Marked Point Processes for Crowd Counting

Weina Ge and Robert T. Collins
The Pennsylvania State University
University Park, PA 16802, USA
{ge, rcollins}@cse.psu.edu

Abstract

A Bayesian marked point process (MPP) model is developed to detect and count people in crowded scenes. The model couples a spatial stochastic process governing number and placement of individuals with a conditional mark process for selecting body shape. We automatically learn the mark (shape) process from training video by estimating a mixture of Bernoulli shape prototypes along with an extrinsic shape distribution describing the orientation and scaling of these shapes for any given image location. The reversible jump Markov Chain Monte Carlo framework is used to efficiently search for the maximum a posteriori configuration of shapes, leading to an estimate of the count, location and pose of each person in the scene. Quantitative results of crowd counting are presented for two publicly available datasets with known ground truth.

1. Introduction

Detecting and counting people in video of a crowded scene is a challenging problem, since the spatial overlap between people makes it difficult to delineate individuals as connected component blobs within a background subtraction image. In this work we define a Marked Point Process (MPP) that couples a spatial stochastic process governing number and placement of individuals with a conditional mark process for selecting body size, shape and orientation. We use Reversible Jump Markov Chain Monte Carlo (RJMCMC) to compare hypothesized configurations of varying numbers of people to find a maximum a posteriori (MAP) estimate of the count and location of overlapping individuals in the scene (Figure 1). Such information has the potential to increase situational awareness for crowd control and public safely by providing real-time estimates of the number of people entering or exiting a venue.

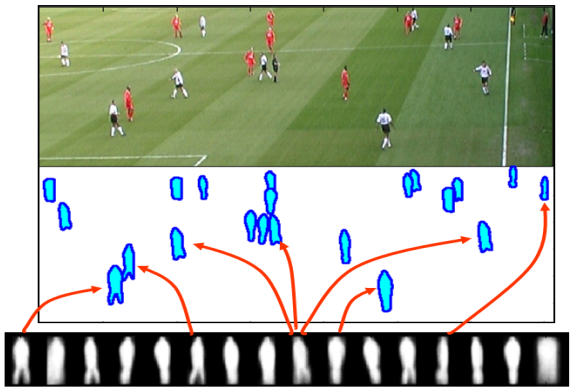


Figure 1. We use a marked point process to determine the number and configuration of multiple people in a scene. In addition to determining the location, scale and orientation of each individual, the MPP also selects an appropriate body shape from a set of learned Bernoulli shape prototypes, as displayed at the bottom.

1.1. Related Work

There have been several papers concerned with people counting in a crowd. In some, the crowd is treated as a static texture, and extracted features are used to classify how many people are present [15]. Other approaches derive area-based estimates by using prior calibration to relate the location and size of an image region to the number of people the region could contain given the specific perspective camera viewpoint [10, 12, 16]. In Rabaud and Belongie [18], motion vectors are clustered to estimate the number of moving objects. Many other works exist for people detection and tracking in less crowded situations [4, 8].

Several authors in the statistics literature have considered the problem of using MPPs to detect and count objects in images [1, 5, 7, 9, 14, 17, 20, 21]. These often include pedagogical, toy "object recognition" examples where simple shapes such as disks or polygons are recognized from noisy synthetic images. However, real-world examples have also been reported, include segmenting cells in confocal microscopy images [20, 21], and various appli-

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

cations in aerial image analysis, such as detecting elliptical shapes representing trees in a plantation [17], tracing out road networks [5] and detecting buildings from Digital Elevation Models [14]. These preceding examples all have a two-dimensional aspect to them, such that object overlap is either uncommon or disallowed completely. A recent exception by van Lieshout develops a sequential MPP for use in tracking depth-ordered overlapping shapes [23].

In the vision literature, the closest work to our own is by Zhao and Nevatia [24], who define a stochastic process to model overlapping spatial configurations of pedestrians while using RJMCMC to estimate a configuration that best explains a given foreground mask. However, there are major differences between the two approaches. Their pedestrian shape model is a hand-made set of parameterized body shapes composed of ellipses, whereas we estimate a mixture model of Bernoulli shape images from observed foreground data in a training sequence, thus providing a flexible mechanism for learning arbitrary object shapes. Their likelihood function is based on a binary "label image" that is compared to the observed foreground data, whereas our label image is composed of spatially varying Bernoulli mean parameters, providing a soft segmentation that captures the uncertainty and variability in observed object boundaries. In addition, we automatically estimate the extrinsic parameters that relate object size and orientation to location in the image.

