distill the correct solution from the candidate configurations. The strategy we employ is to use dynamic programming. The matched neighboring examples are first clustered by different n. For each n, we use LWR as described in section 2.2 to interpolate the locations of the people from the first k best matches (or less if there are fewer than k matches) corresponding to the same n. The n value, together with the locations is represented as a node (state) θ in a dynamic programming formulation. Now consider the task of finding the most likely path through a graph of these nodes over time, where the likelihood of any path is the product of the transition probabilities along the path and the probabilities of the given observations at each state. Essentially, our goal is to maximize, for each time t, the probability $p(\theta_1, \ldots, \theta_t|\mathbf{x}_{1:t})$ where $\mathbf{x}_t$ is the observation at time t, namely, the observed foreground blob. Assuming the process is Markovian, we can write down the following dynamic programming recurrence:

$$\max_{\theta_1,\ldots,\theta_t} p(\theta_1, \ldots, \theta_t, \theta_{t+1}|\mathbf{x}_{1:t+1}) \propto p(\mathbf{x}_{t+1}|\theta_{t+1}) \quad (5)$$
$$\times \max_{\theta_t} \left[p(\theta_{t+1}|\theta_t) \max_{\theta_1,\ldots,\theta_{t-1}} p(\theta_1, \ldots, \theta_t|\mathbf{x}_{1:t}) \right]$$

We define expressions for the observation and transition probabilities and recurse for the best parameter set. The observation model is stationary and does not depend on time. Therefore, we omit the subscript t for the observation probability $p(\mathbf{x}|\theta)$. For the observation model it is reasonable to use the multi-human joint likelihood defined in [20].

$$p(\mathbf{x}|\theta) \propto \Big(\prod_{i=1}^{n} e^{-\lambda_1 A_i}(1 - e^{-\lambda_2 A_i})\Big) e^{-(\lambda_{10} N_{10} + \lambda_{01} N_{01})} \quad (6)$$

where A_i is image size. N_{01} is the number of pixels in the blob absent in the hypothesis (vice versa for N_{10}. See [20] for detail). For the transition probability, we define:

$$p(\theta_t = n_1|\theta_{t-1} = n_2) = \begin{cases} 1 - \epsilon & \text{if } n_1 = n_2 \\ \epsilon & \text{if } n_1 \neq n_2 \end{cases}$$

The idea is that we want to penalize a change in the number of people, given that the blob remains spatially connected. The total number of states at each time is small (6 at most, usually 2 to 3) so that the computation load is small.

4. Experiments

4.1. Indexing Using Look-up Tables

We use motion capture from walking humans and render them using a simple body model. We use an "average" shape of the different poses (shown in figure 5) for training as we are not interested in determining the pose of bodies in a foreground blob and given that articulation effects are mitigated via spatial filtering. This reduces the size and time required to build the lookup table. The training shape does not have a significant impact on the results: We found that an ellipse did about as well as the "average" shape. The "average" image is scaled to a canonical 120×200 pixels and adjusted for perspective w.r.t. its relative position in the blob. We simulated different random positions and numbers of people such that a connected blob is formed, calculated Fourier descriptors for the parameter set and stored the mapping. We chose $80, 400, 2000, 10000$ and 50000 samples for $n = 2, 3, 4, 5$ and 6 respectively. We chose 64 sample

$n\backslash\hat{n}$	1	2	3	4	5	6	Detected
$n=1$	200	0	0	0	0	0	100%
$n=2$	0	148	50	2	0	0	74%
$n=3$	0	11	85	94	9	1	42.5%
$n=4$	0	0	8	82	64	46	41.0%
$n=5$	0	0	0	14	72	114	36.0%
$n=6$	0	0	0	1	27	172	86%

Table 1. Confusion matrix for top match, 200 tests for each n.

$n\backslash\hat{n}'$	1	2	3	4	5	6	Detected
$n=1$	200	0	0	0	0	0	100%
$n=2$	0	200	0	0	0	0	100%
$n=3$	0	1	198	1	0	0	99%
$n=4$	0	0	0	197	1	2	98.5%
$n=5$	0	0	0	0	197	3	98.5%
$n=6$	0	0	0	0	1	199	99.5%

Table 2. Confusion matrix for neighbors within a distance 800, 200 tests for each n.

Figure 5. "Average" Human Shape

points on the boundary and chose 7 Fourier coefficients for the representation. The structuring element for morphological filtering was a rectangle of size 45×60, found empirically to produce the best results. Our limit of 6 on n was motivated by the observation that for moderate crowd densities there are usually no more than 6 people in a single connected blob (there being no limit on the total number of people in the scene).

