Counting People with Low-Level Features and Bayesian Regression

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

***Abstract*—An approach to the problem of estimating the size of inhomogeneous crowds, 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 textures 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 (GPR) 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 non-linear 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 dataset, 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 two hours of video demonstrate the efficiency and robustness of regression-based crowd size estimation over long periods of time.**

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

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 1) the crowd as a whole, or 2) an individual's deviation from the crowd. In both cases, to accomplish the task it should suffice to build good *models for the patterns of crowd behavior*. Events could then be detected as *variations in these patterns*, and abnormal individual actions could be detected as *outliers* with respect to the crowd behavior.

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]; 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 *sub-components with different dynamics,* and appears in *unconstrained outdoor environments*, such as that of Figure 1. In fact, we argue that, when a "crowd-centric" approach is considered, the problem actually appears to become simpler. We simply segment the crowd into sub-parts 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.

A. B. Chan is with the Dept. of Computer Science, City University of Hong Kong. N. Vasconcelos is with the Dept. of Electrical and Computer Engineering, University of California, San Diego.

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

One important aspect of regression-based counting is the choice of regression function used to map segment features into crowd counts. One possibility is to rely on classical regression methods, such as linear, or piece-wise linear, regression and least squares fits [18]. These methods are not very robust to outliers and non-linearities, and are prone to overfitting when the feature space is high-dimensional or there is 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 non-linearities 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. The main limitation of GPR-based counting is, however, 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 non-integer, or even negative, counts. In result, there is a need for sub-optimal post-processing operations, such as quantization and truncation. Furthermore, continuous crowd estimates increase the complexity of subsequent statistical inference, e.g. graphical models that identify dependencies 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 non-negative 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 of 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 limits these hierarchical models to small datasets. In this work, 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 work are three-fold, 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 non-linearities, 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 datasets of pedestrian imagery, and its robustness demonstrated through results on two 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 outdoors, unconstrained, environment for such periods of time.

The paper is organized as follows. Section II reviews related work in crowd counting. GPR is discussed in Sections 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 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 used as an estimate of the number of moving subjects. Examples 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], piece-wise linear [12], or neural networks [11], [15]. In recent years, regression-based counting has also been applied to outdoor scenes. [14] applies neural networks to the histograms of foreground segment areas and edge orientations. [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 6). [32] counts the number of people crossing a line-of-interest using flow vectors and dynamic mosaics. [33] proposes a supervised learning framework, which estimates an image density whose integral over a region-of-interest yields the count. The main contributions of this work, with respect to previous approaches to regression-based counting, are four-fold: 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 non-linear regression, which enables accurate counts in densely crowded scenes; 3) introduction of a Bayesian model for discrete regression, which is suitable for crowd counting; 4) demonstration that the proposed algorithms can robustly operate on video of unconstrained, outdoor environments, through validation on a large dataset with 2 hours of video.

Regarding Bayesian regression for discrete counts, [21]–[23], [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, [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; 3) we kernelize the regression function, thus modeling non-linear log-arrival rates. Our final contribution is a kernelized, closed-form, efficient approximation to Bayesian Poisson regression. 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 four-fold: 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 dataset and from two viewpoints ([17], [24] only tested one viewpoint); 4) we provide an in-depth comparison between regression-based counting and counting with person detection.

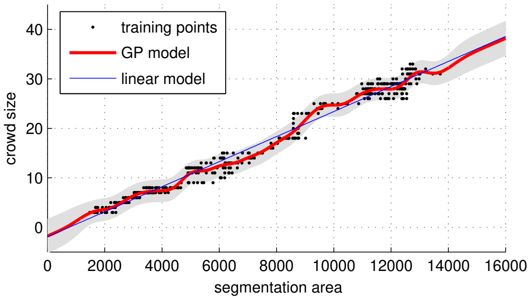

Fig. 2. Correspondence between crowd size and segment area: linear least-squares regression, and a non-linear function learned with Gaussian process regression. Two standard deviations error bars for GPR are plotted (gray area).

III. Gaussian process regression

Figure 1 shows examples of a crowded scene on a pedestrian walkway. We assume that the camera is part of a permanent surveillance installation, and 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 sub-components, 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). Figure 2 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 has 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 non-linearities.