1.2. Contributions

We introduce a conditional mark process to model known correlations between bounding box size/orientation and image location. The mark process parameters are decomposed into extrinsic appearance (geometry) and intrinsic appearance (shape and posture), learned separately. We adopt a Bayesian formulation for learning weighted Bernoulli shape masks using the EM algorithm, which gives us the flexibility to model a variety of shapes within the rectangular MPP framework. In addition to presenting illustrative examples on different scenes, we quantitatively evaluate pedestrian counting performance on two publicly available datasets where ground truth is known.

2. Marked Point Processes for Object Counting

For ease of explanation we describe a particular problem, but the approach is general. Consider counting pedestrians in a frame of video given a foreground mask image produced by background subtraction. The goal is to determine the number and configuration of binary pedestrian shapes that best explains the foreground mask data. One can think of this process as trying to place a set of cutout shapes over the foreground mask to "cover" the foreground pixels while trying to avoid covering the background pixels. For this section we assume rectangular shapes, leading to the problem of finding a rectangular cover. In Section 3.2 we show how to generalize this approach so that the rectangle becomes the bounding box for an arbitrary shape selected from a pool of learned candidate shapes.

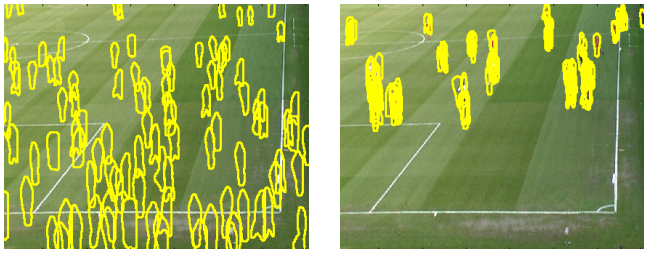
A spatial point process is a stochastic process that is suitable for modeling prior knowledge on the spatial distribution of an *unknown* number of objects. A realization of the process consists of a random countable set of points $\{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ in a bounded region $S \in R^d$. Like previous authors [3, 9, 20], we approach the covering problem within the framework of Marked Point Processes (MPP). An MPP couples a spatial point process Z with a second process defined over a "mark" space M of shapes such that a random mark $m_p \in M$ is associated with each point $p \in Z$. For example, a 2D point process of rectangular marks has elements of the form $\mathbf{s}_i = (p_i, (w_i, h_i, \theta_i))$ specifying the location, width, height and orientation of a specific rectangle in the image. In this paper, we propose a novel marked point process that weaves the prior knowledge of the spatial pattern, extrinsic size and transformation of objects, with intrinsic geometric shape information modeled by a mixture of Bernoulli distributions. Thus, the realization of the MPP in this paper consists of an image location p defined on a bounded subset of R^2, together with a mark m defining a geometric shape to place at point p. Section 2.1 describes the marked point process model and Section 3 explains the learning procedure for estimating extrinsic bounding box geometry and intrinsic shape models.

2.1. The Model

We take a Bayesian approach to model the objects in the scene as a set of configurations from an MPP that incorporates prior knowledge such as expected sizes of people in the image or knowledge about image regions where people will not appear. We denote the prior term for an object as $\pi(\mathbf{s}_i)$, and assume independence among the objects. Priors in MPPs are typically factored so that the mark process is independent from the spatial point process, that is $\pi(\mathbf{s}_i) = \pi(p_i)\pi(m_i)$. However, this common approach ignores obvious and strong correlations between the size and orientation of projected objects and their 2D image locations in views taken by a static camera. Hence, we introduce a *conditional mark process* for rectangles representing the shape and orientation of a 2D bounding box, conditioned on spatial location, leading to a factored prior of the form:

$$\pi(\mathbf{s}_i) = \pi(p_i)\pi(w_i, h_i, \theta_i|p_i) \quad (1)$$

The prior for the point process $\pi(p_i)$ is chosen as a homogeneous Poisson point process. This means that the total number of objects follows a Poisson distribution, and given the number of objects, the locations are i.i.d. and each uniformly distributed in the bounded region. Figure 2 shows a simulation from the prior model. One can also learn through



Prior point process Proposal point process

Figure 2. One hundred random samples drawn from the marked point process associated with the prior (left), and with the MCMC birth proposal (right). Samples from the proposal distribution on the right cluster around objects because we use a data-driven, inhomogeneous Poisson intensity function computed from the observed foreground mask.

passive observation of training video a density estimate of where people are likely to be seen [2, 22], and more importantly, where not to look for them (*e.g.* walls; sky). When available, we represent this information as an inhomogeneous Poisson point process that is defined by its density with respect to the reference Poisson point process.