Table 1 shows the confusion matrix using a naive lookup where the parameter set with smallest $d_{\mathbf{x}}$ is considered to represent a query blob (from 1200 random examples). The row corresponds to the true number of people in the input and the column corresponds to the estimate. The poor performance is indicative of the inherent ambiguity in the mapping process and it should be noted that as long as the correct answer is among the top k returns, it can be filtered out through dynamic programming and LWR (section 3). Rather than a fixed k, we found that for a Fourier threshold of 800, the correct answer is in the set over 98% of the time (see table 2).

4.2. Out-of-Range Detection

There is a fundamental limit to the number of people that can be represented in our indexing. For n within the training set, the indexing can be applied for fast crowd segmentation but for larger n, a more complex searching algorithm may need to take over. It becomes important that the algorithm self-diagnose that it is out of its operating range. One way to do this is to use the probability density of the normalized blob area (accounting for perspectivity) to classify the blob as out-of-range. We estimate this probability distribution via simulation. We calculate a threshold T as follows:

$$\int_{-\infty}^{T} p(f|n=m)\, df = 1 - r\% \tag{7}$$

where $p(f|n)$ is the probability density function of the normalized blob area given n people in it, m is the limit and $r\%$ is the detection rate required. Blobs that have a normalized area more than T will be diagnosed as being out of range. Note that our goal here is to find situations where our method cannot be applied (rather than where it can be applied) for hand-over to a search based algorithm.

4.3. Real Image Sequences

We ran experiments on frames from two different video sequences. One of the sequences was used in previous work ([20]) and the other was shot from a parking lot. We present results on two aspects: (1) Accuracy of n and of the estimated locations of people via manually labeled ground truth and (2) Convergence speed-up of the MCMC based search algorithm described in [20].

4.3.1 Performance

We first applied a Stauffer-Grimson based [17] algorithm for change detection on the video sequence, yielding connected foreground blobs in each frame. As our algorithm is constrained to work on complete blobs for now, we discarded incomplete blobs touching the image borders for our experiments. Frame-to-frame blob correspondence was established via matching blobs based on their proximity and similarity in area. The blobs were scaled for perspective and filtered using the morphological operation described in section 3. Then they are indexed based on their Fourier descriptors followed by LWR and dynamic programming to output the parameter set that best explained the shape and evolution of the blobs. Table 3 shows the confusion matrix on the number of people (row corresponding to the ground truth and column to the estimate). Another pertinent metric is the accuracy of the location estimates of the detected humans. For this we use the "distance error" ϵ_d namely the average normalized Euclidean distance between the relative

$n\backslash\hat{n}$	0	1	2	3	4	5	total
$n=1$	50	1275	10	0	0	0	1875
$n=2$	0	65	299	2	0	0	366
$n=3$	0	0	49	67	0	0	116
$n=4$	0	0	0	4	57	0	61
$n=5$	0	0	0	0	11	7	18
$n=6$	0	0	0	0	0	0	0

Table 3. Confusion matrix for real image sequence

	$n=2$	$n=3$	$n=4$	$n=5$
LWR	0.1351	0.1873	0.1900	0.2088

Table 4. Distance error for $n = 2, 3, 4, 5$.

position parameters of ground truth and the estimate:

$$\epsilon_d = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\left(\frac{x_{i,est}-x_{i,act}}{W_i}\right)^2+\left(\frac{y_{i,est}-y_{i,act}}{H_i}\right)^2} \tag{8}$$

N is the total number of people. The subscript est represents the estimate and act represents actual. W denotes the width of the human in the scene and H denotes the height. Table 4 shows the average distance error for a locally constant fitting function. Figure 6 shows several examples from the sequences where the people are clustered together in various configurations with significant occlusions and articulations. The overall detection rate was found to be 94.25%. Please note that as a post processing step, one could temporally smooth the results using the Kalman filter (not shown here). The run time speed of our algorithm depends on the number of blobs in each frame. For one blob, it takes less than 0.04 second of CPU time with unoptimized C++ code on a Pentium IV 2.8G Hz PC.

4.3.2 Speed-up Performance

We considered ways in which the MCMC based search algorithm of [20] can be aided by a fast indexing method as ours. Roughly speaking the main components of their search are: human hypothesis addition/removal, model and hypothesis switching, and diffusion. One approach to speed up the search is to index from shape to a parameter-set based on the best match (as described in section 2) followed by the normal MCMC search. Doing this alone reduces the number of iterations from 2000 to 1473 on average on the parking lot image sequence. Another approach is to use LWR and dynamic programming to determine n, followed by only diffusion (eliminating the need for hypothesis addition/removal or switching). This reduces to number of iterations required for convergence to 347. Figure 7 shows these two strategies in addition to the original MCMC search algorithm. As can be seen, both the strategies result in significant speed-up of the searching process.