One possibility to implement this regression is to rely on Gaussian process regression (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 $\mathbf{x}$, $\phi : \mathbb{R}^d \to \mathbb{R}^D$. 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.

A. Bayesian prediction

Let $X = [\mathbf{x}_1, \cdots \mathbf{x}_N]$ be the matrix of observed feature vectors $\mathbf{x}_i$, and $\mathbf{y} = [y_1 \ \cdots \ y_N]^T$ 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,

$p(\mathbf{w}|X,\mathbf{y}) = \frac{p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})d\mathbf{w}}$. Given a novel input $\mathbf{x}_*$, the predictive distribution for $f_* = f(\mathbf{x}_*)$ is the average, over all 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

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

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

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 $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \Sigma_p \phi(\mathbf{x}')$, and hence the predictive distribution only depends on inner products between the inputs $\mathbf{x}_i$.

B. Compound kernel functions

The class of functions that can be approximated by GPR depends on the covariance, or 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, while a squared-exponential (RBF) kernel, $k_r(\mathbf{x}, \mathbf{x}') = \theta_1^2 e^{-\frac{1}{\theta_2^2}\|\mathbf{x}-\mathbf{x}'\|^2}$, yields Bayesian regression for locally smooth, infinitely differentiable, functions. As shown in Figure 2, the segment area exhibits a linear trend with the crowd size, with some local non-linearities due to occlusions and segmentation errors. To model the dominant linear trend, as well as these non-linear effects, we can use a compound kernel with linear and RBF components,

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

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

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

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

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 $\{X, \mathbf{y}\}$

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

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

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 *non-negative integers*, $y \in \mathbb{Z}_+ = \{0, 1, 2, \cdots\}$, 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 which is a function of the input variables, resulting in the standard *Poisson* regression or *negative binomial* 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.,

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

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

$$p(y|\mathbf{x}, \beta) = \tfrac{1}{y!}e^{-\lambda(\mathbf{x})}\lambda(\mathbf{x})^y.$$

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

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

Due to the lack of conjugacy between the Poisson likelihood and the Gaussian prior, (11) does not have a closed-form expression, and so 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 $\{X, \mathbf{y}\}$,

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

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

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., $\nu \sim \text{LogGamma}(y + c, 1)$ where

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

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],

$$\text{LogGamma}(k, \theta) \approx \mathcal{N}(\mu, \sigma^2) \quad (15)$$

where the parameters are related by

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

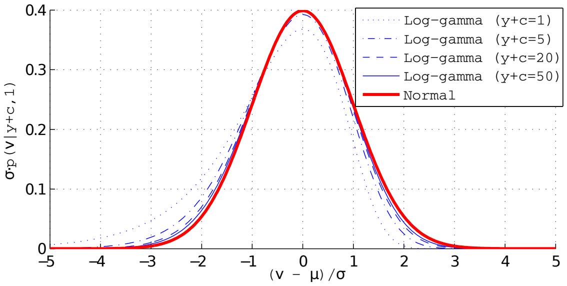

Fig. 3. Gaussian approximation of the log-gamma distribution for different values of $y+c$. The plot is normalized so that the distributions have zero-mean and unit variance.

Hence, (14) can be approximated as

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

Figure 3 illustrates the accuracy of the approximation for different values of $y+c$. Applying (17) to replace the bracket term in (13), and defining $\Phi = [\phi(\mathbf{x}_1)\cdots\phi(\mathbf{x}_N)]$,

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

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

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

$$\begin{aligned} \log p(\beta|X,\mathbf{y}) &\propto \log p(\mathbf{y}|X,\beta) + \log p(\beta) \qquad (20) \\ &\approx -\tfrac{1}{2}\left\|\Phi^T\beta-\mathbf{s}\right\|^2_{\Sigma_y} - c\mathbf{1}^T\Phi^T\beta - \tfrac{1}{2}\left\|\beta\right\|^2_{\Sigma_p}, \end{aligned}$$

