Counting People With Low-Level Features and Bayesian Regression

Antoni B. Chan, *Member, IEEE*, and Nuno Vasconcelos, *Senior Member, IEEE*

***Abstract*—An approach to the problem of estimating the size of inhomogeneous crowds, which are composed of pedestrians that travel in different directions, without using explicit object segmentation or tracking is proposed. Instead, the crowd is segmented into components of homogeneous motion, using the mixture of dynamic-texture motion model. A set of holistic low-level features is extracted from each segmented region, and a function that maps features into estimates of the number of people per segment is learned with Bayesian regression. Two Bayesian regression models are examined. The first is a combination of Gaussian process regression with a compound kernel, which accounts for both the global and local trends of the count mapping but is limited by the real-valued outputs that do not match the discrete counts. We address this limitation with a second model, which is based on a Bayesian treatment of Poisson regression that introduces a prior distribution on the linear weights of the model. Since exact inference is analytically intractable, a closed-form approximation is derived that is computationally efficient and kernelizable, enabling the representation of nonlinear functions. An approximate marginal likelihood is also derived for kernel hyperparameter learning. The two regression-based crowd counting methods are evaluated on a large pedestrian data set, containing very distinct camera views, pedestrian traffic, and outliers, such as bikes or skateboarders. Experimental results show that regression-based counts are accurate regardless of the crowd size, outperforming the count estimates produced by state-of-the-art pedestrian detectors. Results on 2 h of video demonstrate the efficiency and robustness of the regression-based crowd size estimation over long periods of time.**

***Index Terms*—Bayesian regression, crowd analysis, Gaussian processes, Poisson regression, surveillance.**

I. 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, urban planning, or advertising. From a technological standpoint, computer vision solutions typically focus on detecting, tracking, and analyzing individuals (e.g., finding and tracking a person walking in a parking lot or identifying the interaction between two people). While there has been some success with this type of "individual-centric" surveillance, it is not scalable to scenes with large crowds, where each person is depicted by a few image pixels, people occlude each other in complex ways, and the number of targets to track is overwhelming. Nonetheless, there are many problems in 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 or globally, e.g., monitoring of traffic flows, detection of disturbances in public spaces, detection of highway speeding, or estimation of crowd sizes. By definition, these tasks are based on either properties of the crowd as a whole or 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.

An example surveillance task that can be solved by a "crowd-centric" approach is that of pedestrian counting. Yet, it is frequently addressed with "individual-centric" methods: detect the people in the scene [1]–[6], track them over time [3], [7]–[9], and count the number of tracks. The problem is that, as the crowd becomes larger and denser, both individual detection and tracking become close to impossible. In contrast, a "crowd-centric" approach analyzes global low-level features extracted from crowd imagery to produce accurate counts. While a number of "crowd-centric" counting methods have been previously proposed [10]–[16], they have not fully established the viability of this approach. This has a multitude of reasons: from limited applications to indoor environments with controlled lighting (e.g., subway platforms) [10]–[13], [15] to ignoring crowd dynamics (i.e., treating people moving in different directions as the same) [10]–[14], [16], to assumptions of homogeneous crowd density (i.e., spacing between people) [15], to measuring a surrogate of the crowd size (e.g., crowd density or percent crowding) [10], [11], [15], to questionable scalability to scenes involving more than a few people [16], and to limited experimental validation of the proposed algorithms [10]–[12], [14], [15].

Unlike these proposals, we show that there is 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 subcomponents with different dynamics and appears in unconstrained outdoor environments, such as that of Fig. 1. In fact, we argue that when a "crowd-centric" approach is considered, the problem appears to become simpler. We simply segment the

Manuscript received March 01, 2011; revised July 24, 2011; accepted September 22, 2011. Date of publication October 19, 2011; date of current version March 21, 2012. This work was supported in part by the NSF under Grant CCF-0830535 and Grant IIS-0812235 and in part by the Research Grants Council of the Hong Kong Special Administrative Region of China under Grant 9041552 (CityU 110610). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Patrick Flynn.

A. B. Chan is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong.

N. Vasconcelos is with the Department of Electrical and Computer Engineering, University of California, San Diego, San Diego, CA 92093-0407 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2172800

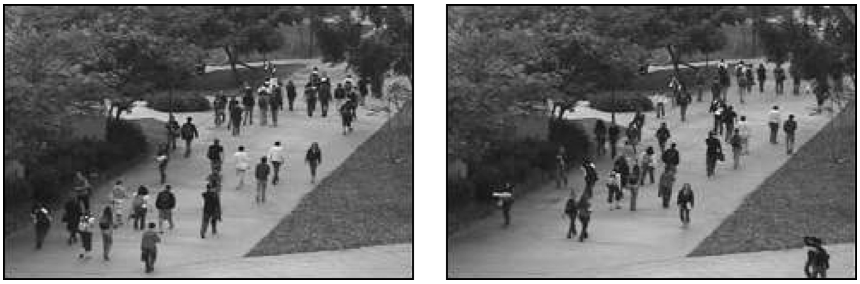

Fig. 1. Scenes containing a sizable crowd with inhomogeneous dynamics due to pedestrian motion in different directions.

crowd into subparts of interest (e.g., groups of people moving in different directions), extract a set of holistic features from each segment, and estimate the crowd size with a suitable regression function [17]. By bypassing intermediate processing stages, such as people detection or tracking, which are susceptible to occlusion problems, the proposed approach produces robust and accurate crowd counts, even when the crowd is large and dense.

One important aspect of regression-based counting is the choice of the regression function used to map segment features into crowd counts. One possibility is to rely on classical regression methods, such as linear or piecewise linear, regression, and least squares fits [18]. These methods are not very robust to outliers and nonlinearities and are prone to overfitting when the feature space is high-dimensional or when there are little training data. In these cases, better performance can usually be obtained with more recent methods, such as Gaussian process regression (GPR) [19]. GPR has several advantages, including adaptation to nonlinearities with kernel functions, robust selection of kernel hyperparameters via maximization of marginal likelihoods (namely type-II maximum likelihood), and a Bayesian formalism for inference that enables better generalization from small training sets. However, the main limitation of GPR-based counting is that it relies on a continuous real-valued function to map visual features into discrete counts. This reduces the effectiveness of Bayesian inference. For example, the predictive distribution does not assign zero probability to noninteger, or even negative, counts. In result, there is a need for suboptimal postprocessing operations, such as quantization and truncation. Furthermore, continuous crowd estimates increase the complexity of subsequent statistical inference, e.g., graphical models that identify dependence between counts measured at different nodes of a camera network. Since this type of inference is much simpler for discrete variables, the continuous representation that underlies GPR adds undue complexity.

A standard method for learning mappings into the set of nonnegative integers is Poisson regression [20], which models the output variable as a Poisson distribution with a log-arrival rate that is a linear function of the input feature vector. To obtain a Bayesian model, a popular extension of Poisson regression is to adopt a hierarchical model, where the log-arrival rate is modeled with a GP prior [21]–[23]. However, due to the lack of conjugacy between the Poisson and the GP, exact inference is analytically intractable. Existing models [21]–[23] rely on Markov-chain Monte Carlo (MCMC) methods, which limit these hierarchical models to small data sets. In this paper, we take a different approach and directly analyze Poisson regression from a Bayesian perspective, by imposing a Gaussian prior on the weights of the linear log-arrival rate [24]. We denote this model as Bayesian Poisson regression (BPR). While exact inference is still intractable, it is shown that effective closed-form approximations can be derived. This leads to a regression algorithm that is much more efficient than those previously available [21]–[23].

The contributions of this paper are threefold, spanning open questions in computer vision and machine learning. First, a "crowd-centric" methodology for estimating the sizes of crowds moving in different directions, which does not depend on object detection or feature tracking, is presented. Second, a Bayesian regression procedure is derived for the estimation of counts, which is a Bayesian extension of Poisson regression. A closed-form approximation to the predictive distribution, which can be kernelized to handle nonlinearities, is derived, together with an approximate procedure for optimizing the hyperparameters of the kernel function, under the Type-II maximum marginal likelihood criteria. It is also shown that the proposed approximation to BPR is related to a GPR with a specific noise term. Third, the proposed crowd counting approach is validated on two large data sets of pedestrian imagery, and its robustness demonstrated through results on 2 hours of video. To our knowledge, this is the first pedestrian counting system that accounts for multiple pedestrian flows and successfully operates continuously in an outdoor unconstrained environment for such periods of time.

This paper is organized as follows. Section II reviews related work in crowd counting. GPR is discussed in Section III, and BPR is proposed in Section IV. Section V introduces a crowd counting system based on motion segmentation and Bayesian regression. Finally, experimental results on the application of Bayesian regression to the crowd counting problem are presented in Section VI.

II. Related Work

Current solutions to crowd counting follow three paradigms: 1) pedestrian detection; 2) visual feature trajectory clustering; and 3) regression. Pedestrian detection algorithms can be based on boosting appearance and motion features [1], Bayesian model-based segmentation [2], [3], histogram-of-gradients [25], or integrated top-down and bottom-up processing [4]. Because they detect whole pedestrians, these methods are not very effective in densely crowded scenes involving significant occlusion. This problem has been addressed to some extent by the development of part-based detectors [5], [6], [26], [27]. Detection results can be further improved by tracking detections between multiple frames, e.g., via a Bayesian approach [28] or boosting [29], or by using stochastic spatial models to simultaneously detect and count people as foreground shapes [30].

The second paradigm consists of identifying and tracking visual features over time. Feature trajectories that exhibit coherent motion are clustered, and the number of clusters is used as an estimate of the number of moving subjects. Examples of this formulation include [7], which uses the KLT tracker and agglomerative clustering, and [8], which relies on an unsupervised Bayesian approach. Counting of feature trajectories has two disadvantages. First, it requires sophisticated trajectory management (e.g., handling broken feature tracks due to occlusions or measuring similarities between trajectories of different length) [31]. Second, in crowded environments, it is frequently the case that coherently moving features do not belong to the same person. Hence, equating the number of people to the number of trajectory clusters can be quite error prone.

Regression-based 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 [10], [11], [13], edge count [11]–[13], or texture [15]; and 3) estimating the crowd density or crowd count with a regression function, e.g., linear [10], [13], piecewise linear [12], or neural networks [11], [15]. In recent years, regression-based counting has also been applied to outdoor scenes. For example, Kong *et al.* [14] apply neural networks to the histograms of foreground segment areas and edge orientations. Dong *et al.* [16] 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 six). Cong *et al.* [32] count the number of people crossing a line of interest using flow vectors and dynamic mosaics. Lempitsky and Zisserman [33] proposes a supervised learning framework, which estimates an image density whose integral over a region of interest (ROI) yields the count. The main contributions of this paper, with respect to previous approaches to regression-based counting, are fourfold: 1) integration of regression and robust motion segmentation, which enables counts for crowds moving in different directions (e.g., traveling into or out of a building); 2) integration of suitable features and Bayesian nonlinear regression, which enables accurate counts in densely crowded scenes; 3) introduction of a Bayesian model for discrete regression, which is suitable for crowd counting; and 4) demonstration that the proposed algorithms can robustly operate on video of unconstrained outdoor environments, through validation on a large data set containing 2 hours of video.

Regarding Bayesian regression for discrete counts, Diggle *et al.* [21]–[23] and Adams *et al.* [34] propose hierarchical Poisson models, where the log-arrival rate is modeled with a GP prior. Inference is approximated with MCMC, which has been noted to exhibit slow mixing times and poor convergence properties [21]. Alternatively, El-Sayyad [35] directly performs a Bayesian analysis of standard Poisson regression by adding a Gaussian prior on the linear weights and proposes a Gaussian approximation to the posterior weight distribution. In this paper, we extend [35] in three ways: 1) we derive a Gaussian posterior that can handle observations of zero count; 2) we derive a closed-form predictive count distribution; and 3) we kernelize the regression function, thus modeling nonlinear log-arrival rates. Our final contribution is a kernelized closed-form efficient approximation to BPR. Finally, a regression task similar to counting is ordinal regression, which learns a mapping to an ordinal scale (ranking or ordered set), e.g., letter grades. A Bayesian version of ordinal regression using GP priors was proposed in [36]. However, ordinal regression cannot elegantly be used for counting; the ordinal scale is fixed upon training, and hence, it cannot predict counts outside of the training set.

With respect to our previous work, our initial solution to crowd counting using GPR was presented in [17], and BPR was proposed in [24]. The contributions of this paper, with respect to our previous work, are fourfold: 1) we present the complete derivation for BPR, which was shortened in [24]; 2) we derive BPR so that it handles zero count observations; 3) we validate Bayesian regression-based counting on a larger data set and from two viewpoints (Chan *et al.* [17], [24] only tested one viewpoint); and 4) we provide an in-depth comparison between regression-based counting and counting using person detection.

III. Gaussian Process Regression

Fig. 1 shows examples of a crowded scene on a pedestrian walkway. We assume that the camera is part of a permanent surveillance installation; hence, the viewpoint is fixed. The goal of crowd counting is to estimate the number of people moving in each direction. The basic idea is that, given a segmentation into the two crowd subcomponents, certain low-level global features extracted from each crowd segment are good predictors of the number of people in that segment. Intuitively, assuming proper normalization for the scene perspective, one such feature is the area of the crowd segment (number of segment pixels). Fig. 2(a) plots the segment area versus the crowd size, along with the least squares fit by a line. Note that, while there is a global linear trend relating the two variables, the data have local deviations from this linear trend, due to confounding factors such as occlusion. This suggests that additional features are needed to accurately model crowd counts, along with a regression framework that can accommodate the local nonlinearities.