Figure 6. Sample images from videos used. Left column: input; right column: output. Top 4 samples from [20]

5. Conclusions

We introduced the idea of viewing the crowd-segmentation problem as that of shape matching, and proposed a novel example-based formulation of the problem. Our approach was motivated by need for real-time operation, as well as the insight that the shape of a foreground blob encodes a lot of information about the number and positions of humans that generated the blob. We chose Fourier descriptors for representing a shape and built an indexing

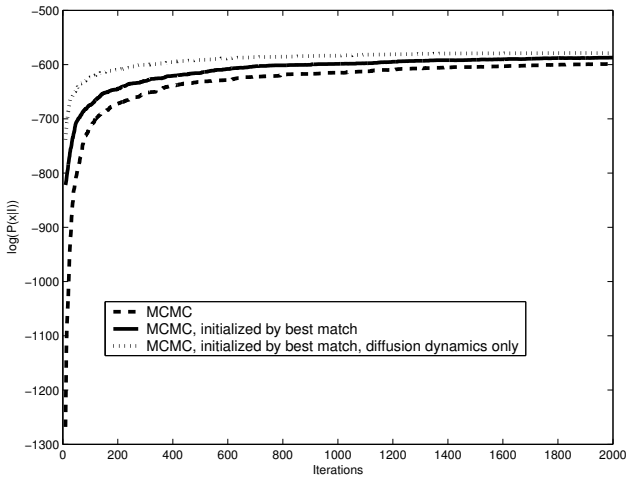

Figure 7. Three convergence curves of log posterior probability of a parking lot image. Each curve is averaged over 50 randoms runs on the image.

function that mapped observed descriptors to candidate parameter sets explaining the shape. We used locally weighted regression to calculate the best parameter set. Articulations and errors introduced in the foreground generation process were dealt with using morphological filtering operations while the inherent ambiguity involved in the mapping process was dealt with using dynamic programming. Our method provides very fast estimates of the parameter set for low to moderate number of people in a single blob and can provide initial guesses for a search based algorithm should the returned error be large, improving upon the speed of convergence. We showed how the algorithm can self-diagnose and report data to be out of its range of operation, so that a more complex search based algorithm can then be invoked. For future work, we want to characterize the input and output spaces better in terms of probability distributions of the Fourier descriptors over marginals of the parameter set. We also want to find out what is the minimum number of samples required for a given n. Finally, we want to systematically bring in additional cues to improve the robustness of the system. Cues such as moments, motion, edges, etc. can be brought in as additional features while retaining the same general framework.

References

[1] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2:432–439, 2003.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[3] G. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion in crowds. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1:594–601, 2006.

[4] W. Cleveland. Robust locally weighted regression and smoothing scatter plots. *Journal of the American Statistical Association*, 74:829–836, 1979.

[5] W. Cleveland and S. Delvin. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of American Statistical Association*, 83(403):596–610, 1988.

[6] T. Cover. Estimation by the nearest neighbor rule. *IEEE Trans. on Information Theory*, 14:21–27, 1968.

[7] A. E. Elgammal and L. S. Davis. Probabilistic framework for segmenting people under occlusion. *Proc. IEEE Intl. Conf. on Computer Vision*, 2:145–152, 2001.

[8] I. Haritaoglu, D. Harwood, and L. Davis. Hydra: multiple people detection and tracking using silhouettes. *Proc. IEEE workshop on Visual Surveillance*, pages 6–13, 1999.

[9] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions Information Theory*, 8(2):179–187, 1962.

[10] M. Isard and J. MacCormick. Bramble: a bayesian multiple-blob tracker. *Proc. IEEE Intl. Conf. on Computer Vision*, 2:34–41, 2001.

[11] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 878–885, 2005.

[12] R. Poppe and M. Poel. Comparison of silhouette shape descriptors for example-based human pose recovery. *Proc. Intl. Conf. on Automatic Face and Gesture Recognition*, pages 541–546, 2006.

[13] V. Rabaud and S. Belongie. Counting crowded moving objects. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1:705–711, 2006.

[14] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Trans. on Image Processing*, 14(3):294–307, 2005.

[15] J. Rittscher, P. Tu, and N. Krahnstoever. Simultaneous estimation of segmentation and shape. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2:486–493, 2005.

[16] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. *Proc. IEEE Intl. Conf. on Computer Vision*, 2:750–757, 2003.

[17] C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[18] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. *Proc. Intl. Conference on Computer Vision*, 1:90–97, 2005.

[19] T. Zhao and R. Nevatia. Stochastic human segmentation from a static camera. *Proc. IEEE workshop on Motion and Video Computing*, pages 9–14, 2002.

[20] T. Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2:18–20, 2003.