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

$$\begin{aligned} \log p(\beta|X,\mathbf{y}) &\propto -\tfrac{1}{2}(\beta^T\Phi\Sigma_y^{-1}\Phi^T\beta - 2\beta^T\Phi\Sigma_y^{-1}\mathbf{s} \\ &\quad + \mathbf{s}^T\Sigma_y^{-1}\mathbf{s}) - c\mathbf{1}^T\Phi^T\beta - \tfrac{1}{2}\beta^T\Sigma_p^{-1}\beta \qquad (21) \\ &\propto -\tfrac{1}{2}[\beta^T(\Phi\Sigma_y^{-1}\Phi^T+\Sigma_p^{-1})\beta - 2\beta^T(\Phi\Sigma_y^{-1}\mathbf{s}-c\Phi\mathbf{1})] \qquad (22) \\ &\propto -\tfrac{1}{2}\left(\beta^T(\Phi\Sigma_y^{-1}\Phi^T+\Sigma_p^{-1})\beta - 2\beta^T\Phi\Sigma_y^{-1}\mathbf{t}\right), \qquad (23) \end{aligned}$$

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

$$p(\beta|X,\mathbf{y}) \approx \mathcal{N}(\beta|\hat{\mu}_\beta,\hat{\Sigma}_\beta), \tag{24}$$

with mean and variance

$$\begin{aligned} \hat{\mu}_\beta &= (\Phi\Sigma_y^{-1}\Phi^T+\Sigma_p^{-1})^{-1}\Phi\Sigma_y^{-1}\mathbf{t}, \qquad (25) \\ \hat{\Sigma}_\beta &= (\Phi\Sigma_y^{-1}\Phi^T+\Sigma_p^{-1})^{-1}. \qquad (26) \end{aligned}$$

Note that setting $c = 0$ will yield the original posterior approximation in [35]. The 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 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 (24) that the posterior distribution of ν_* is approximately Gaussian,

$$p(\nu_*|\mathbf{x}_*,X,\mathbf{y}) \approx \mathcal{N}(\nu_*|\hat{\mu}_\nu,\hat{\sigma}^2_\nu), \tag{27}$$

with mean and variance

$$\begin{aligned} \hat{\mu}_\nu &= \phi(\mathbf{x}_*)^T(\Phi\Sigma_y^{-1}\Phi^T+\Sigma_p^{-1})^{-1}\Phi\Sigma_y^{-1}\mathbf{t}, \qquad (28) \\ \hat{\sigma}^2_\nu &= \phi(\mathbf{x}_*)^T(\Phi\Sigma_y^{-1}\Phi^T+\Sigma_p^{-1})^{-1}\phi(\mathbf{x}_*). \qquad (29) \end{aligned}$$

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

$$\begin{aligned} \hat{\sigma}^2_\nu &= \phi(\mathbf{x}_*)^T(\Sigma_p - \Sigma_p\Phi(\Phi^T\Sigma_p\Phi+\Sigma_y)^{-1}\Phi^T\Sigma_p)\phi(\mathbf{x}_*) \\ &= k(\mathbf{x}_*,\mathbf{x}_*) - \mathbf{k}_*^T(K+\Sigma_y)^{-1}\mathbf{k}_*, \qquad (30) \end{aligned}$$

where $k(\cdot,\cdot)$, K, and $\mathbf{k}_*$ are defined as in Section III-A. Using (41) from the Appendix, the posterior mean $\hat{\mu}_\nu$ can also be rewritten in terms of the kernel function,

$$\begin{aligned} \hat{\mu}_\nu &= \phi(\mathbf{x}_*)^T\Sigma_p\Phi(\Phi^T\Sigma_p\Phi+\Sigma_y)^{-1}\mathbf{t} \qquad (31) \\ &= \mathbf{k}_*^T(K+\Sigma_y)^{-1}\mathbf{t}. \qquad (32) \end{aligned}$$

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 non-linear log-arrival rate functions.