One possibility to implement this regression is to rely on GPR [19]. This is a Bayesian approach to the prediction of a real-valued function $f(\mathbf{x})$ of a feature vector $\mathbf{x} \in \mathbb{R}^d$, from a training sample. Let $\phi(\mathbf{x})$ be a high-dimensional feature transformation of $\mathbf{x}$, $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$. Consider the case where $f(\mathbf{x})$ is linear in the transformation space and the target count y modeled as

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w}, \qquad y = f(\mathbf{x}) + \epsilon \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^D$, and the observation noise is assumed independent identically distributed (i.i.d.), and Gaussian $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. The Bayesian formulation requires a prior distribution on the weights, which is assumed Gaussian $\mathbf{w} \sim \mathcal{N}(0, \Sigma_p)$ of covariance Σ_p.

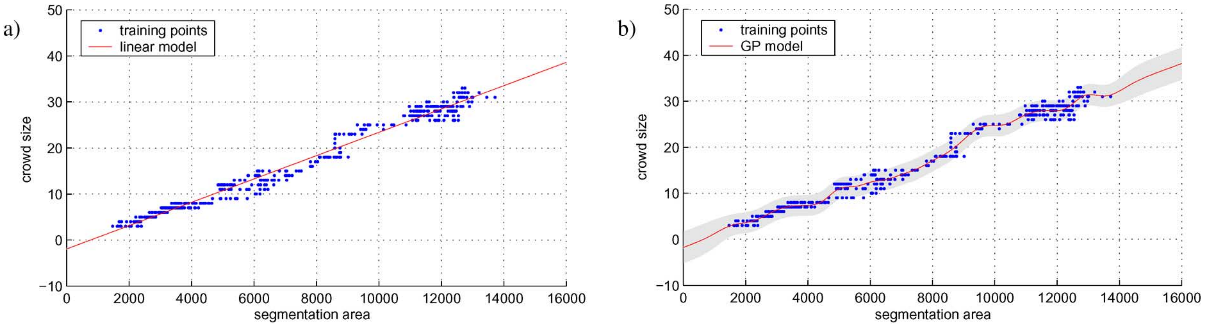

Fig. 2. Correspondence between crowd size and segment area. (a) Line learned with least squares regression. (b) Nonlinear function learned with GPR. The two standard deviations error bars are plotted (gray area).

A. Bayesian Prediction

Let $X = [\mathbf{x}_1, \cdots \mathbf{x}_N]$ be the matrix of observed feature vectors $\mathbf{x}_i$, and let $\mathbf{y} = [y_1 \ \cdots \ y_N]^T$ be the vector of the corresponding counts y_i. The posterior distribution of the weights $\mathbf{w}$, given the observed data $\{X, \mathbf{y}\}$, is given by Bayes' rule $p(\mathbf{w}|X,\mathbf{y}) = (p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w}) / \int p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})d\mathbf{w})$. Given the novel input $\mathbf{x}_*$, the predictive distribution for $f_* = f(\mathbf{x}_*)$ is the average overall possible model parameterizations [19]

$$p(f_*|\mathbf{x}_*, X, \mathbf{y}) = \int p(f_*|\mathbf{x}_*, \mathbf{w})p(\mathbf{w}|X,\mathbf{y})d\mathbf{w} \quad (2)$$

$$= \mathcal{N}(f_*|\mu_*, \sigma_*^2) \quad (3)$$

where the predictive mean and covariance are

$$\mu_* = \mathbf{k}_*^T \left(K + \sigma_n^2 I\right)^{-1} \mathbf{y} \quad (4)$$

$$\sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T \left(K + \sigma_n^2 I\right)^{-1} \mathbf{k}_*. \quad (5)$$

K is the kernel matrix with entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, and $k_* = [k(\mathbf{x}_*, \mathbf{x}_1) \cdots k(\mathbf{x}_*, \mathbf{x}_N)]^T$. The kernel function is $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \Sigma_p \phi(\mathbf{x}')$; hence, the predictive distribution only depends on inner products between inputs $\mathbf{x}_i$.

B. Compound Kernel Functions

The class of functions that can be approximated by GPR depends on the covariance or the kernel function employed. For example, the linear kernel $k_l(\mathbf{x}, \mathbf{x}') = \theta_1^2(\mathbf{x}^T\mathbf{x}' + 1)$ leads to standard Bayesian linear regression, whereas a squared-exponential (RBF) kernel $k_r(\mathbf{x}, \mathbf{x}') = \theta_1^2 e^{-(1/\theta_2^2)\|\mathbf{x}-\mathbf{x}'\|^2}$ yields Bayesian regression for locally smooth infinitely differentiable functions. As shown in Fig. 2(a), the segment area exhibits a linear trend with the crowd size, with some local nonlinearities due to occlusions and segmentation errors. To model the dominant linear trend, as well as these nonlinear effects, we can use a compound kernel with linear and RBF components as follows:

$$k_{LR}(\mathbf{x}_i, \mathbf{x}_j) = \theta_1^2 \left(\mathbf{x}_i^T \mathbf{x}_j + 1\right) + \theta_2^2 e^{-\frac{1}{2\theta_3^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2}. \quad (6)$$

Fig. 2(b) shows an example of a GPR function adapting to local nonlinearities using the linear-RBF compound kernel. The inclusion of additional features (particularly texture features) can make the dominant trend nonlinear. In this case, a kernel with two RBF components is more appropriate, as shown in

$$k_{RR}(\mathbf{x}_i, \mathbf{x}_j) = \theta_1^2 e^{-\frac{1}{2\theta_2^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2} + \theta_3^2 e^{-\frac{1}{2\theta_4^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2}. \quad (7)$$

The first RBF has a larger scale parameter θ_2 and models the overall trend, whereas the second relies on a smaller scale parameter θ_4 to model local nonlinearities.

The kernel hyperparameters θ_i can be estimated from a training sample by Type-II maximum likelihood, which maximizes the marginal likelihood of the training data $\{X, \mathbf{y}\}$

$$\log p(\mathbf{y}|X, \theta) = \log \int p(\mathbf{y}|\mathbf{w}, X, \theta)p(\mathbf{w}|\theta)d\mathbf{w} \quad (8)$$

$$= -\frac{1}{2}\mathbf{y}^T K_y^{-1}\mathbf{y} - \frac{1}{2}\log|K_y| - \frac{N}{2}\log 2\pi \quad (9)$$

where $K_y = K + \sigma_n^2 I$, with respect to the parameters θ, e.g., using standard gradient ascent methods. Details of this optimization can be found in [19,Chapter 5].

IV. Bayesian Poisson Regression

While GPR is a Bayesian framework for regression problems with real-valued output variables, it is not a natural regression formulation when the outputs are nonnegative integers, i.e., $y \in \mathbb{Z}_+ = \{0, 1, 2, \cdots\}$, as is the case for counts. A typical solution is to model the output variable as Poisson or negative binomial (NB), with an arrival-rate parameter that is a function of the input variables, resulting in the standard Poisson regression or NB regression [20]. Although both these methods model counts, they do not support Bayesian inference, i.e., do not consider the weight vector β as a random variable. This limits their generalization from small training samples and prevents a principled probabilistic approach to learning hyperparameters in a kernel formulation.

In this section, we propose a Bayesian model for count regression. We start from the standard Poisson regression model, where the input is $\mathbf{x} \in \mathbb{R}^d$, and the output variable y is Poisson distributed, with a log-arrival rate that is a linear function in the transformation space $\phi(\mathbf{x}) \in \mathbb{R}^D$, i.e.,

$$\nu(\mathbf{x}) = \phi(\mathbf{x})^T\beta, \quad \lambda(\mathbf{x}) = e^{\nu(\mathbf{x})}, \quad y \sim \text{Poisson}\left(\lambda(\mathbf{x})\right) \quad (10)$$

where $\nu(\mathbf{x})$ is the log of the arrival rate, $\lambda(\mathbf{x})$ the arrival rate (or mean of y), and $\beta \in \mathbb{R}^D$ is the weight vector. The likelihood of y given observation $\mathbf{x}$ is

$$p(y|\mathbf{x}, \beta) = \frac{e^{-\lambda(\mathbf{x})}\lambda(\mathbf{x})^y}{y!}.$$

We assume a Gaussian prior on the weight vector $\beta \sim \mathcal{N}(0, \Sigma_p)$. The posterior distribution of β, given a training sample $\{X, \mathbf{y}\}$, is given by Bayes' rule as follows:

$$p(\beta|X, \mathbf{y}) = \frac{p(\mathbf{y}|X, \beta)p(\beta)}{\int p(\mathbf{y}|X, \beta)p(\beta)d\beta}. \tag{11}$$

Due to the lack of conjugacy between the Poisson likelihood and the Gaussian prior, (11) does not have a closed-form expression; therefore, an approximation is necessary.

A. Approximate Posterior Distribution

We first derive a closed-form approximation to the posterior distribution in (11), which is based on the approximation of [35]. Consider the data likelihood of a training set $\{X, \mathbf{y}\}$ as follows:

$$p(\mathbf{y}|X, \beta) = \prod_{i=1}^{N} \frac{1}{y_i!} e^{\nu(\mathbf{x}_i)y_i} e^{-e^{\nu(\mathbf{x}_i)}} \tag{12}$$

$$= \prod_{i=1}^{N} \left[\frac{e^{\nu(\mathbf{x}_i)(y_i+c)} e^{-e^{\nu(\mathbf{x}_i)}}}{\Gamma(y_i + c)} \right] e^{-c\nu(\mathbf{x}_i)} \frac{\Gamma(y_i + c)}{y_i!} \tag{13}$$

where $c \geq 0$ is a constant. The approximation is based on two facts. First, the term in the square brackets is the likelihood of the data under a log-gamma distribution of parameters $(y+c, 1)$, i.e., $\nu \sim$ Log Gamma$(y + c, 1)$ where

$$p(\nu|y + c, 1) = \frac{1}{\Gamma(y + c)} e^{\nu(y+c)} e^{-e^{\nu}}. \tag{14}$$

A log-gamma random variable ν is the log of a gamma random variable λ, where $\nu = \log \lambda$. This implies that λ is gamma distributed with parameters $(y+c, 1)$. Second, for a large number of arrivals k, the log-gamma is closely approximated by a Gaussian [35], [37], [38], i.e.,

$$\text{Log Gamma}(k, \theta) \approx \mathcal{N}(\mu, \sigma^2) \tag{15}$$

where the parameters are related by

$$k = \sigma^{-2}, \theta = \sigma^2 e^{\mu} \quad \iff \quad \sigma^2 = k^{-1}, \mu = \log(k\theta). \tag{16}$$

Hence, (14) can be approximated as

$$p(\nu|y + c, 1) \approx \mathcal{N}\left(\nu| \log(y + c), (y + c)^{-1}\right). \tag{17}$$

This is illustrated in Fig. 3, which depicts the accuracy of the approximation for different values of $y + c$. Applying (17) to replace the square-bracket term in (13) and defining $\Phi = [\phi(\mathbf{x}_1) \cdots \phi(\mathbf{x}_N)]$

$$p(\mathbf{y}|X, \beta) \approx \prod_{i=1}^{N} \left[\mathcal{N}\left(\nu(\mathbf{x}_i)| \log(y_i + c), (y_i + c)^{-1}\right)\right] \cdot e^{-c\nu(\mathbf{x}_i)} \frac{\Gamma(y_i + c)}{y_i!} \tag{18}$$

$$= \frac{e^{-\frac{1}{2}\|\Phi^T\beta - \mathbf{s}\|^2_{\Sigma_y} - c\mathbf{1}^T\Phi^T\beta}}{(2\pi)^{\frac{N}{2}} |\Sigma_y|^{\frac{1}{2}}} \prod_{i=1}^{N} \frac{\Gamma(y_i + c)}{y_i!} \tag{19}$$

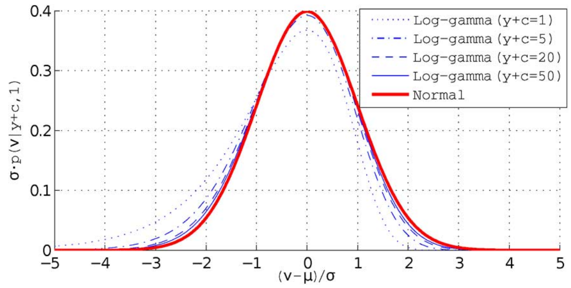

Fig. 3. Gaussian approximation of the log-gamma distribution for different values of $y + c$. The plot is normalized so that the distributions have zero mean and unit variance.

where $\Sigma_y = \text{diag}([1/(y_1+c) \cdots 1/(y_N+c)])$, and $\mathbf{s} = \log(\mathbf{y}+c)$ is the elementwise logarithm of $\mathbf{y} + c$. Substituting into (11)

$$\log p(\beta|X, \mathbf{y}) \propto \log p(\mathbf{y}|X, \beta) + \log p(\beta) \tag{20}$$

$$\approx -\frac{1}{2}\|\Phi^T\beta - \mathbf{s}\|^2_{\Sigma_y} - c\mathbf{1}^T\Phi^T\beta - \frac{1}{2}\|\beta\|^2_{\Sigma_p} \tag{21}$$

where we have ignored terms independent of β. Expanding the norm terms yields

$$\log p(\beta|X, \mathbf{y}) \propto -\frac{1}{2}\left(\beta^T\Phi\Sigma_y^{-1}\Phi^T\beta - 2\beta^T\Phi\Sigma_y^{-1}\mathbf{s} + \mathbf{s}^T\Sigma_y^{-1}\mathbf{s}\right) - c\mathbf{1}^T\Phi^T\beta - \frac{1}{2}\beta^T\Sigma_p^{-1}\beta \tag{22}$$

$$\propto -\frac{1}{2}\left[\beta^T\left(\Phi\Sigma_y^{-1}\Phi^T + \Sigma_p^{-1}\right)\beta - 2\beta^T\left(\Phi\Sigma_y^{-1}\mathbf{s} - c\Phi\mathbf{1}\right)\right] \tag{23}$$

$$= \frac{1}{2}\left(\beta^T\left(\Phi\Sigma_y^{-1}\Phi^T + \Sigma_p^{-1}\right)\beta - 2\beta^T\Phi\Sigma_y^{-1}\mathbf{t}\right) \tag{24}$$

where $\mathbf{t} = \mathbf{s} - c\Sigma_y\mathbf{1}$ has elements $t_i = \log(y_i+c) - c/(y_i+c)$. Finally, by completing the square, the posterior distribution is approximately Gaussian, i.e.,

$$p(\beta|X, \mathbf{y}) \approx \mathcal{N}(\beta|\hat{\mu}_\beta\hat{\Sigma}_\beta) \tag{25}$$

with mean and variance

$$\hat{\mu}_\beta = \left(\Phi\Sigma_y^{-1}\Phi^T + \Sigma_p^{-1}\right)^{-1}\Phi\Sigma_y^{-1}\mathbf{t} \tag{26}$$

$$\hat{\Sigma}_\beta = \left(\Phi\Sigma_y^{-1}\Phi^T + \Sigma_p^{-1}\right)^{-1}. \tag{27}$$

Note that setting $c = 0$ will yield the original posterior approximation in [35]. Constant c acts as a parameter that controls the smoothness of the approximation around $y = 0$, avoiding the logarithm of or division by zero. In the experiments, we set this parameter to $c = 1$.