The conditional mark process. We represent the prior for $\pi(w_i, h_i, \theta_i|p_i)$ as independent Gaussian distributions on the width, height and orientation of a pedestrian bounding box centered at a given image location p_i. The spatially-varying mean and variance parameters for each random mark are stored in lookup tables indexed by the image location. Section 3.1 describes how these distributions are automatically estimated offline.

Likelihood. Recall the data we are dealing with in the object counting problem is a foreground mask, assuming the foreground is formed by pedestrians in the scene. Formally, let y_i be the binary value of pixel i in the observed foreground mask data, with 1 = foreground, 0 = background. To compute the goodness of fit of a proposed configuration of shapes to the data, a common way is to first map the configuration into a *label* image [1] where pixels are labeled foreground if any of the shapes cover it, and background otherwise, so that each pixel in the label image has a one-to-one counterpart in the observed foreground mask. Let x_i be the values in the label image. Since both x_i and y_i are binary variables, Bernoulli distributions are used to characterize $p(y_i|x_i)$. In previous work [24], all the foreground pixels share the same Bernoulli distribution $p(y_i|x_i = 1) \sim \text{Bern}(y_i|\mu_f)$, while the background pixels share another $p(y_i|x_i = 0) \sim \text{Bern}(y_i|\mu_b)$.

Different from the previous work, instead of two Bernoulli distributions depending on the binary labeling of the pixel, we propose a novel way to create a *soft* label image by generating shape configurations represented by a mixture of Bernoulli distributions (3.2). The resulting x_i in the label image is therefore no longer a binary variable, but a continuous variable ranging from $[0, 1]$, the mean parameter of the Bernoulli distribution $p(y_i|x_i)$.

Assuming conditional independence among the pixels, the joint likelihood function can be written as

$$\log \mathcal{L}(Y|X) = \log \prod_{i=1}^{N} p(y_i|x_i)$$
$$= \sum (x_i \log y_i + (1 - x_i) \log(1 - y_i)) \quad (2)$$

This likelihood function is biased towards MAP solutions with multiple overlapping rectangles that claim almost the same set of foreground pixels. Although such over-counting does not increase the likelihood score; neither does it decrease it, since there is no penalty for overlap. Many authors address this problem by including a "hard-core" penalty that disallows any overlap by adding an infinite penalty when overlap occurs [9, 20, 21]. This is too strict for the present application of people counting, who may overlap in the view. Another principled approach is to add pairwise interaction terms into the likelihood function to penalize the area of overlap between each pair of shapes [1]. We take this latter approach, and implement a simple scheme where the number of overlapping pixels is multiplied by a non-negative factor ρ to form a penalty term subtracted from the log likelihood function. We set $\rho = 0.1$. That is sufficient to discourage completely overlapping shapes while still allowing small to moderate overlap to occur when it increases the log likelihood value.

The likelihood function and the prior combine to form a posterior that measures how well the observed foreground mask can be described as a noisy instantiation of an MPP consisting of zero or more overlapping bounding boxes at varying orientations and scales, along with a soft Bernoulli weight assigning pixels within the bounding box a probability of being foreground or background. Pedestrian detection and counting then becomes the problem of finding the MAP estimation over a configuration space. In section 4 we address the search for the number of shapes and their parameters using RJMCMC.

3. Learning the Conditional Mark Process

Simple geometric shapes such as rectangles or ellipses are only a coarse approximation to the shape of objects we want to count. In this section, we learn the parameters of a mark process that well-approximates the appearance of foreground shapes. Our notion of shape appearance is decomposed into two parameter sets: an extrinsic shape mapping and a set of intrinsic shape classes. The extrinsic shape mapping determines the translation, rotation and scaling of a centered shape model into image pixel coordinates. The intrinsic shape classes specify a library of different reference shape prototypes that can be selected for matching.

Complete characterization of a foreground mask thus involves selecting the appropriate intrinsic shape prototypes and then translating, rotating and scaling them into the image to cover the foreground pixels as well as possible.

3.1. Estimating Extrinsic Shape Mappings

The image size and orientation of a standing person will vary as a function of image location. We use lookup tables of Gaussian distributions at each pixel to represent the distribution of width, height, and orientation of pedestrian bounding boxes at different locations in the image. We automatically estimate the means and variances of the Gaussian distributions from a small sample of the sequence where the crowd density is low. Inferring camera calibration parameters from watching people in the scene has also been considered in previous work [11, 13, 19].