The predictive distribution for y_* is

$$p(y_*|\mathbf{x}_*,X,\mathbf{y}) = \int p(y_*|\nu_*)p(\nu_*|\mathbf{x}_*,X,\mathbf{y})d\nu_*, \tag{33}$$

where $p(y_*|e^{\nu_*})$ is a Poisson distribution of 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)-(16) that ν_* is well approximated by a log-gamma distribution. From $\nu_* = \log\lambda_*$ it then follows that λ_* is approximately gamma distributed,

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

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 $\hat{\sigma}^2_\nu e^{\hat{\mu}_\nu}$. Hence, $\lambda_* \approx e^{\hat{\mu}_\nu}$, which is a sensible approximation. (33) can then be rewritten as

$$p(y_*|\mathbf{x}_*,X,\mathbf{y}) = \int_0^\infty p(y_*|\lambda_*)p(\lambda_*|\mathbf{x}_*,X,\mathbf{y})d\lambda_*, \tag{34}$$

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

$$\begin{aligned} p(y_*|\mathbf{x}_*,X,\mathbf{y}) &= \tfrac{\Gamma(y_*+\hat{\sigma}_\nu^{-2})}{\Gamma(y_*+1)\Gamma(\hat{\sigma}_\nu^{-2})}(\hat{p})^{\hat{\sigma}_\nu^{-2}}(1-\hat{p})^{y_*}, \qquad (35) \\ \hat{p} &= \tfrac{\hat{\sigma}_\nu^{-2}}{\hat{\sigma}_\nu^{-2}+\exp(\hat{\mu}_\nu)}. \qquad (36) \end{aligned}$$

In summary, the predictive distribution of y_* can be approximated by a negative binomial,

$$y_*|\mathbf{x}_*,X,\mathbf{y} \sim \mathrm{NegBin}(e^{\hat{\mu}_\nu},\hat{\sigma}^2_\nu) \tag{37}$$

of mean $e^{\hat{\mu}_\nu}$ and scale $\hat{\sigma}^2_\nu$, given by (28). The prediction variance is $\mathrm{var}(y_*) = e^{\hat{\mu}_\nu}(1+\hat{\sigma}^2_\nu e^{\hat{\mu}_\nu})$, 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

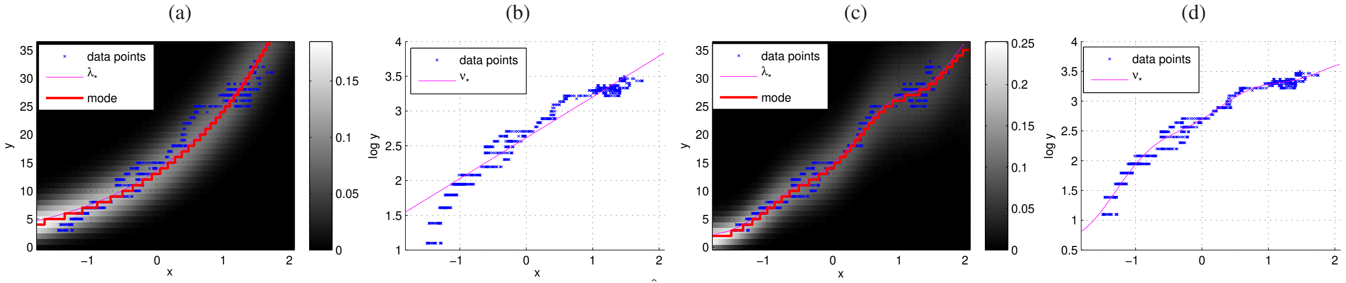

Fig. 4. BPR with (a) linear and (c) RBF kernels. The mean parameter $e^{\hat{\mu}_\nu}$ and the mode are shown superimposed on the NB predictive distribution. The corresponding log-arrival rate functions are shown in (b) and (d).

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. $\mathrm{unc}(y_*) = \hat{\sigma}_\nu e^{\hat{\mu}_\nu}$. Finally, the mode of y_* is adjusted downward depending on the amount of overdispersion, $\mathrm{mode}(y) = \left\{ \begin{smallmatrix} \lfloor (1-\hat{\sigma}_\nu^2) e^{\hat{\mu}_\nu} \rfloor, & \hat{\sigma}_\nu^2 < 1 \\ 0, & \hat{\sigma}_\nu^2 \geq 1 \end{smallmatrix} \right.$, where $\lfloor \cdot \rfloor$ is the floor function.