B. Bayesian Prediction

Given a novel observation $\mathbf{x}_*$, we start by considering the predicted log-arrival rate $\nu_* = \phi(\mathbf{x}_*)^T\beta$. It follows from (25) that the posterior distribution of ν_* is approximately Gaussian:

$$p(\nu_*|\mathbf{x}_*, X, \mathbf{y}) \approx \mathcal{N}\left(\nu_*|\hat{\mu}_\nu, \hat{\sigma}^2_\nu\right) \tag{28}$$

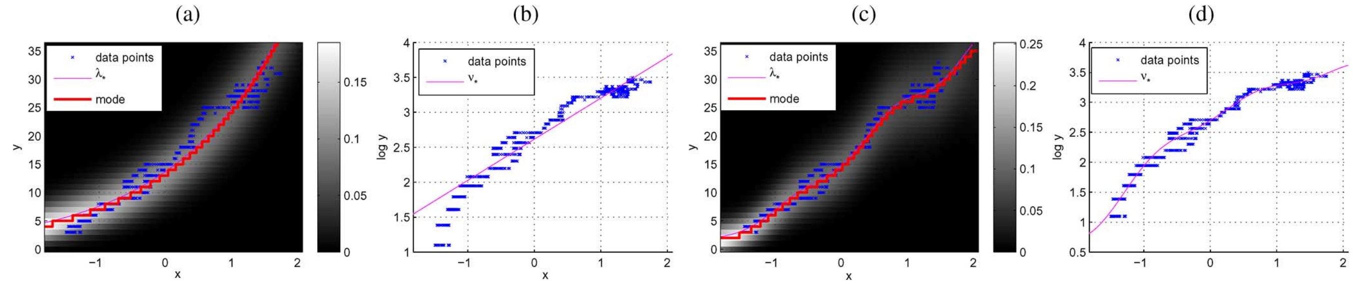

Fig. 4. BPR with (a) linear and (c) RBF kernels. The mean parameter $e^{\hat{\mu}_\nu}$ and the mode are shown superimposed on the NB predictive distribution. The corresponding log-arrival rate functions are shown in (b) and (d).

with mean and variance

$$\hat{\mu}_\nu = \phi(\mathbf{x}_*)^T \left(\Phi\Sigma_y^{-1}\Phi^T + \Sigma_p^{-1}\right)^{-1} \Phi\Sigma_y^{-1}\mathbf{t} \qquad (29)$$

$$\hat{\sigma}_\nu^2 = \phi(\mathbf{x}_*)^T \left(\Phi\Sigma_y^{-1}\Phi^T + \Sigma_p^{-1}\right)^{-1} \phi(\mathbf{x}_*). \qquad (30)$$

Applying the matrix inversion lemma, $\hat{\sigma}_\nu^2$ can be rewritten in terms of the kernel function

$$\begin{aligned}\hat{\sigma}_\nu^2 &= \phi(\mathbf{x}_*)^T \left(\Sigma_p - \Sigma_p\Phi(\Phi^T\Sigma_p\Phi + \Sigma_y)^{-1}\Phi^T\Sigma_p\right)\phi(\mathbf{x}_*) \\ &= k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T(K + \Sigma_y)^{-1}\mathbf{k}_* \qquad (31)\end{aligned}$$

where $k(\cdot,\cdot)$, K, and $\mathbf{k}_*$ are defined, as in Section III-A. Using (42) from the Appendix, the posterior mean $\hat{\mu}_\nu$ can also be rewritten in terms of the kernel function

$$\begin{aligned}\hat{\mu}_\nu &= \phi(\mathbf{x}_*)^T\Sigma_p\Phi(\Phi^T\Sigma_p\Phi + \Sigma_y)^{-1}\mathbf{t} \qquad (32) \\ &= \mathbf{k}_*^T(K + \Sigma_y)^{-1}\mathbf{t}. \qquad (33)\end{aligned}$$

Since the posterior mean and variance of ν_* depend only on the inner product between the inputs, we can apply the "kernel trick" to obtain nonlinear log-arrival rate functions.

The predictive distribution for y_* is

$$p(y_*|\mathbf{x}_*, X, \mathbf{y}) = \int p(y_*|\nu_*)p(\nu_*|\mathbf{x}_*, X, \mathbf{y})d\nu_* \qquad (34)$$

where $p(y_*|e^{\nu_*})$ is a Poisson distribution of the arrival rate $\lambda_* = e^{\nu_*}$. While this integral does not have analytic solution, a closed-form approximation is possible. Since ν_* is approximately Gaussian, it follows from (15) and (16) that ν_* is well approximated by a log-gamma distribution. From $\nu_* = \log\lambda_*$, it then follows that λ_* is approximately gamma distributed:

$$\lambda_*|\mathbf{x}_*, X, \mathbf{y} \sim \text{Gamma}\left(\hat{\sigma}_\nu^{-2}, \hat{\sigma}_\nu^2 e^{\hat{\mu}_\nu}\right).$$

Note that the expected time λ_* between arrivals of the Poisson process is modeled as the time between $\hat{\sigma}_\nu^{-2}$ arrivals of a Poisson process of rate $\hat{\sigma}_\nu^2 e^{\hat{\mu}_\nu}$. Hence, $\lambda_* \approx e^{\hat{\mu}_\nu}$, which is a sensible approximation. (34) can then be rewritten as

$$p(y_*|\mathbf{x}_*, X, \mathbf{y}) = \int_0^\infty p(y_*|\lambda_*)p(\lambda_*|\mathbf{x}_*, X, \mathbf{y})d\lambda_* \qquad (35)$$

where $p(y_*|\lambda_*)$ is a Poisson distribution and $p(\lambda_*|\mathbf{x}_*, X, \mathbf{y})$ is a gamma distribution. Since the latter is the conjugate prior for the former, the integral has an analytical solution, which is an NB, i.e.,

$$p(y_*|\mathbf{x}_*, X, \mathbf{y}) = \frac{\Gamma\left(y_* + \hat{\sigma}_\nu^{-2}\right)}{\Gamma(y_* + 1)\Gamma\left(\hat{\sigma}_\nu^{-2}\right)}(\hat{p})^{\hat{\sigma}_\nu^{-2}}(1 - \hat{p})^{y_*}, \qquad (36)$$

$$\hat{p} = \frac{\hat{\sigma}_\nu^{-2}}{\hat{\sigma}_\nu^{-2} + \exp(\hat{\mu}_\nu)}. \qquad (37)$$

In summary, the predictive distribution of y_* can be approximated by an NB as follows:

$$y_*|\mathbf{x}_*, X, \mathbf{y} \sim \text{NegBin}\left(e^{\hat{\mu}_\nu}, \hat{\sigma}_\nu^2\right) \qquad (38)$$

of mean $e^{\hat{\mu}_\nu}$ and scale $\hat{\sigma}_\nu^2$, given by (29). The prediction variance is $\text{var}(y_*) = e^{\hat{\mu}_\nu}(1 + \hat{\sigma}_\nu^2 e^{\hat{\mu}_\nu})$ and grows proportionally to the variance of ν_*. This is sensible since uncertainty in the prediction of ν_* is expected to increase the uncertainty of the count prediction y_*. In the ideal case of no uncertainty ($\hat{\sigma}_\nu^2 = 0$), the NB reduces to a Poisson distribution with both mean and variance of $e^{\hat{\mu}_\nu}$. Thus, a useful measure of uncertainty for the prediction y_* is the square root of this "extra" variance (i.e., overdispersion), i.e., $\text{unc}(y_*) = \hat{\sigma}_\nu e^{\hat{\mu}_\nu}$. Finally, the mode of y_* is adjusted downward depending on the amount of overdispersion, i.e., $\text{mode}(y) = \begin{cases} \lfloor(1 - \hat{\sigma}_\nu^2)e^{\hat{\mu}_\nu}\rfloor, & \hat{\sigma}_\nu^2 < 1 \\ 0, & \hat{\sigma}_\nu^2 \geq 1 \end{cases}$, where $\lfloor\cdot\rfloor$ is the floor function.

C. Learning the Kernel Hyperparameters

The hyperparameters θ of kernel $k(\mathbf{x}, \mathbf{x}')$ can be estimated by maximizing the marginal likelihood $p(\mathbf{y}|X, \theta)$. Using the log-gamma approximation in (19), $p(\mathbf{y}|X, \theta)$ is approximated in closed form with (see Appendix for derivation)

$$\log p(\mathbf{y}|X, \theta) \propto -\frac{1}{2}\log|K + \Sigma_y| - \frac{1}{2}\mathbf{t}^T(K + \Sigma_y)^{-1}\mathbf{t}. \quad (39)$$

Fig. 4 presents two examples of BPR learning using the linear and RBF kernels. The predictive distributions are plotted in Fig. 4(a) and (c), and the corresponding log-arrival rate functions are plotted in Fig. 4(b) and (d). While the linear kernel can only account for exponential trends in the data, the RBF kernel can easily adapt to the local deviations of the arrival rate.

D. Relationship With GPR

The proposed approximate BPR is closely related to GPR. The equations for $\hat{\mu}_\nu$ and $\hat{\sigma}_\nu^2$ in (31) and (33) are almost iden-

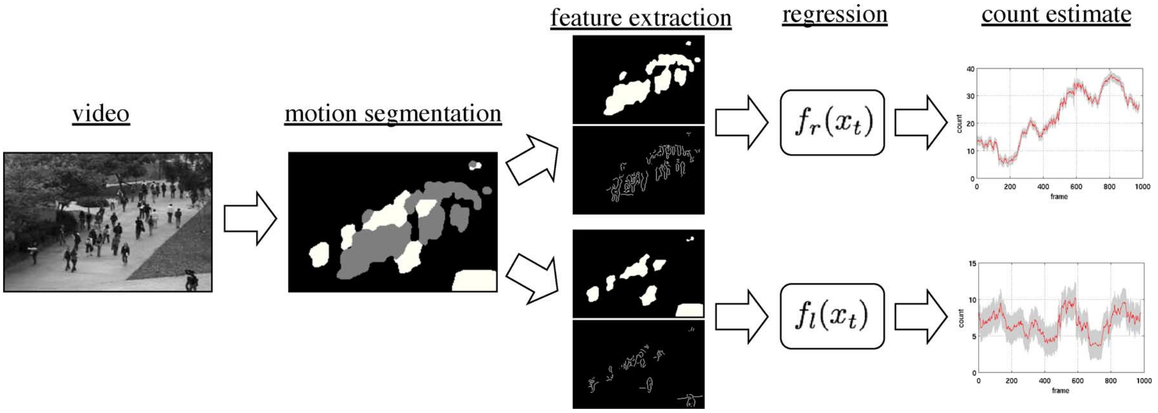

Fig. 5. Crowd counting from low-level features. The scene is segmented into crowds moving in different directions. Features are extracted from each segment and normalized to account for perspective. The number of people in each segment is estimated with Bayesian regression.

tical to those of the GPR predictive distribution in (4) and (5). There are two main differences: 1) the noise term Σ_y of BPR in (31) is dependent on predictions y_i (this is a consequence of assuming a Poisson noise model), whereas the GPR noise term in (5) is i.i.d. ($\sigma_n^2 I$); 2) the predictive mean $\hat{\mu}_\nu$ in (33) is computed with the log counts $\mathbf{t}$ (assuming $c = 0$), rather than the counts $\mathbf{y}$ of GPR (this is due to the fact that BPR predicts log-arrival rates, whereas GPR predicts counts). This suggests the following interpretation for the approximate BPR. Given the observed data $\{X, \mathbf{y}\}$ and novel input $\mathbf{x}_*$, approximate BPR models the predictive distribution of the log-arrival rate ν_* as a GP with non-i.i.d. observation noise of covariance Σ_y. The posterior mean $\hat{\mu}_\nu$ and variance $\hat{\sigma}_\nu^2$ of ν_* then serve as parameters of the predictive distribution of y_*, which is approximated by an NB of mean $e^{\hat{\mu}_\nu}$ and the scale parameter $\hat{\sigma}_\nu^2$. Note that the posterior variance of ν_* is the scale parameter of the NB. Hence, increased uncertainty in the predictions of ν_*, by the GP, translates into increased uncertainty in the prediction of y_*. The approximation to the BPR marginal likelihood in (39) differs from that of the GPR in a similar manner and, hence, has a similar interpretation. In summary, the proposed closed-form approximation to BPR is equivalent to GPR on the log-arrival rate parameter of the Poisson distribution. This GP includes a special noise term, which approximates the uncertainty that arises from the Poisson noise model. Since BPR can be implemented as GPR, the proposed closed-form approximate posterior is more efficient than the Laplace or EP approximations, which both use iterative optimization. In addition, the approximate predictive distribution is also calculated efficiently since it avoids numerical integration. Finally, standard Poisson regression belongs to the family of generalized linear models [39], which is a general regression framework for linear covariate regression problems. Generalized kernel machines and the associated kernel Poisson regression were proposed in [40]. The proposed BPR is a Bayesian formulation of kernel Poisson regression.