Orientation Estimation: Since we know pedestrians will be oriented vertically, it suffices to determine the vertical vanishing point of the scene, which completely determines the 2D image orientation of a vertical object at any pixel. Conversely, we can estimate the vertical vanishing point from the measured major axis of elliptical blobs extracted from foreground masks of walking people. Often automatically generated foreground masks are noisy, requiring robust estimation techniques.

Figure 3 illustrates computation of the vertical vanishing point for a sample sequence. Foreground masks are computed for each frame via background subtraction. Blobs are found by connected components, followed by ellipse fitting to compute their center of mass and second moments. We repeat the process of extracting major axis orientation of blobs for all frames in a short sequence of video. In this example, over 7000 axes are observed. Some of these represent vertical orientation of individuals who are found as a single blob; however, many others are outliers representing the orientation of multi-person blobs, fragmented blobs, or blobs whose second moments are corrupted away from vertical by arms and legs extending out from the person.

To find the vertical vanishing point, we assume that the inlier axes will converge to a vanishing point. We further assume that the outlier axes have no consistent bias: they will not intersect in significant numbers at any other point in the image. We use RANSAC to find the intersection point voted for by the most axes.

In the example shown, out of 7013 axes, the largest inlier set contains 232 axes. Computing the best intersection point from this inlier set involves finding the eigenvector associated with the smallest eigenvalue of a 3x3 scatter matrix formed from the observed inlier axes. We see that the computed vanishing point correctly captures the change in image orientation of people at different parts of this scene. The orientation of a blob centered at any pixel in the image can now be computed, and stored in a lookup table representing the mean of a Gaussian distribution on orientation.

Height and Width Estimation: A reasonable first-order model of many scenes can assume that people are walking or standing on a planar ground surface. The planarity assumption regularizes the computation of size, by constraining the relative depth of people in the scene as a function of image location.

Explanation of fitting height and width distributions based on the size of observed foreground blobs is simplified by considering views where the vertical vanishing point is along the y axis of the image coordinate system, which can be achieved with an in-plane image rotation. If the vanishing point is far from the image, *i.e.* for small tilt angles, such as from an elevated camera looking down a hallway, size in the image is dominated by depth from the camera, and is linearly proportional to row number in the image. We learn the linear relationship of height and width as a function of image row using iteratively reweighted least squares fitting.

3.2. Learning Intrinsic Shape Classes

Rather than treating all pixels in a rotated and scaled bounding box as foreground, we consider a "soft" segmentation of shape by representing the probability of each pixel being foreground. Specifically, instead of associating the entire bounding box with a single Bernoulli mean parameter representing probability of foreground, we use a mixture of Bernoulli distributions to model the learned shape prototypes that are rectangular patches of spatially varying $\mu(x_i)$ values, one per pixel, learned from a training dataset of observed foreground masks. The μ values are high in areas of the shape patch that often contain foreground pixels, and low in places that often contain background, as visualized in the grayscale shape images in Figures 1 and 6. The mixture model allows more varied and realistic shape prototypes and thus can result in more accurate foreground fitting. We also propose a weighted version of the mixture model to allow for "soft" weighting of pixels whose status is more uncertain due to shape variation.

To learn the shape prototypes from a training video sequence, we first select a random subset of frames labeled with ground truth bounding boxes, then run background subtraction to get fg/bg masks, which are overlaid with the bounding boxes to extract a set of binary shape patterns, each scaled to a standard size. The training samples are a set of binary variables. Denote $\mathbf{X} = \{\mathbf{x}_i, \ldots, \mathbf{x}_N\}$ as the collection of N training shape patterns, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{iD})^T$ (D being the size of the shape pattern). We model $\mathbf{X}$ by a mixture of Bernoulli distributions. Formally, the mixture model is defined as

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k p(x|\boldsymbol{\mu}_k) \tag{3}$$