C. Learning the kernel hyperparameters

The hyperparameters θ of the 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)

$$\log p(\mathbf{y}|X, \theta) \propto -\tfrac{1}{2} \log |K + \Sigma_y| - \tfrac{1}{2} \mathbf{t}^T (K + \Sigma_y)^{-1} \mathbf{t}. \quad (38)$$

Figure 4 presents two examples of BPR learning using the linear and RBF kernels. The predictive distributions are plotted in Figures 4 a) and 4 c), and the the corresponding log-arrival rate functions are plotted in Figures 4 b) and 4 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 Gaussian process regression

The proposed approximate BPR is closely related to GPR. The equations for $\hat{\mu}_\nu$ and $\hat{\sigma}_\nu^2$ in (30, 32) are almost identical to those of the GPR predictive distribution in (4, 5). There are two main differences: 1) the noise term Σ_y of BPR in (30) is dependent on the 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 (32) 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, while 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 a negative binomial of mean $e^{\hat{\mu}_\nu}$ and 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 (38) 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], 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.

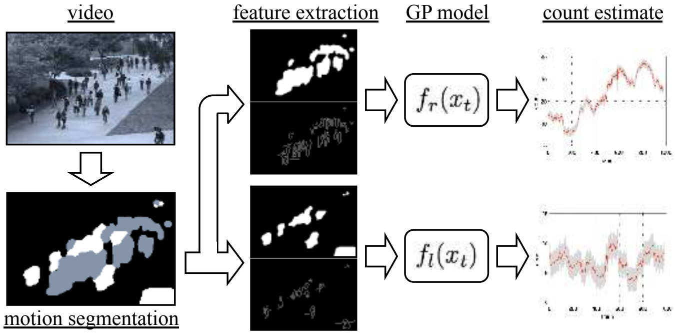

Fig. 5. Crowd counting from low-level features. The scene is segmented into crowds moving in different directions. Features are extracted from each segment and normalized to account for perspective. The number of people in each segment is estimated with Bayesian regression.

V. Crowd counting using low-level features and Bayesian regression

An outline of the proposed crowd counting system is shown in Figure 5. Video is first segmented into crowd regions moving in different directions. Features are then extracted from each crowd segment, after application of a perspective map that weighs pixels according to their approximate size in the 3D 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 sub-components of interest. The goal is to count people moving in different directions or with different speeds. This

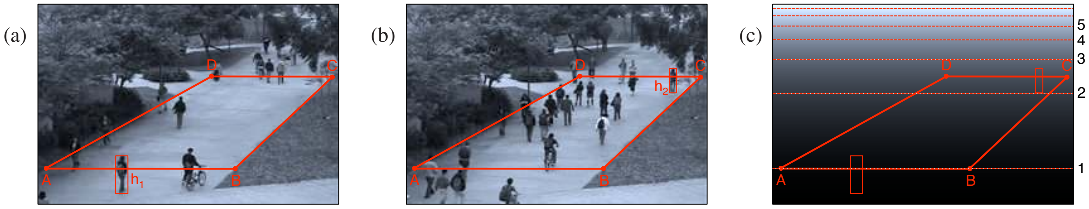

Fig. 6. Perspective map: a) reference person at the front of walkway, and b) at the end; c) the perspective map, which scales pixels by their relative size in the true 3d scene.

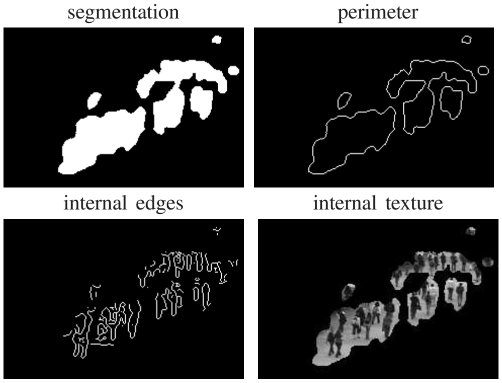

Fig. 7. Examples of the segment mask, segment perimeter, internal edges, and internal texture for the image in Figure 1