V. Crowd Counting Using Low-Level Features and Bayesian Regression

An outline of the proposed crowd counting system is shown in Fig. 5. The video is first segmented into crowd regions moving in different directions. Features are then extracted from each crowd segment, after the application of a perspective map that weighs pixels according to their approximate size in the 3-D world. Finally, the number of people per segment is estimated from the feature vector, using the BPR module of the previous section. The remainder of this section describes each of these components.

A. Crowd Segmentation

The first step of the system is to segment the scene into the crowd subcomponents of interest. The goal is to count people moving in different directions or with different speeds. This is accomplished by first using a mixture of dynamic textures [41] to segment the crowd into subcomponents of distinct motion flow. The video is represented as collection of spatiotemporal patches, which are modeled as independent samples from a mixture of dynamic textures. The mixture model is learned with the expectation–maximization algorithm, as described in [41]. Video locations are then scanned sequentially; a patch is extracted at each location and assigned to the mixture component of the 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 (a representative clip). The remaining video can then be segmented by simple computation of the posterior assignments. Full implementation details are available in [41].

B. Perspective Normalization

The extraction of features from crowd segments should take into account the effects of perspective. Because objects closer to the camera appear larger, any pixels associated with a close foreground object account for a smaller portion of it than those of an object farther away. This can be compensated by normalizing for perspective during feature extraction (e.g., when computing the segment area). In this paper, each pixel is weighted according to a perspective normalization map, which is based on the expected depth of the object that generated the pixel. Pixel weights encode the relative size of an object at different depths, with larger weights given to far objects.

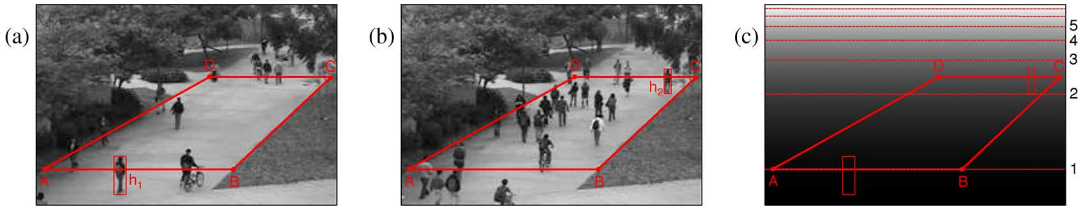

Fig. 6. Perspective map: (a) reference person at the front of walkway, and (b) at the end, and (c) the perspective map, which scales pixels by their relative size in the true 3d scene.

The perspective map is estimated by linearly interpolating the size of a reference person (or object) between two extremes of the scene. First, a rectangle is marked in the ground plane, by specifying points $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$, as in Fig. 6(a). It is assumed that 1) $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ form a rectangle in 3-D, and 2) $\overline{\mathbf{AB}}$ and $\overline{\mathbf{CD}}$ are horizontal lines in the image plane. A reference person is then selected in the video, and heights h_1 and h_2 estimated as the center of the person move over $\overline{\mathbf{AB}}$ and $\overline{\mathbf{CD}}$, as in Fig. 6(a) and (b). In particular, the pixels on the near and far sides of the rectangle are assigned weights based on the area of the object at these extremes: pixels on $\overline{\mathbf{AB}}$ receive weight 1 and those on $\overline{\mathbf{CD}}$ weight equal to the area ratio $(h_1 w_1)/(h_2 w_2)$, where w_1 is the length of $\overline{\mathbf{AB}}$ and w_2 is the length of $\overline{\mathbf{CD}}$. The remaining pixel weights are obtained by linearly interpolating the width of the rectangle and the height of the reference person at each image coordinate and computing the area ratio. Fig. 6(c) shows the resulting perspective map for the scene in Fig. 6(a). In this case, objects in the foreground $(\overline{\mathbf{AB}})$ are approximately 2.4 times bigger than objects in the background $(\overline{\mathbf{CD}})$. In other words, pixels on $\overline{\mathbf{CD}}$ are weighted 2.4 times as much as pixels on $\overline{\mathbf{AB}}$. We note that many other methods could be used to estimate the perspective map, e.g., a combination of a standard camera calibration technique and a virtual person who is moved around in the scene [42] or even the inclusion of the spatial weighting in the regression itself. We found this simple interpolation procedure sufficient for our experiments.

C. *Feature Extraction*

In principle, features such as segment area should vary linearly with the number of people in the scene [10], [31]. Fig. 2(a) shows a plot of this feature versus the crowd size. While the overall trend is indeed linear, local nonlinearities arise from a variety of factors, including occlusion, segmentation errors, and pedestrian configuration (e.g., variable spacing of people within a segment). To model these nonlinearities, an additional 29 features, which are based on segment shape, edge information, and texture, are extracted from the video. When computing features based on area or size, each pixel is weighted by the corresponding value in the perspective map. When the features are based on edges (e.g., edge histogram), each edge pixel is weighted by the square root of the perspective map value.

1) Segment Features: Features are extracted to capture segment properties such as shape and size. Features are also extracted from the segment perimeter, i.e., computed by morphological erosion with a disk of radius 1.

- *Area*—number of pixels in the segment.

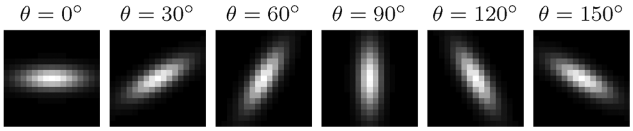

Fig. 7. Filters used to compute edge orientation.

- *Perimeter*—number of pixels on the segment perimeter.
- *Perimeter edge orientation*—a 6-bin histogram of the orientation of the segment perimeter. The orientation of each edge pixel is estimated by the orientation of the filter of maximum response within a set of 17 × 17 oriented Gaussian filters (see Fig. 7 for examples).
- *Perimeter-area ratio*—ratio between the segment perimeter and area. This feature measures the complexity of the segment shape: segments of high ratio contain irregular perimeters, which may be indicative of the number of people contained within.
- *"Blob" count*—number of connected components, with more than 10 pixels, in the segment.

2) Internal Edge Features: The edges within a crowd segment are a strong clue about the number of people in it [13], [14]. A Canny edge detector [43] is applied to the image, the output is masked to form the internal edge image (see Fig. 8), and a number of features are extracted.

- *Edge length*—number of edge pixels in the segment.
- *Edge orientation*—6-bin histogram of edge orientations.
- *Minkowski dimension*—fractal dimension of the internal edges, which estimates the degree of "space-filling" of the edges [44].

3) Texture Features: Texture features, which are based on the gray-level cooccurrence matrix, were used in [15] to classify image patches into five classes of crowd density (very low, low, moderate, high, and very high). In this paper, we adopt a similar set of measurements for estimating the number of pedestrians in each segment. The image is first quantized into eight gray levels and masked by the segment. The joint probability of neighboring pixel values $p(i,j|\theta)$ is then estimated for four orientation, $\theta \in \{0°, 45°, 90°, 135°\}$. A set of three features is extracted for each θ for a total of 12 texture features.

- *Homogeneity*: the texture smoothness, $g_\theta = \sum_{i,j} p(i,j|\theta)/(1+|i-j|)$.
- *Energy*: the total sum-squared energy, $e_\theta = \sum_{i,j} p(i,j|\theta)^2$.
- *Entropy*: the randomness of the texture distribution, $h_\theta = \sum_{i,j} p(i,j|\theta) \log p(i,j|\theta)$.

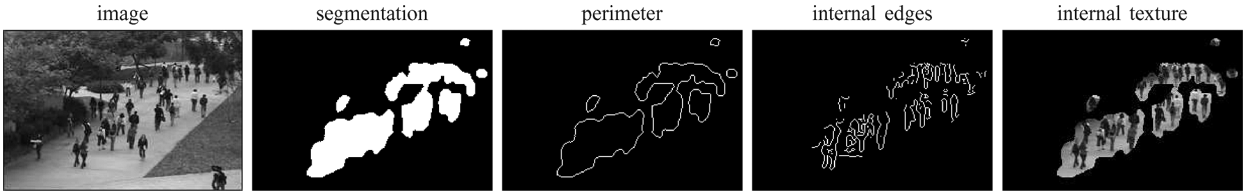

Fig. 8. Examples of an image frame, segment mask, segment perimeter, internal edges, and internal texture.

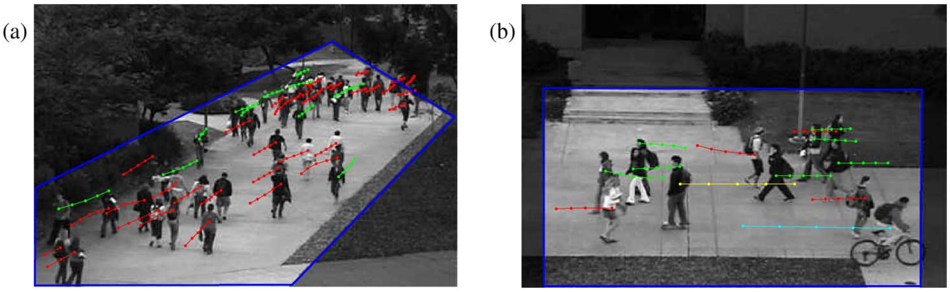

Fig. 9. Ground-truth annotations. (a) Peds1 database: red and green tracks indicate people moving away from, and toward the camera. (b) Peds2 database: red and green tracks indicate people walking right or left, whereas cyan and yellow tracks indicate fast objects moving right or left. The ROI used in all experiments is highlighted and outlined in blue.

Finally, a feature vector is formed by concatenating the 30 features, into the vector $\mathbf{x} \in \mathbb{R}^{30}$, which is used as the input for the regression module of the previous section.

VI. Experimental Evaluation

The proposed approach to crowd counting was tested on two pedestrian databases.

A. Pedestrian Databases

Two hours of video were collected from two viewpoints overlooking a pedestrian walkway at the University of California San Diego, using a stationary digital camcorder. The first viewpoint, shown in Fig. 9(a), is an oblique view of a walkway, containing a large number of people. The second, shown in Fig. 9(b), is a side view of a walkway, containing fewer people. We refer to these two viewpoints as Peds1 and Peds2, respectively. 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 4000 frames (400 seconds) of each video sequence were used for ground-truth annotation.

An ROI was selected on the main walkway (see Fig. 9), and the traveling direction (motion class) and the visible center of each pedestrian[1] were manually annotated every five frames. Pedestrian locations in the remaining frames were estimated by linear interpolation. Note that the pedestrian locations are only used to test detection performance of the pedestrian detectors in Section VI-E. For regression-based counting, only the counts in each frame are required for training. Peds1 was annotated with two motion classes: "away" from or "towards" the camera. For Peds2, the motion was split by direction and speed, resulting in four motion classes: "right-slow," "left-slow," "right-fast," and "left-fast." In addition, each data set also has a "scene" motion class, which is the total number of moving people in the frame (i.e., the sum of the individual motion classes). Example annotations are shown in Fig. 9.

Each database was split into a training set, which was used to learn the regression model, and a test set, which was used for validation. On Peds1, the training set contains 1200 frames (frames 1401-2600), with the remaining 2800 frames held out for testing. On Peds2, the training set contains 1000 frames (frames 1501-2500) with the remaining 3000 frames held out for testing. Note that these splits test the ability of crowd-counting algorithms to extrapolate beyond the training set. In contrast, spacing the training set evenly throughout the data set would only test the ability to interpolate between the training data, which provides little insight into generalization ability.

B. Experimental Setup

Since Peds1 contains two dominant crowd motions ("away" and "towards"), a mixture of dynamic textures [41] with $K = 2$ components was learned from $7 \times 7 \times 20$ spatiotemporal patches, which were extracted from a short video clip. The model was then used to segment the full video into two segments. The segment for the overall "scene" motion class is obtained by taking the union of the segments of the two motion classes. Peds2 contains four dominant crowd motions ("right-slow," "left-slow," "right-fast," or "left-fast"); hence, a $K = 4$ component mixture was learned from $13 \times 13 \times 10$ patches (larger patches are required since the people are larger in this video).

We treat each motion class (e.g., "away") as a separate regression problem. The 30-D feature vector of Section V-C was computed from each crowd segment and each video frame, and each feature was normalized to zero mean and unit variance. The GPR and BPR functions were then learned, using maximum marginal likelihood to obtain the optimal kernel hyperparameters. We used the GPML implementation [19] to find the maximum, which uses gradient ascent. For BPR, we modify GPML to include the special BPR noise term. GPR and BPR were learned with two kernels: the linear kernel (denoted GPR-l and

[1] Bicyclists and skateboarders in Peds1 were treated as regular pedestrians.