where K is the number of mixture components, $\boldsymbol{\mu} =$

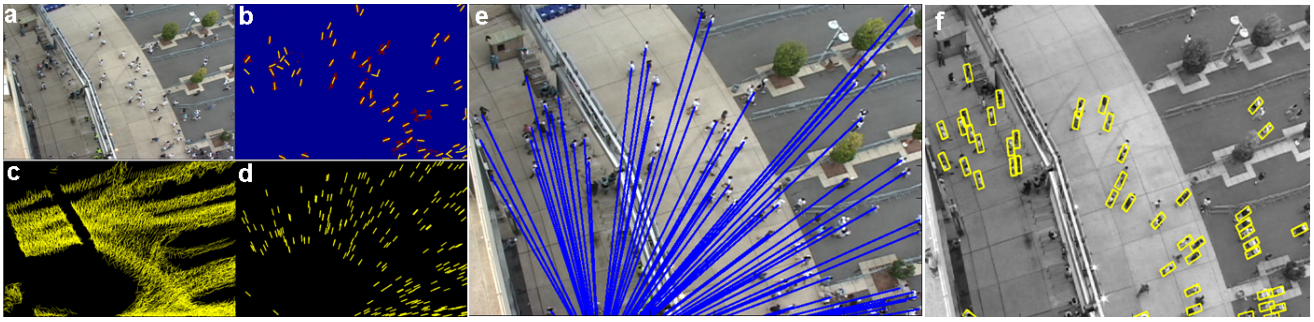


Figure 3. The vertical vanishing point is determined automatically by watching people walk through the scene, thus determining the image orientation of upright people at any location in the image. (a) Image frame. (b) Major axis orientation of each connected component blob in the foreground mask. (c) Axis orientations of blobs extracted from a training sequence. (d) Intersecting inliers found by RANSAC. (e) Lines connecting blob centroids to the computed vertical vanishing point. (f) Mean orientation and scale at each pixel displayed as overlaid rectangles. See text for more details.

$\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$ are the Bernoulli mean parameters, each being a vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_D)^T$, $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$ are the component mixing weights, and $p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{d=1}^{D} \mu_d^{x_d}(1-\mu_d)^{(1-x_d)}$ is the single Bernoulli distribution. We extend the above classic mixture model to a weighted Bernoulli mixture, motivated by the observation that certain pixels vary more across different shapes than other pixels. For example, the boundary pixels of the body shape usually present much bigger variance than the background pixels or the pixels surrounding the center of mass. It is therefore advantageous to make the model spend more effort explaining the higher-variance parts of the shape so that we can get a better shape class model with more distinctive components. For this purpose, we introduce pixel-wise weights $\mathbf{v} = (v_1, \ldots, v_D)^T$ that are estimated by the variance at each pixel across all the training patterns. Hence, $p(\mathbf{x}|\boldsymbol{\mu})$ can be re-written as

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{d=1}^{D} \mu_d^{x_d v_d}(1-\mu_d)^{(1-x_d)v_d} \tag{4}$$

The complete derivation of the above equation is provided on our project web page[1], but the intuition is simple: we can treat the weight as a replication factor; the higher the weights, the more important the pixels and the more times they get duplicated in the sample set. The weighted mixture model gives a very flexible way to incorporate other kinds of prior knowledge about the samples. For example, we can introduce another set of weights on the patterns so reliable patterns get higher weights. Yet another difficulty with mixture models is determining the number of components. We automatically determine the number of components K by imposing a Dirichlet prior over the mixing weights $p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \propto \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$. By setting $\alpha_k \simeq 0$, we have a broad prior that squashes some of the mixing weights to zero, i.e., the Bayesian model automatically trades off between fitting the data and the complexity of the model. In our experiments, we set α_k to be a small positive number. Thresholding on α_k automatically determines the number of intrinsic shapes learned, leading to possibly different numbers of learned intrinsic shapes for different sequences, such as shown in Figures 1 and 6.

The model parameters are estimated by the EM algorithm. We first hypothesize a set of latent variables $\mathbf{z} = (z_{i1}, \ldots, z_{iK})^T$, a set of indicator variables that represent the component membership of each sample. So the set of latent variables over the entire collection is $\mathbf{Z} = \{\mathbf{z}_i, \ldots, \mathbf{z}_N\}$. Now we can form the *complete-data* log likelihood function as

$$\log \mathcal{L}(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\Big\{\log \pi_k + \sum_{d=1}^{D} v_d x_{nd} \log \mu_{kd} + v_d(1-x_{nd})\log(1-\mu_{kd})\Big\} \tag{5}$$

The expected value of the log of the posterior is

$$\mathbf{E}_{\mathbf{Z}}[\log p(\boldsymbol{\mu}, \boldsymbol{\pi}|\mathbf{X}, \mathbf{Z})] = Q(\theta, \theta^{\text{old}}) + \log p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{nk})\Big\{\log \pi_k + \sum_{d=1}^{D} v_d x_{nd} \log \mu_{kd} + v_d(1-x_{nd})\log(1-\mu_{kd})\Big\} + \sum_{k=1}^{K}(\alpha_k - 1)\log \pi_k \tag{6}$$

Because the posterior probability is proportional to the likelihood function times the prior, it can be proven that the prior will have no effect in the E-step, which computes the responsibilities $\gamma(z_{nk}) = E(z_{nk})$. These responsibilities represent the probability that the nth sample belongs to the kth component.