TABLE I
COMPARISON OF REGRESSION APPROACHES AND FEATURE SETS ON PEDS1

		MSE				err			
Feat.	Method	away	towards	scene	total	away	towards	scene	total
$\mathcal{F}_{all}$	linear	3.335	2.868	3.751	9.953	1.451	1.324	1.513	4.288
$\mathcal{F}_{all}$	GPR-l	3.260	2.692	3.654	9.606	1.435	1.278	1.489	4.203
$\mathcal{F}_{all}$	GPR-rr	2.970	2.029	3.787	8.785	1.408	**1.093**	1.551	4.051
$\mathcal{F}_{all}$	Poisson	2.917	3.065	3.040	9.022	1.336	1.360	1.331	4.027
$\mathcal{F}_{all}$	BPR-l	2.936	2.120	**2.910**	7.966	1.336	1.160	**1.308**	3.804
$\mathcal{F}_{all}$	BPR-rr	**2.441**	**1.996**	2.975	**7.412**	**1.210**	1.124	1.320	**3.654**
$\mathcal{F}_{se}$	BPR-rr	2.751	3.019	6.702	8.867	1.307	1.378	1.365	4.050
$\mathcal{F}_t$	BPR-rr	23.300	12.142	60.178	95.619	3.478	2.846	5.824	12.149
$\mathcal{F}_e$	BPR-rr	3.460	4.071	3.406	10.938	1.478	1.590	1.431	4.499
$\mathcal{F}_s$	BPR-rr	3.396	2.895	4.734	11.025	1.384	1.347	1.761	4.491
$\mathcal{F}_a$	BPR-rr	3.923	3.224	6.117	13.264	1.461	1.470	1.951	4.883
[13]	BPR-rr	3.264	3.105	3.640	10.010	1.416	1.418	1.478	4.312
[14]	BPR-rr	3.118	2.808	3.661	9.587	1.385	1.339	1.500	4.224

BPR-l) and the RBF–RBF compound kernel (denoted as GPR-rr and BPR-rr). For GPR-l and BPR-l, the initial hyperparameters were set to $\theta = [1 \ \cdots \ 1]$, whereas for GPR-rr and BPR-rr, the optimization was performed over five trials with random initializations to avoid bad local maxima. For completeness, standard linear least squares and Poisson regression were also tested.

For GPR, counts were estimated by the mean prediction value μ_*, which is rounded to the nearest nonnegative integer. The standard deviation σ_* was used as an uncertainty measure. For BPR, counts were estimated by the mode of the predictive distribution, and $\mathrm{unc}(y_*)$ was used as uncertainty measure. The accuracy of the estimates was evaluated by the mean square error MSE $= (1/M)\sum_{i=1}^{M}(\hat{c}_i - c_i)^2$ and by the absolute error $\mathrm{err} = (1/M)\sum_{i=1}^{M}|\hat{c}_i - c_i|$, where c_i and $\hat{c}_i$ are the true and estimated counts for frame i and M is the number of test frames. Experiments were conducted with different subsets of the 30 features: only the segment area (denoted as $\mathcal{F}_a$), segment-based features ($\mathcal{F}_s$), edge-based features ($\mathcal{F}_e$), texture features ($\mathcal{F}_t$), and segment and edge features ($\mathcal{F}_{\mathrm{se}}$). The full set of 30 features is denoted as $\mathcal{F}_{\mathrm{all}}$. The feature sets of [14] (segment size histogram and edge orientation histogram) and [13] (segment area and total edge length) were also tested.

C. Results on Peds1

Table I presents counting error rates for Peds1 for each of the motion classes ("away," "towards," and "scene"). In addition, we also report the total MSE and total absolute error as an indicator of overall performance of each method. A number of conclusions are possible. First, Bayesian regression has better performance than the non-Bayesian approaches. For example, BPR-l achieves an overall error rate of 3.804 versus 4.027 for standard Poisson regression. The error is further decreased to 3.654 by adopting the compound kernel BPR-rr. Second, the comparison of the two Bayesian regression models shows that BPR outperforms GPR. With linear kernels, BPR-l outperforms GPR-l on all classes (total error 3.804 versus 4.203). In the nonlinear case, BPR-rr has significantly lower error than GPR-rr on the "away" and "scene" classes (e.g., 1.210 versus 1.408 on the "away" class), and comparable performance (1.124 versus 1.093) on the "towards" class. In general, BPR has the largest gains in the sequences where GPR has larger error. Third, the use of sophisticated regression models does make a difference. The error rate of the best method (BPR-rr, 3.654) is 85% of that of the worst method (linear least squares, 4.288).

Fourth, performance is also strongly affected by the features used. This is particularly noticeable on the "away" class, which has larger crowds. On this class, the error steadily decreases as more features are included in the model. Using just the area feature $(\mathcal{F}_a)$ yields a counting error of 1.461. When the segment features $(\mathcal{F}_s)$ are used, the error decreases to 1.384, and adding the edge features $(\mathcal{F}_{\mathrm{se}})$ leads to a further decrease to 1.307. Finally, adding the texture features $(\mathcal{F}_{\mathrm{all}})$ achieves the lowest error of 1.21. This illustrates the different components of information contributed by the different feature subsets: the estimate produced from segment features is robust but coarse, and the refinement by edge and texture features allows the modeling of various nonlinearities. Note also that the isolated use of texture features results in very poor performance (overall error of 12.149). However, these features provide important supplementary information when used in conjunction with others, as in $\mathcal{F}_{\mathrm{all}}$. Compared with [13] and [14], the full feature set $\mathcal{F}_{\mathrm{all}}$ performs better on all crowd classes (total errors 3.654 versus 4.312 and 4.224).

The effect of varying the training set size was also examined by using subsets of the original training set. For a given training set size, results were averaged over different subsets of evenly spaced frames. Fig. 10 shows plots of the MSE versus training set size. Table II summarizes the results obtained with 100 training images. The experiment was repeated for 12 different splits of the training and test sets, with the mean and standard devitations reported. Note how the Bayesian methods (BPR and GPR) have much better performance than linear or Poisson regression when the training set is small. In practice, this means that the Bayesian crowd counting requires much fewer training examples and a reduced number of manually annotated images.

We observe that Poisson and BPR perform similarly on the "scene" class for large training sizes. Combining the two motion segments to form the "scene" segment removes segmentation errors and small segments containing partially occluded people traveling against the main flow. Hence, the features extracted from the "scene" segment have fewer outliers, resulting

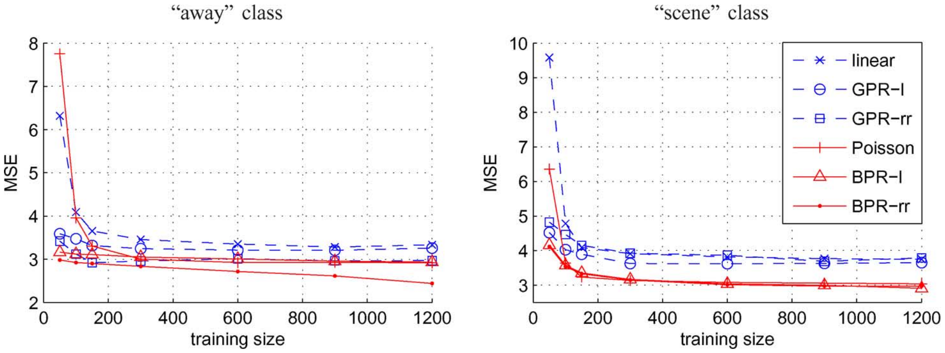

Fig. 10. Error rate for training sets of different sizes on Peds1, for the "away" (left) and "scene" (right) classes. Similar plots were obtained for the "towards" class and are omitted for brevity.

TABLE II
RESULTS ON PEDS1 USING 100 TRAINING IMAGES. STANDARD DEVIATIONS ARE GIVEN IN PARENTHESIS

	MSE			err		
Method	away	towards	scene	away	towards	scene
linear	4.090(0.609)	3.659(0.500)	4.780(0.818)	1.590(0.124)	1.495(0.109)	1.713(0.158)
GPR-l	3.472(0.288)	**1.923**(0.128)	4.029(0.298)	1.467(0.079)	**1.053**(0.041)	1.553(0.060)
GPR-rr	3.118(0.154)	2.272(0.604)	4.465(0.495)	1.408(0.055)	1.160(0.119)	1.656(0.106)
Poisson	3.956(0.598)	3.605(0.395)	3.643(0.370)	1.488(0.086)	1.469(0.077)	1.469(0.080)
BPR-l	3.118(0.094)	2.358(0.093)	3.569(0.141)	1.362(0.019)	1.243(0.028)	1.450(0.037)
BPR-rr	**2.924**(0.093)	2.320(0.089)	**3.537**(0.127)	**1.324**(0.017)	1.233(0.028)	**1.442**(0.031)

TABLE III
COMPARISON OF REGRESSION APPROACHES ON PEDS1 USING DIFFERENT SEGMENTATION METHODS AND $\mathcal{F}_{\text{all}}$ ("SCENE" CLASS)

	scene MSE			scene err		
Method	DTM	median	GMM	DTM	median	GMM
linear	3.751	4.009	5.563	1.513	1.551	1.898
GPR-l	3.654	3.934	5.623	1.489	1.540	1.900
GPR-rr	3.787	3.676	4.576	1.551	1.476	1.691
Poisson	3.040	3.585	4.178	1.331	1.449	1.585
BPR-l	**2.910**	3.453	3.597	**1.308**	1.428	1.445
BPR-rr	2.975	3.378	3.391	1.320	1.415	1.383

in a simpler regression problem. This justifies the similar performance of Poisson and BPR. On the other hand, Bayesian regression improves performance for the other two motion classes, where segmentation errors or occlusion effects originate a larger number of outlier features.

As an alternative to motion segmentation, two background subtraction methods, i.e., a temporal median filter and an adaptive GMM [45], were used to obtain the "scene" segment, which was then used for count regression. The counting results were improved by applying two postprocessing steps to the foreground segment: 1) a spatial median filter to remove spurious noise and 2) morphological dilation (a disk with a radius of 2) to fill in holes and include pedestrian edges. The results are summarized in Table III. Counting using DTM motion segmentation outperforms both background subtraction methods (1.308 error versus 1.415 and 1.383). Because the DTM segmentation is based on motion differences, rather than gray-level differences, it tends to have fewer segmentation errors (i.e., completely missing part of a person) when a person has similar gray level to the background.

Finally, Fig. 11 displays the crowd count estimates obtained with BPR-rr. These estimates track the ground-truth well in most of the test set. Furthermore, the uncertainty measure (shown in green) indicates when BPR has lower confidence in the prediction. This is usually when the size of the crowd increases. Fig. 12 shows crowd estimates for several test frames of Peds1. A video is also available from [46]. In summary, the count estimates produced by the proposed algorithm are accurate for a wide range of crowd sizes. This is due to both the inclusion of texture features, which are informative for high-density crowds, and the Bayesian nonlinear regression model, which is quite robust.

D. Crowd Counting Results on Peds2

The Peds2 data set contains smaller crowds (at most 15 people). We found that the segment and edge features $\mathcal{F}_{\text{se}}$ worked the best on this data set. Table IV shows the error rates for the five crowd segments, using the different regression models. The best overall performance is achieved by GPR-l, with an overall error of 1.586. The exclusion of the texture features and the smaller crowd originates a strong linear trend in the data, which is better modeled with GPR-l than the nonlinear GPR-rr. Both BPR-l and BPR-rr perform worse than GPR-l overall (1.927 and 1.776 versus 1.586). This is due two reasons. First, at lower counts, the $\mathcal{F}_{\text{se}}$ features tend to grow linearly with the count. This does not fit well the exponential model that underlies BPR-l. Due to the nonlinear kernel, BPR-rr can adapt to this but appears to suffer from some overfitting. Second, the observation noise of BPR is inversely proportional to the count. Hence, uncertainty is high for low counts, limiting how well BPR can learn local variations in the data. These problems are due to reduced accuracy of the log-gamma approximation of (15) when k is small. Finally, the estimates obtained with $\mathcal{F}_{\text{se}}$ are more accurate than those of [13] and [14] on all motion classes and particularly more accurate in the two fast classes. This indicates that the feature space now proposed is richer and more informative.