[1] http://vision.cse.psu.edu/projects/mpp/mpp.html

E-step: compute responsibilities

$$\gamma(z_{nk}) = \frac{\pi_k p(x_n \mid \mu_k)}{\sum_{j=1}^{K} \pi_j p(x_n \mid \mu_j)} \quad (7)$$

M-step: update the parameters by maximizing Eqn. 6

$$\mu_{kd} = \frac{\sum_n \gamma(z_{nk}) v_d x_{nd}}{\sum_n \gamma(z_{nk}) v_d} = \frac{\sum_n \gamma(z_{nk}) x_{nd}}{\sum_n \gamma(z_{nk})} \quad (8)$$

$$\pi_k = \frac{\sum_n \gamma(z_{nk}) + \alpha_k - 1}{\sum_n \sum_k \gamma(z_{nk}) + \sum_k \alpha_k - K} \quad (9)$$

$$= \frac{\sum_n \gamma(z_{nk}) + \alpha_k - 1}{N + \sum_k \alpha_k - K} \quad (10)$$

Although it might appear that the pixel-wise weights v_d have canceled out in the update for μ_{kd} , recall that the responsibilities $\gamma(z_{nk})$ are defined in terms of functions $p(x_n|\mu_k)$ (Eqn. 4) that still have the v_d weights in them. The v_d weights therefore do have an effect.

The appeal of the weighted Bernoulli shape mixture model is that the parameter estimation is very efficient and the model itself is flexible enough to be generalized to encode different shapes (as demonstrated in Figures 1 and 6). Note that different numbers of intrinsic shapes are automatically learned through our Bayesian framework with a Dirichlet prior over the mixing weights.

4. Inference

To perform Bayesian inference of the best configuration of person shapes in the image, we define a prior term for the marked point process to combine with the likelihood. Finding the mode of the resulting posterior then provides a MAP estimate over the configuration space.

Recent development of Markov Chain Monte Carlo methods advances the simulation and inference of spatial point processes, which enables us to work on relatively large spatial point patterns [7, 20, 24]. We use reversible jump MCMC to explore configuration sets of different dimensionality (numbers of objects). RJMCMC is an iterative sampling procedure that involves proposing local updates to a current configuration or a reversible jump between configurations of differing dimensions, and then deciding stochastically whether or not to go to the new configuration based on the value of the Metropolis-Hastings acceptance ratio

$$a(x, x') = \min(1, \frac{p(x')}{p(x)} \frac{q(x', x)}{q(x, x')})$$

where x and x' are the current and proposed configurations, $p(\cdot)$ is the MPP posterior distribution evaluated for a given configuration, and $q(a, b)$ is the probability of proposing a transition from a to b.

We use a simple RJMCMC sampler composed of birth, death and update proposals [6]. Each of the proposals is described briefly below:

Birth proposal: A point and mark are proposed and added to the current configuration. We sample the point location according to the foreground mask, which makes it a data-driven proposal. The width, height and orientation of the rectangular mark are sampled from the conditional mark process, represented as Gaussian distributions indexed by the spatial points. An intrinsic Bernoulli shape is chosen uniformly at random (u.a.r) from the set of learned shape prototypes. The reverse move of birth is death.

Death proposal: The death proposal chooses one rectangle at random and removes it from the configuration. If there are no shapes in the configuration, this proposal does nothing. The reverse move is birth.

Update proposal: One rectangle from the configuration is chosen at random and either its location or mark parameters are modified. Modification of location is done as a random walk of the shape center. Modification of the mark is done in two parts: either the mark width, height and orientation dimensions are updated by sampling from the conditional mark process associated with the current location, or the intrinsic Bernoulli shape is updated u.a.r from the shape prototype set. The update proposal is its own reverse move.

For all the experiments, we always start with an empty configuration as the initial state. The RJMCMC procedure is iterated between 500 and 3000 times, with the larger number of iterations being needed when there are more people in the scene. The move probability for birth, death and update proposals are set to be 0.4, 0.2 and 0.4, respectively.