TABLE IV
COMPARISON OF REGRESSION METHODS AND FEATURE SETS ON PEDS2

		MSE						err					
Feat.	Method	right-slow	left-slow	right-fast	left-fast	scene	total	right-slow	left-slow	right-fast	left-fast	scene	total
$\mathcal{F}_{se}$	GPR-l	**0.686**	0.476	**0.009**	**0.004**	**0.990**	**2.165**	**0.485**	0.417	**0.009**	**0.004**	**0.671**	**1.586**
$\mathcal{F}_{se}$	GPR-rr	0.877	0.508	0.024	0.009	1.142	2.560	0.576	0.442	0.024	0.009	0.740	1.790
$\mathcal{F}_{se}$	BPR-l	1.055	0.598	0.017	0.009	1.253	2.932	0.698	0.451	0.017	0.009	0.753	1.927
$\mathcal{F}_{se}$	BPR-rr	0.933	**0.458**	0.016	0.008	1.132	2.547	0.615	**0.394**	0.016	0.008	0.743	1.776
[13]	GPR-l	0.736	0.614	0.017	0.032	1.144	2.543	0.528	0.510	0.017	0.018	0.729	1.802
[14]	GPR-l	0.706	0.491	0.020	0.011	1.048	2.277	0.499	0.424	0.020	0.009	0.714	1.666

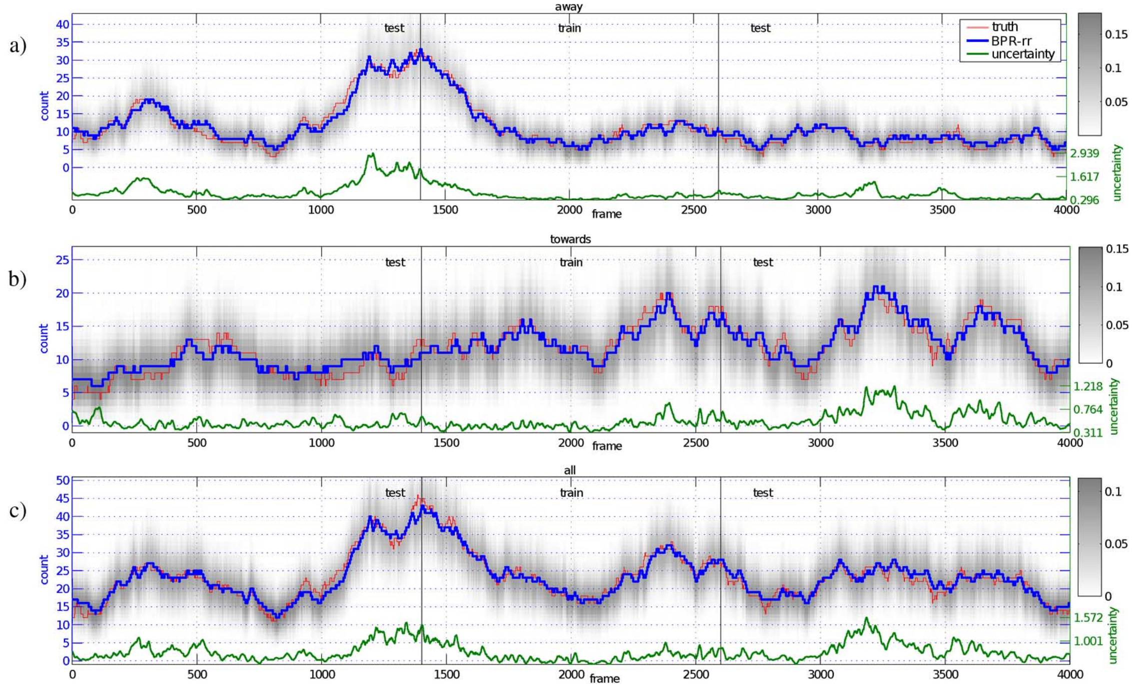

Fig. 11. Crowd counting results on Peds1: (a) "away," (b) "towards," and (c) "scene" classes. Gray levels indicate probabilities of the predictive distribution. The uncertainty measure of the prediction is plotted in green, with the axes on the right.

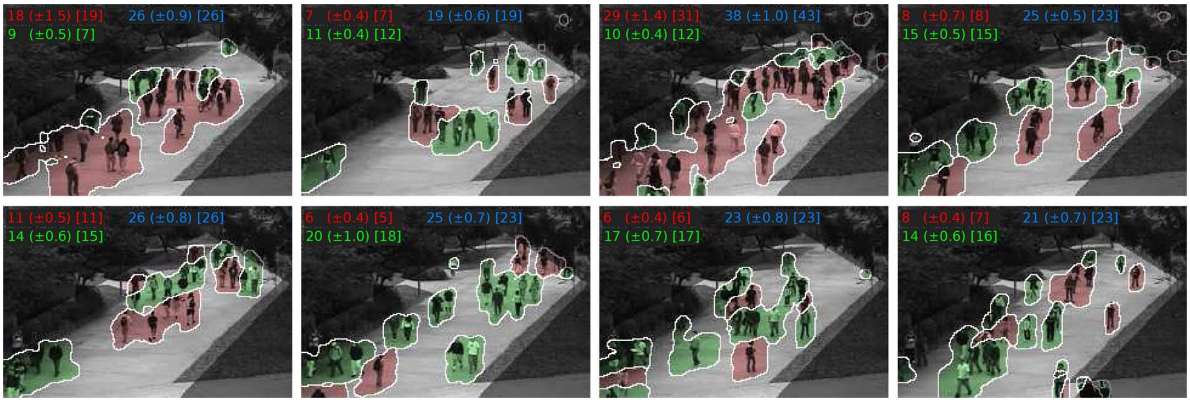

Fig. 12. Crowd counting examples: The red and green segments are the "away" and "towards" components of the crowd. The estimated crowd count for each segment is shown in the top left, with the (uncertainty) and the (ground truth). The prediction for the "scene" class, which is count of the whole scene, is shown in the top right. The ROI is also highlighted.

Fig. 13 shows the crowd count estimates (using $\mathcal{F}_{\text{se}}$ and GPR-l) for the five motion classes over time, and Fig. 14 presents the crowd estimates for several frames in the test set. Video results are also available from [46]. The estimates track the ground-truth well in most frames, for both the fast and slow motion classes. One error occurs for the "right-fast" class, where one skateboarder is missed due to an error in the segmentation, as displayed in the last image of Fig. 14. In summary, the results on Peds2, again, suggest the efficacy of regression-based crowd counting from low-level features.

E. Comparison With Pedestrian Detection Algorithms

In this section, we compare regression-based crowd counting with counting using two state-of-the-art pedestrian detectors.

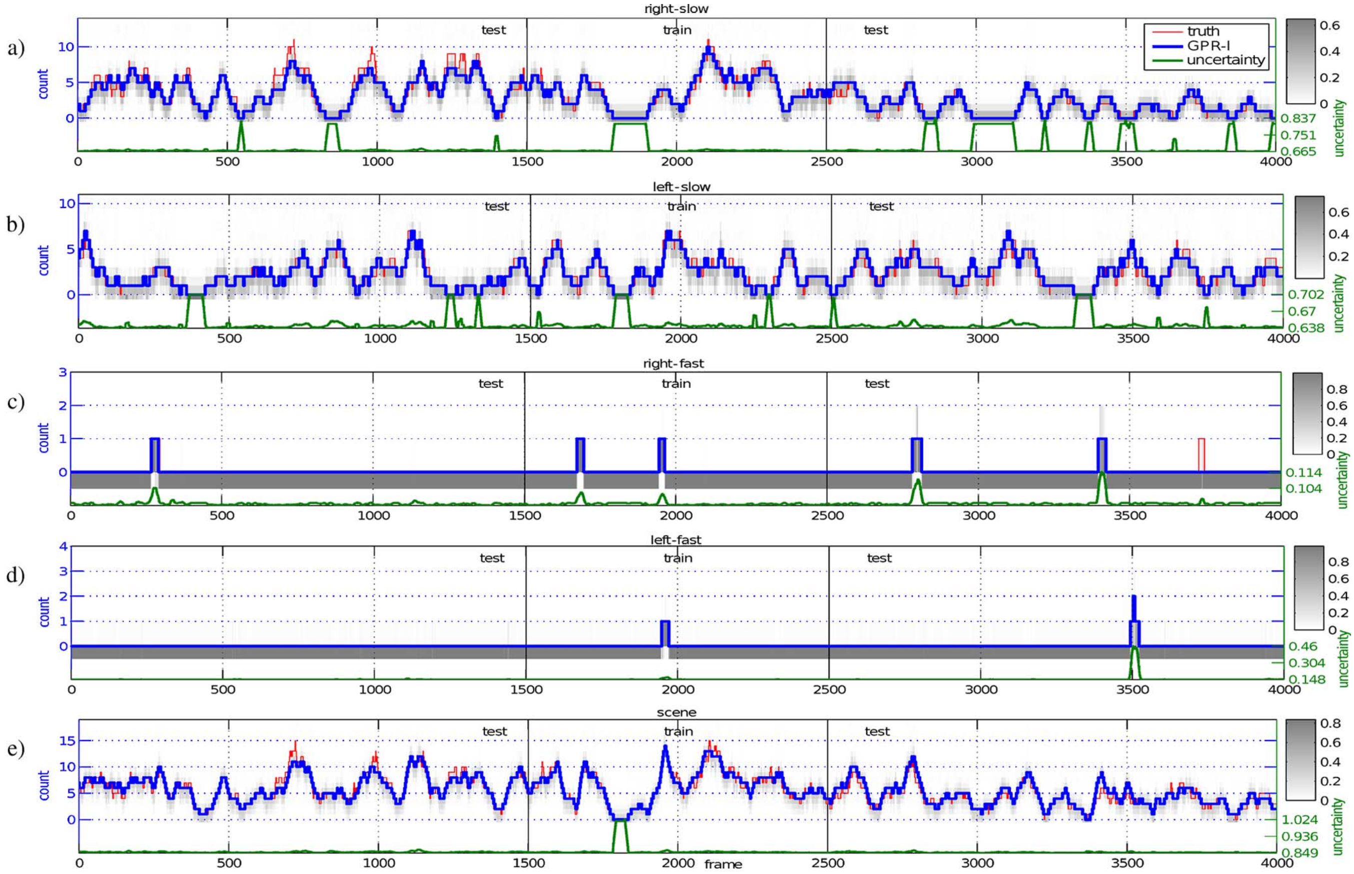

Fig. 13. Crowd counting results on Peds2 for: (a) "right-slow," (b) "left-slow," (c) "right-fast," (d) "left-fast," (e) "scene." Gray levels indicate probabilities of the predictive distribution.

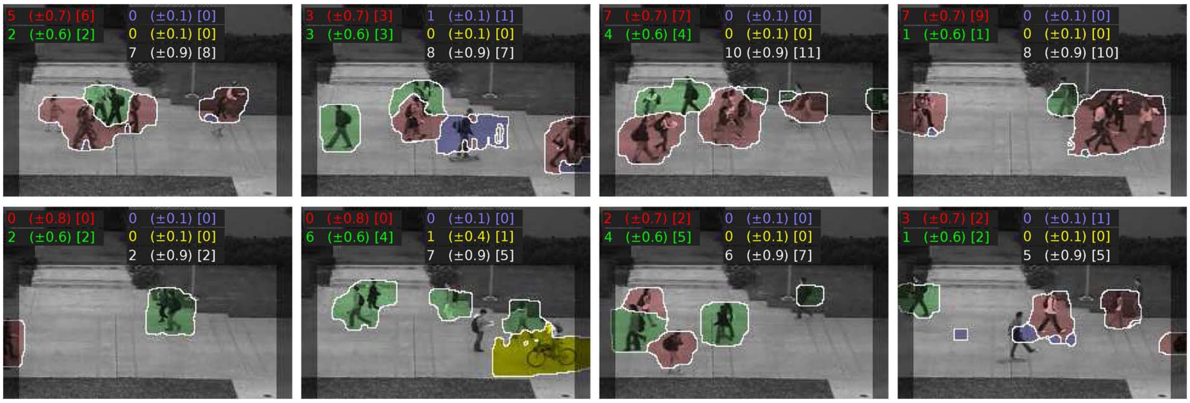

Fig. 14. Counting examples from Peds2: The red and green segments are the "right-slow" and "left-slow" components of the crowd, and the blue and yellow segments are the "right-fast" and "left-fast" ones. The estimated crowd count for each segment is shown in the top left, with the (uncertainty) and the (ground truth). The count for the "scene" class, which is the count of the whole scene, is shown in white text.

The first detects pedestrians with an SVM and the histogram-of-gradient (HOG) feature [25]. The second is based on a discriminatively trained deformable part model (DPM) [26]. The detectors were provided by the respective authors. They were both run on the full-resolution video frames (740 × 480), and a filter was applied to remove detections that are outside the ROI, inconsistent with the perspective of the scene or given low confidence. Nonmaximum suppression was also applied to remove multiple detections of the same object.

We start by evaluating the performance of the two detectors. Each ground-truth pedestrian was uniquely mapped to the closest detection, and a true positive (TP) was recorded if the ground-truth location was within the detection bounding box. A false positive (FP) was recorded otherwise. Fig. 15 plots the ROC curves for HOG and DPM on Peds1 and Peds2. These curves are obtained by varying the threshold of the confidence filter. HOG outperforms DPM on both data sets, with a smaller FP rate per image. However, neither algorithm is able to achieve

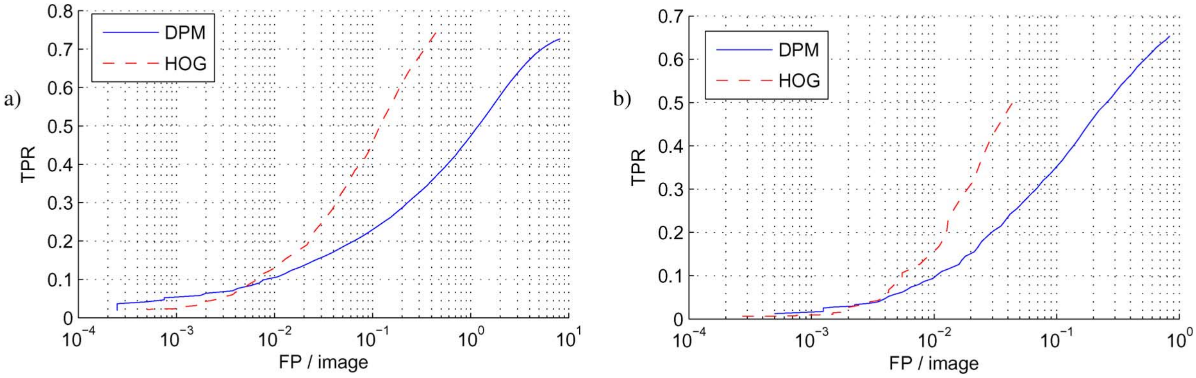

Fig. 15. ROC curves of the pedestrian detectors on (a) Peds1 and (b) Peds2.

TABLE V
COUNTING ACCURACY OF BAYESIAN REGRESSION (BPR AND GPR) AND PEDESTRIAN DETECTION (HOG AND DPM)

Peds1				
Method	MSE	err	bias	var.
$\mathcal{F}_{all}$ BPR-rr	**2.975**	**1.320**	**0.101**	**2.966**
DPM [26]	24.721	4.012	1.621	22.100
HOG [25]	39.755	5.321	−5.315	11.510
DPM BPR-l	51.489	6.298	5.256	23.875
HOG BPR-l	33.222	4.893	3.498	20.995

Peds2				
Method	MSE	err	bias	var.
$\mathcal{F}_{se}$ GPR-l	**0.990**	**0.671**	**0.150**	**0.968**
DPM [26]	4.645	1.565	−0.983	3.680
HOG [25]	10.834	2.607	−2.595	4.103
DPM GPR-l	4.312	1.507	−0.741	3.765
HOG GPR-l	4.455	1.563	−0.595	4.103

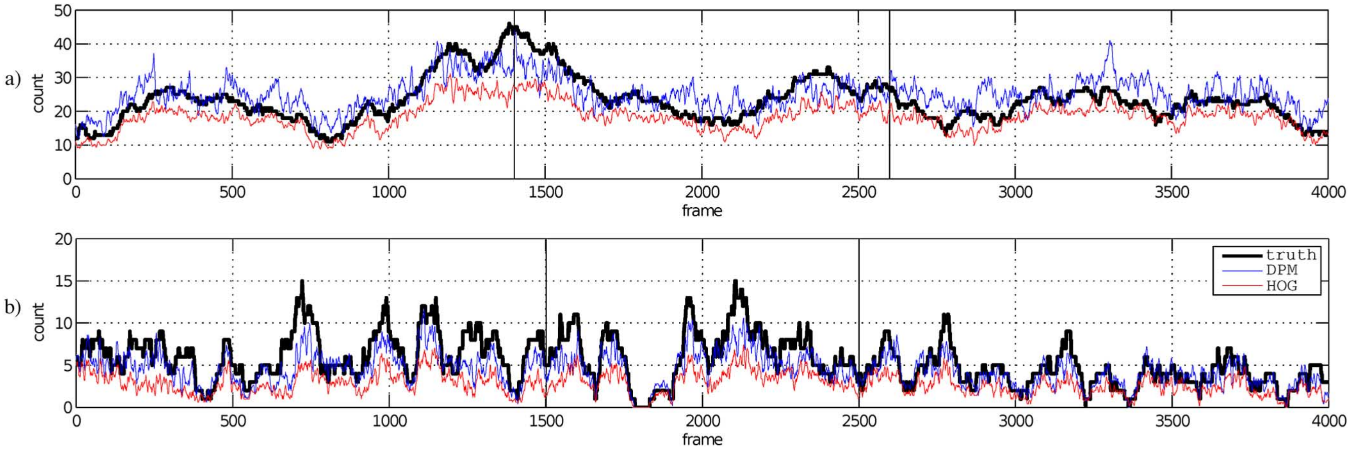

Fig. 16. Crowd counts produced by the HOG [25] and DPM [26] detectors on (a) Peds1 and (b) Peds2.

a very high TP rate (the maximum TP rate is 74% on Peds1), due to the large number of occlusions in these scenes.

Next, each detector was used to count the number of people in each frame, regardless of the direction of motion (corresponding to the "scene" class). The confidence threshold was chosen to minimize the counting error on the training set. In addition to the count error and MSE, we also report the bias and variance of the estimates, bias $= (1/M)\sum_{i=1}^{M}(c_i - \hat{c}_i)$ and var $= (1/M)\sum_{i=1}^{M}(c_i - \text{bias})^2$. The counting performance of DPM and HOG is summarized in Table V, and the crowd counts are displayed in Fig. 16. For crowd counting, DPM has a lower average error rate than HOG (e.g., 4.012 versus 5.321 on Peds1). This is an artifact of the high FP rate of DPM; the false detections artificially boost the count, although the algorithm has a lower TP rate. On the other hand, HOG always underestimates the crowd count, as is evident in Fig. 16 and the biases of −5.315 and −2.595. Both detectors perform significantly worse than regression-based crowd counting (BPR or GPR). In particular, the average error of the former is more than double that of the latter (e.g., 4.012 for DPM versus 1.320 for BPR on Peds1). Fig. 17 shows the error as a function of ground-truth crowd size. For the pedestrian detectors, the error increases significantly with the crowd size, due to occlusion. On the other hand, the performance of Bayesian regression remains relatively constant. These results demonstrate that regression-based counting can perform well above state-of-the-art pedestrian detectors, particularly when the crowd is dense.

Finally, we applied Bayesian regression (BPR or GPR) on the detector counts (HOG or DPM), in order to remove any systematic bias in the count prediction. Using the training set, a Bayesian regression function was learned to map the detector count to the ground-truth count. The counting accuracy on the

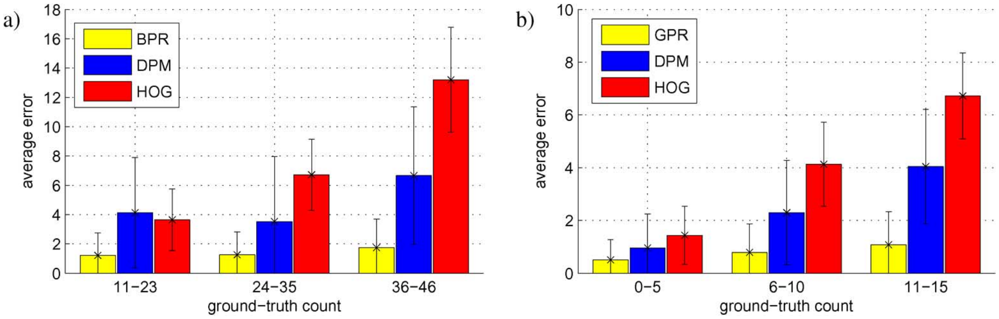

Fig. 17. Error for different crowd sizes on (a) Peds1 and (b) Peds2.

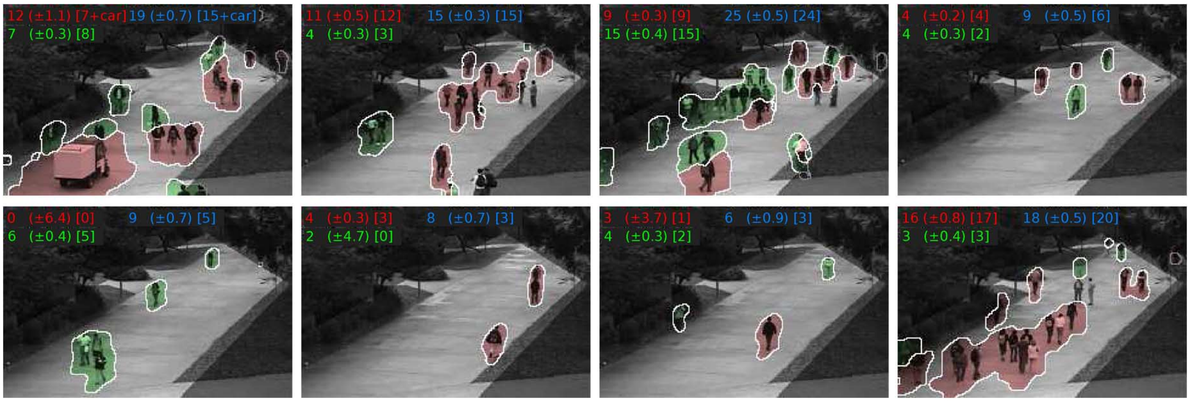

Fig. 18. Example counting results on the full Peds1 data set.

test set was then computed using the regression function. The (best) results are presented in the bottom half of Table V. There is not a significant improvement compared with the raw counts, suggesting that there is no systematic warping between the detector counts and the actual counts.

F. Extended Results on Peds1 and Peds2

The final experiment tested the robustness of regression-based counting on 2 hours of video from Peds1 and Peds2. For both data sets, the top-performing model and feature set (BPR-rr with $\mathcal{F}_{\mathrm{all}}$ for Peds1 and GPR-l with $\mathcal{F}_{\mathrm{se}}$ for Peds2) were trained using 2000 frames of the annotated data set (every other frame). Counts were then estimated on the remaining 50 min of each video. Examples of the predictions on Peds1 are shown in Fig. 18, and full video results are available from [46]. Qualitatively, the counting algorithm 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 in Peds1. This problem could be easily fixed by adding more training examples. Note that BPR signals lack confidence in these estimates, by assigning them large standard deviations (e.g., fifth and sixth images of Fig. 18).

A more challenging set of errors occurs when bicycles, skateboarders, and golf carts travel quickly on the Peds1 walkway (e.g., first image of Fig. 18). Again, these errors are reasonable since there are very few examples of fast moving bicycles and no examples of carts in the training set. These cases could be handled by either: 1) adding more mixture components to the segmentation algorithm to label fast moving objects as a different class, or 2) detecting outlier objects that have different appearance or motion from the dominant crowd. In both cases, the segmentation task is not as straightforward due to the scene perspective; people moving in the foreground areas travel at the same speed as bikes moving in the background areas. Future work will be directed at developing segmentation algorithms to handle these cases.

Examples of prediction on Peds2 are displayed in Fig. 19. Similar to Peds1, the algorithm tracks the changes in pedestrian traffic fairly well. Most errors tend to occur on objects that are not seen in the database, e.g., three people pulling carts (sixth image in Fig. 19), or the small truck (final image of Fig. 19). Again, these errors are reasonable, considering that these objects were not seen in the training set, and the problem could be fixed by simply adding training examples of such cases or by detecting them as outliers.

VII. Conclusion

In this paper, we have proposed the use of Bayesian regression to estimate the size of inhomogeneous crowds, which are composed of pedestrians traveling in different directions, without using intermediate vision operations, such as object detection or feature tracking. Two solutions were presented, based on the GPR and BPR. The intractability of the latter was addressed through the derivation of closed-form approximations to the predictive distribution. It was shown that the BPR model can be kernelized, to represent nonlinear log-arrival rates, and that the hyperparameters of the kernel can be estimated by approximate maximum marginal likelihood. Regression-based

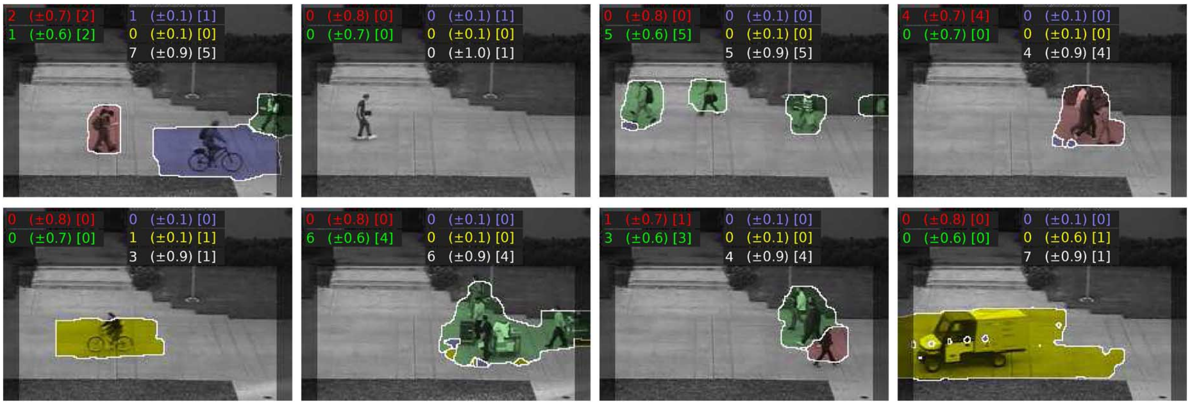

Fig. 19. Example counting results on the full Peds2 video.

counting was validated on two large data sets and was shown to provide robust count estimates regardless of the crowd size.

Comparing the two Bayesian regression methods, BPR was found more accurate for denser crowds, whereas GPR performed better when the crowd was less dense (in which case, the regression mapping is more linear). Both Bayesian regression models were shown to generalize well from small training sets, requiring significantly smaller amounts of hand-annotated data than non-Bayesian crowd counting approaches. The regression-based count estimates were also shown substantially more accurate than those produced by state-of-the-art pedestrian detectors. Finally, regression-based counting was successfully applied to 2 hours of video, suggesting that systems based on the proposed approach could be used in real-world environments for long periods of time.

One limitation, for crowd counting, of Bayesian regression is that it requires training for each particular viewpoint. This is an acceptable restriction for permanent surveillance systems. However, the training requirement may hinder the ability to quickly deploy a crowd counting system (e.g., during a parade). The lack of viewpoint invariance likely stems from several colluding factors: 1) changes in segment shape due to motion and perspective; 2) changes in a person's silhouette due to viewing angle; and 3) changes in the appearance of dense crowds. Future work will be directed at improving training across viewpoints, by developing perspective invariant features, by transferring knowledge across viewpoints (using probabilistic priors), or by accounting for a perspective within the kernel function itself. Further improvements to the performance of Bayesian counting from sparse crowds should also be possible. On BPR, a training example associated with a sparse crowd has less weight (more uncertainty) than one associated with a denser crowd. This derives from the Poisson noise model and diminishes the ability of BPR to model local variations of sparse crowds (in the presence of count uncertainty, Bayesian regression tends to smoothen the regression mapping). Future work will study noise models without this restriction.