5. Experimental Results

We tested our model on various crowd sequences with different viewing angles, indoor and outdoor scenes, and varying crowd density. We first report the results from quantitative evaluation of the detection algorithm on two human-labeled benchmark sequences: the EU CAVIAR dataset [2] (Figure 6) and the VS-PETS soccer sequence[3] (Figure 5). We tested on a subset of six CAVIAR sequences (5970 frames in total and each frame is of size 288×384) where people are walking in a shopping mall. The soccer sequence was captured in an outdoor football field. There are 2500 frames in total and the image size is 576×720. Players are occluded more often as they run back and forth than the people walking in the hallway.

The RJMCMC procedure iterates 1000 times for the CAVIAR sequence and 3000 times for the soccer sequence. Two key factors of the computing time are the number of iterations and the cost of the likelihood function evaluation in each iteration. The former is affected by the number of people in the scene. More crowded scenes need more iterations. The latter is dominated by the size of each per-

[2] http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

[3] http://www.cvg.cs.rdg.ac.uk/VSPETS/vspets-db.html

son (number of pixels), since we update one person in the configuration set at each iteration. The sampling procedure takes 1.6 seconds to process 500 iterations on a 720×480 frame, 9 people per frame on average, using unoptimized matlab code.

To compare our counting results to the ground truth bounding boxes, we run detections at every 10th frame. Following the evaluation criteria used in [24], the detected object regions are matched to the ground truth box using a greedy algorithm based on the percentage of overlap between a detected foreground region and the ground truth bounding boxes. A correct detection is claimed if the overlap ratio is over 50%. Unmatched detections are considered as false positives. The evaluation results are shown in Table 1. Detailed results for individual CAVIAR sequences are shown in Figure 4.

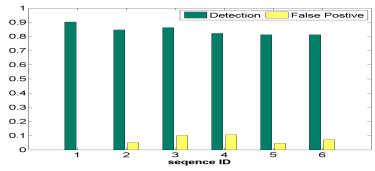


Figure 4. Counting results on six CAVIAR sequences, with 1258 ground truth detections. The average detection rate and false positive rate are 84.3% and 6.2% respectively. Note that people that appear along the image boundary are not counted in this evaluation because the human coders often missed people in those areas.

Dataset	Total # People	Detection Rate	False Positive Rate
CAVIAR	1258	.84	.06
SOCCER	3728	.92	.02

Table 1. Quantitative evaluation of the counting algorithm.

More results for the above sequences showing Bernoulli shapes overlaid are shown in Figures 5 and 6. Videos of the complete set of detection results can be accessed from our project web page. Our detection algorithm works well under reasonable occlusion. The major limitation of the algorithm is its sensitivity to errors in the fg/bg segmentation. In addition, our intrinsic shape learning procedure can only capture shapes that frequently appear in the training set. For example, we cannot detect falling people in the soccer sequence correctly. As a general rule for the success of any learning method, we expect the training data to be representative of the testing data.

6. Conclusion

We propose a marked point process model to detect and count people in crowds. Our model captures the correlations between the mark process (i.e., bounding box size/orientation) and the spatial point process by automatically estimating an extrinsic shape mapping. We also augment the model with intrinsic shape information modeled by a weighted mixture of Bernoulli distributions. The learned shape prototypes are more realistic than simple geometric shapes, which leads to more accurate foreground fitting. Experimental results show that our method can detect varying numbers of pedestrians under different crowd density and reasonable occlusion. In the future, we plan to extend the current model to a spatial-temporal process that can exploit the temporal information in the video. We also plan to apply our shape model to a larger variety of object categories.

7. Acknowledgements

This work was partially funded by the NSF under grants IIS-0729363 and IIS-0535324.

References

[1] A.J.Baddeley and M. vanLieshout. Stochastic geometry models in high-level vision. In K.V.Mardia and G. Kanji, editors, *Statistics and Images*, volume 1, pages 231–256. Abingdon, 1993.

[2] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. In *European Conference on Computer Vision*, volume 2, pages 1–14, October 2008.

[3] A. Baddeley and M. van Lieshout. Recognition of overlapping objects using markov spatial processes. Technical Report BS-R9109, Centrum voor Wiskunde en Informatica, Amsterdam, March 1991.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Vision and Pattern Recognition*, volume 2, pages 886–893, 2005.

[5] J. Descombes, X. Zerubia. Marked point process in image analysis. *IEEE Signal Processing Magazine*, 19(5):77–84, Sept 2002.

[6] C. J. Geyer and J. Mller. Simulation and likelihood inference for spatial point processes. *Scandinavian Journal of Statistics*, 21:359–373, 1994.