Appendix

1) Property 1: Consider the following:

$$\begin{aligned}\Phi\Sigma_y^{-1}(\Phi^T\Sigma_p\Phi+\Sigma_y) &= \Phi\Sigma_y^{-1}\Phi^T\Sigma_p\Phi+\Phi &(40)\\ &= \left(\Phi\Sigma_y^{-1}\Phi^T+\Sigma_p^{-1}\right)\Sigma_p\Phi. &(41)\end{aligned}$$

Premultiplying by $(\Phi\Sigma_y^{-1}\Phi^T+\Sigma_p^{-1})^{-1}$ and postmultiplying by $(\Phi^T\Sigma_p\Phi+\Sigma_y)^{-1}$ yield

$$\left(\Phi\Sigma_y^{-1}\Phi^T+\Sigma_p^{-1}\right)^{-1}\Phi\Sigma_y^{-1} = \Sigma_p\Phi(\Phi^T\Sigma_p\Phi+\Sigma_y)^{-1}. \quad (42)$$

2) BPR Marginal Likelihood: We derive the BPR marginal likelihood of Section IV-C. In all equations, we only write the terms that depend on kernel $\{\Phi, \Sigma_p, \beta\}$. Using (19), the joint log-likelihood of $\{\mathbf{y}, \beta\}$ can be approximated as

$$\begin{aligned}&\log p(\mathbf{y},\beta|X,\theta)\\ &\quad= \log p(\mathbf{y}|X,\beta,\theta)+\log p(\beta|\theta) &(43)\\ &\quad\approx -\frac{N}{2}\log(2\pi)-\frac{1}{2}\log|\Sigma_y|-\frac{1}{2}\|\Phi^T\beta-\mathbf{s}\|^2_{\Sigma_y}-c\mathbf{1}^T\Phi^T\beta\\ &\qquad+\left(\sum_{i=1}^N\log\frac{\Gamma(y_i+c)}{y_i!}\right)\\ &\qquad-\frac{d}{2}\log(2\pi)-\frac{1}{2}\log|\Sigma_p|-\frac{1}{2}\beta^T\Sigma_p^{-1}\beta &(44)\\ &\quad\propto -\frac{1}{2}\left(\beta^TA\beta-2\beta^T\Phi\Sigma_y^{-1}\mathbf{s}+2\beta^T\Phi\mathbf{1}c\right)-\frac{1}{2}\log|\Sigma_p|\\ & &(45)\\ &\quad= -\frac{1}{2}\left(\beta^TA\beta-2\beta^T\Phi\Sigma_y^{-1}\mathbf{t}\right)-\frac{1}{2}\log|\Sigma_p| &(46)\end{aligned}$$

where $A = \Phi\Sigma_y^{-1}\Phi^T+\Sigma_p^{-1}$ and $\mathbf{t}$ and $\mathbf{s}$ are defined, as in Section IV-A. By completing the square

$$\begin{aligned}&\log p(\mathbf{y}|X,\beta,\theta)+\log p(\beta|\theta) &(47)\\ &\quad\approx -\frac{1}{2}\Big(\left\|\beta-A^{-1}\Phi\Sigma_y^{-1}\mathbf{t}\right\|^2_{A^{-1}}\\ &\qquad\qquad -\mathbf{t}^T\Sigma_y^{-1}\Phi^TA^{-1}\Phi\Sigma_y^{-1}\mathbf{t}\Big)-\frac{1}{2}\log|\Sigma_p| &(48)\\ &\quad\propto -\frac{1}{2}\Big(\left\|\beta-A^{-1}\Phi\Sigma_y^{-1}\mathbf{t}\right\|^2_{A^{-1}}\\ &\qquad\qquad +\mathbf{t}^T\Sigma_y^{-1}\mathbf{t}-\mathbf{t}^T\Sigma_y^{-1}\Phi^TA^{-1}\Phi\Sigma_y^{-1}\mathbf{t}\Big)\\ &\qquad -\frac{1}{2}\log|\Sigma_p| &(49)\\ &\quad= -\frac{1}{2}\Big(\left\|\beta-A^{-1}\Phi\Sigma_y^{-1}\mathbf{t}\right\|^2_{A^{-1}}\\ &\qquad\qquad +\mathbf{t}^T(\Sigma_y+\Phi^T\Sigma_p\Phi)^{-1}\mathbf{t}\Big)-\frac{1}{2}\log|\Sigma_p| &(50)\end{aligned}$$

where in (50), we use the matrix inversion lemma. The marginal likelihood can thus be approximated as

$$\begin{aligned} p(\mathbf{y}|X,\beta,\theta) &= \int p(\mathbf{y},\beta|X,\theta)d\beta && (51)\\ &\approx |\Sigma_p|^{-\frac{1}{2}} e^{-\frac{1}{2}\mathbf{t}^T(\Sigma_y+\Phi^T\Sigma_p\Phi)^{-1}\mathbf{t}} \\ &\quad\times \int e^{-\frac{1}{2}\left\|\beta-A^{-1}\Phi\Sigma_y^{-1}\mathbf{t}\right\|^2_{A^{-1}}} d\beta && (52)\\ &\propto |\Sigma_p|^{-\frac{1}{2}}|A^{-1}|^{\frac{1}{2}} e^{-\frac{1}{2}\mathbf{t}^T(\Sigma_y+\Phi^T\Sigma_p\Phi)^{-1}\mathbf{t}} && (53)\\ &= (|\Sigma_p||A|)^{-\frac{1}{2}}\, e^{-\frac{1}{2}\mathbf{t}^T(\Sigma_y+K)^{-1}\mathbf{t}}. && (54)\end{aligned}$$

Using the block determinant property, $|A|$ can be rewritten as

$$\begin{aligned} |A| &= \left|\Sigma_p^{-1}+\Phi\Sigma_y^{-1}\Phi^T\right| && (55)\\ &= \left|\Sigma_p^{-1}\right|\left|-\Sigma_y^{-1}\right|\left|-\Sigma_y-\Phi^T\Sigma_p\Phi\right| && (56)\\ &= \left|\Sigma_p^{-1}\right|\left|\Sigma_y^{-1}\right|\left|\Sigma_y+K\right|. && (57)\end{aligned}$$

Substituting into the log of (54) yields

$$\begin{aligned} \log p(\mathbf{y}|X,\beta,\theta) \approx \frac{1}{2}\log|\Sigma_y| &- \frac{1}{2}\log|\Phi^T\Sigma_p\Phi+\Sigma_y| \\ &- \frac{1}{2}\mathbf{t}^T(\Phi^T\Sigma_p\Phi+\Sigma_y)^{-1}\mathbf{t}. \quad (58)\end{aligned}$$

Finally, dropping the term that does not depend on the kernel hyperparameters θ yields (39).

Acknowledgment

The authors would like to thank J. Cuenco and Z.-S. J. Liang for annotating part of the ground-truth data, N. Dalal and P. Felzenszwalb for the detection algorithms from [25] and [26], P. Dollar for running these algorithms, and the anonymous reviewers for their helpful comments.

References

[1] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *Int. J. Comput. Vis.*, vol. 63, no. 2, pp. 153–161, 2005.

[2] T. Zhao and R. Nevatia, "Bayesian human segmentation in crowded situations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2003, vol. 2, pp. 459–466.

[3] T. Zhao, R. Nevatia, and B. Wu, "Segmentation and tracking of multiple humans in crowded environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1198–1211, Jul. 2008.

[4] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 875–885.

[5] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, vol. 1, pp. 90–97.

[6] S.-F. Lin, J.-Y. Chen, and H.-X. Chao, "Estimation of number of people in crowded scenes using perspective transformation," *IEEE Trans. Syst., Man, Cybern.*, vol. 31, no. 6, pp. 645–654, Nov. 2001.

[7] V. Rabaud and S. J. Belongie, "Counting crowded moving objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 705–711.

[8] G. J. Brostow and R. Cipolla, "Unsupervised Bayesian detection of independent motion in crowds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 1, pp. 594–601.

[9] B. Leibe, K. Schindler, and L. Van Gool, "Coupled detection and trajectory estimation for multi-object tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[10] N. Paragios and V. Ramesh, "A MRF-based approach for real-time subway monitoring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2001, vol. 1, pp. 1034–1040.

[11] 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. B, Cybern.*, vol. 29, no. 4, pp. 535–541, Aug. 1999.

[12] C. S. Regazzoni and A. Tesei, "Distributed data fusion for real-time crowding estimation," *Signal Process.*, vol. 53, no. 1, pp. 47–63, Aug. 1996.

[13] A. C. Davies, J. H. Yin, and S. A. Velastin, "Crowd monitoring using image processing," *Electron. Commun. Eng. J.*, vol. 7, no. 1, pp. 37–47, Feb. 1995.

[14] D. Kong, D. Gray, and H. Tao, "Counting pedestrians in crowds using viewpoint invariant training," in *Proc. Brit. Mach. Vis. Conf*, 2005.

[15] A. N. Marana, L. F. Costa, R. A. Lotufo, and S. A. Velastin, "On the efficacy of texture analysis for crowd monitoring," in *Proc. Comput. Graphics, Image Process. Vis.*, 1998, pp. 354–361.

[16] L. Dong, V. Parameswaran, V. Ramesh, and I. Zoghlami, "Fast crowd segmentation using shape indexing," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[17] A. B. Chan, Z. S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–7.

[18] N. R. Draper and H. Smith*, Applied Regression Analysis*. New York: Wiley-Interscience, 1998.

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

[20] A. C. Cameron and P. K. Trivedi*, Regression Analysis of Count Data*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[21] P. J. Diggle, J. A. Tawn, and R. A. Moyeed, "Model-based geostatistics," *Appl. Statist.*, vol. 47, no. 3, pp. 299–350, 1998.

[22] C. J. Paciorek and M. J. Schervish, "Nonstationary covariance functions for Gaussian process regression," in *Proc. Neural Inf. Process. Syst.*, 2004.

[23] J. Vanhatalo and A. Vehtari, "Sparse log Gaussian processes via MCMC for spatial epidemiology," in *Proc. JMLR Workshop Conf.*, 2007, pp. 73–89.

[24] A. B. Chan and N. Vasconcelos, "Bayesian Poisson regression for crowd counting," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 545–551.

[25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 2, pp. 886–893.

[26] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[27] P. Dollár, B. Babenko, S. Belongie, P. Perona, and Z. Tu, "Multiple component learning for object detection," in *Proc. ECCV*, 2008, pp. 211–224.

[28] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. II-406–II-413.

[29] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: HybridBoosted multi-target tracker for crowded scene," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 2953–2960.

[30] W. Ge and R. T. Collins, "Marked point processes for crowd counting," in *Proc. CVPR*, 2009, pp. 2913–2920.

[31] B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1114–1127, Aug. 2008, Special Issue Video Surveillance.

[32] Y. Cong, H. Gong, S.-C. Zhu, and Y. Tang, "Flow mosaicking: Real-time pedestrian counting without scene-specific learning," in *Proc. CVPR*, 2009, pp. 1093–1100.

[33] V. Lempitsky and A. Zisserman, "Learning to count objects in images," *Adv. Neural Inf. Process. Syst.*, pp. 1324–1332, 2010.

[34] R. P. Adams, I. Murray, and D. J. C. MacKay, "Tractable nonparametric Bayesian inference in poisson processes with Gaussian process intensities," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 9–16.

[35] G. M. El-Sayyad, "Bayesian and classical analysis of poisson regression," *J. Roy. Statist. Soc. Ser. B (Methodological)*, vol. 35, no. 3, pp. 445–451, 1973.

[36] W. Chu and Z. Ghahramani, "Gaussian processes for ordinal regression," *J. Mach. Learn. Res.*, vol. 6, no. 1, pp. 1019–1041, 2004.

[37] M. S. Bartlett and D. G. Kendall, "The statistical analysis of variance-heterogeneity and the logarithmic transformation," *Suppl. J. Roy. Statist. Soc.*, vol. 8, no. 1, pp. 128–138, 1946.

[38] R. L. Prentice, "A log gamma model and its maximum likelihood estimation," *Biometrika*, vol. 61, no. 3, pp. 539–544, 1974.

[39] J. A. Nedler and R. W. M. Wedderburn, "Generalized linear models," *J. R. Statist. Soc. Ser. A*, vol. 135, pp. 370–384, 1972.

[40] G. C. Cawley, G. J. Janacek, and N. L. C. Talbot, "Generalised kernel machines," in *Proc. Int. Joint Conf. Neural Netw.*, 2007, pp. 1720–1725.

[41] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 909–926, May 2008.

[42] A. B. Chan, M. Morrow, and N. Vasconcelos, "Analysis of crowded scenes using holistic properties," in *Proc. 11th IEEE Int. Workshop PETS*, Jun. 2009, pp. 101–108.

[43] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[44] 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 Process.*, 1999, vol. 6, pp. 3521–3524.

[45] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. ICVR*, 2004, pp. 28–31.

[46] A. B. Chan and N. Vasconcelos*, Counting people with low-level features and Bayesian regression* 2010 [Online]. Available: http://www.svcl.ucsd.edu/projects/peoplecnt/journal/, to be published

Antoni B. Chan (M'06) received the B.S. and M.Eng. degrees in electrical engineering from Cornell University, Ithaca, NY, in 2000 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California, San Diego (UCSD), San Diego, in 2008.

From 2001 to 2003, he was a Visiting Scientist with the Vision and Image Analysis Laboratory, Cornell University, Ithaca, NY, and in 2009, he was a Postdoctoral Researcher with the Statistical Visual Computing Laboratory, UCSD. In 2009, he joined the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong, as an Assistant Professor. His research interests include computer vision, machine learning, pattern recognition, and music analysis.

Dr. Chan was the recipient of an NSF IGERT Fellowship from 2006 to 2008.

Nuno Vasconcelos (S'92–M'00–SM'08) received the License in electrical engineering and computer science from the Universidade do Porto, Porto, Portugal, in 1988 and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, MA, in 1993 and 2000, respectively.

From 2000 to 2002, he was a member of the research staff with Compaq Cambridge Research Laboratory, Cambridge, MA, which in 2002, became the HP Cambridge Research Laboratory. In 2003, he joined the Department of Electrical and Computer Engineering, University of California, San Diego, San Diego, where he served as the Head of the Statistical Visual Computing Laboratory. He is the author of more than 75 peer-reviewed publications. His research interests include computer vision, machine learning, signal processing and compression, and multimedia systems.

Dr. Vasconcelos was a recipient of a US National Science Foundation CAREER award and a Hellman Fellowship.