[7] P. Green. Mcmc in image analysis. In R. Gilks and Spiegelhalter, editors, *Markov Chain Monte Carlo in Practice*, pages 381–400. Chapman and Hall/CRC, 1995.

[8] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[9] M. Harkness and P. Green. Delayed rejection and reversible jump mcmc for object recognition. In *British Machine Vision Conference*, pages 725–825, 2000.

[10] D. Kong, D. Gray, and H. Tao. A viewpoint invariant approach for crowd counting. In *International Conference on Pattern Recognition*, pages 1187–1190, 2006.

[11] N. Krahnstoever and P. Mendonca. Bayesian autocalibration for surveillance. In *IEEE International Conference on Computer Vision*, pages 1858–1865, Oct 2005.

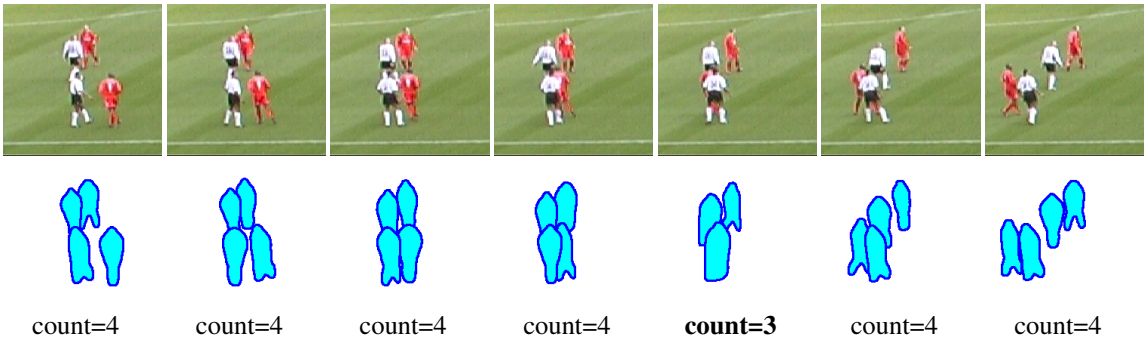


Figure 5. Detailed view of shapes fit to a cluster of four soccer players as they overlap and cross each other. Detection is accurate until a frame where two players completely overlap (incorrect count is marked in bold font). Note that these detections/counts are determined independently for each frame.

Figure 6. Shape fitting to the CAVIAR sequences. Regions where no people are appearing are cropped to save space. The leftmost panel shows the learned intrinsic shapes represented as Bernoulli mixture models. The pixel value represents the value of the Bernoulli mean parameter μ_{kd}.

[12] S. Lin, J. Chen, and H. Chao. Estimation of number of people in crowded scenes using perspective transformation. *IEEE Transactions on Systems, Man and Cybernetics A*, 31(6):645–654, 2001.

[13] F. Lv, T. Zhao, and R. Nevatia. Camera calibration from video of a walking human. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 28(9):1513–1518, Sept 2006.

[14] X. D. M. Ortner and J. Zerubia. A marked point process of rectangles and segments for automatic analysis of digital elevation models. *IEEE Trans Pattern Analysis and Machine Intelligence*, 30(1):105–119, 2008.

[15] A. Marana, L. Costa, R. Lotufo, and S. Velastin. On the efficacy of texture analysis for crowd monitoring. In *Proc. Computer Graphics, Image Processing and Vision*, pages 354–361, 1998.

[16] N. Paragios and V. Ramesh. A mrf-based approach for real-time subway monitoring. In *IEEE Computer Vision and Pattern Recognition*, pages 1034–1040, 2001.

[17] G. Perrin, X. Descombes, and J. Zerubia. A marked point process model for tree crown extraction in plantations. In *IEEE International Conference on Image Processing*, volume 1, pages 661–664, 2005.

[18] V. Rabaud and S. Belongie. Counting crowded moving objects. In *IEEE Computer Vision and Pattern Recognition*, pages 705–711, 2006.

[19] D. Rother, K. A. Patwardhan, and G. Sapiro. What can casual walkers tell us about a 3d scene? In *IEEE International Conference on Computer Vision*, pages 1–8, Oct 2007.

[20] H. Rue and M. Hurn. Bayesian object identification. *Biometrika*, 86(3):649–660, 1999.

[21] H. Rue and A. Syversveen. Bayesian object recognition with Baddeley's delta loss. *Advances in Applied Probability*, 30(1):64–84, 1998.

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

[23] M. van Lieshout. Depth map calculation for a variable number of moving objects using markov sequential object processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1308–1312, 2008.

[24] T. Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 459–466, June 2003.