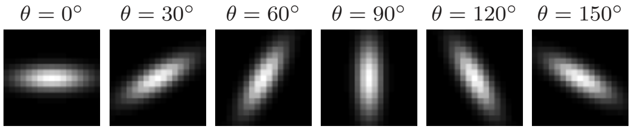

Fig. 8. Filters used to compute edge orientation.

is accomplished by first using a *mixture of dynamic textures* [41] to segment the crowd into sub-components of distinct motion flow. The video is represented as collection of spatio-temporal patches, which are modeled as independent samples from a mixture of dynamic textures. The mixture model is learned with the expectation-maximization (EM) 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 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 work, each pixel is weighted according to a perspective normalization map, based on the expected depth of the object which generated the pixel. Pixel weights encode the relative size of an object at different depths, with larger weights given to far objects.

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 Figure 6 a). It is assumed that 1) $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ form a rectangle in 3D, 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 the heights h_1 and h_2 estimated as the center of the person moves over $\overline{\mathbf{AB}}$ and $\overline{\mathbf{CD}}$, as in Figures 6 a) and 6 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, those on $\overline{\mathbf{CD}}$ weight equal to the area ratio $\frac{h_1 w_1}{h_2 w_2}$, 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. Figure 6 c) shows the resulting perspective map for the scene of Figure 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. For example, 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], [13]. Figure 2 shows a plot of this feature versus the crowd size. While the overall trend is indeed linear, local non-linearities 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 non-linearities, an additional 29 features, 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, computed by morphological erosion with a disk of radius 1.

- *Area* – number of pixels in the segment.
- *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 Figure 8 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 Figure 7), 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" [44].

3) Texture features: Texture features, based on the gray-level co-occurrence matrix (GLCM), were used in [15] to classify image patches into 5 classes of crowd *density* (very low, low, moderate, high, and very high). In this work, we adopt a similar set of measurements for estimating the *number* of pedestrians in each segment. The image is first quantized into 8 gray-levels, and masked by the segment. The joint probability of neighboring pixel values, $p(i,j|\theta)$, is estimated for four orientation, $\theta \in \{0°, 45°, 90°, 135°\}$. A set of three features is extracted for each θ (12 total texture features).

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

Finally, a feature vector is formed by concatenating the 30 features, into a 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 UC San Diego, using a stationary digital camcorder. The first viewpoint, shown in Figure 9 (left), is an oblique view of a walkway, containing a large number of people. The second, shown in Figure 9 (right), 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.

A region-of-interest (ROI) was selected on the main walkway (see Figure 9), and the traveling direction (motion class) and 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 dataset 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 Figure 9.

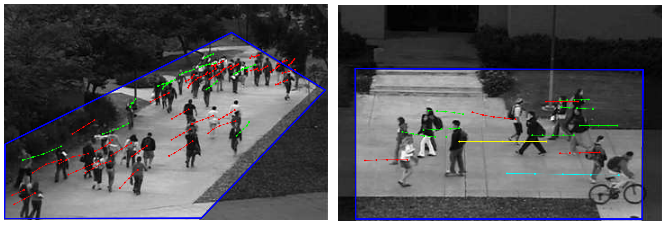

Fig. 9. Ground-truth annotations. (left) Peds1 database: red and green tracks indicate people moving away from, and towards the camera. (right) Peds2 database: red and green tracks indicate people walking right or left, while cyan and yellow tracks indicate fast objects moving right or left. The ROI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

Each database was split into a training set, used to learn the regression model, and a test set, 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 dataset would only test the ability to *interpolate* between the training data, which provides little insight into generalization ability.

B. Experimental Setup

Since Peds1 contains 2 dominant crowd motions ("away" and "towards"), a mixture of dynamic textures [41] with $K = 2$ components was learned from $7 \times 7 \times 20$ spatio-temporal patches, extracted from a short video clip. The model was then used to segment the full video into 2 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 4 dominant crowd motions ("right-slow", "left-slow", "right-fast", or "left-fast"), thus 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 dimensional 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 BPR-l) and the RBF-RBF compound kernel (denoted GPR-rr and BPR-rr). For

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

TABLE I

COMPARISON OF REGRESSION METHODS 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

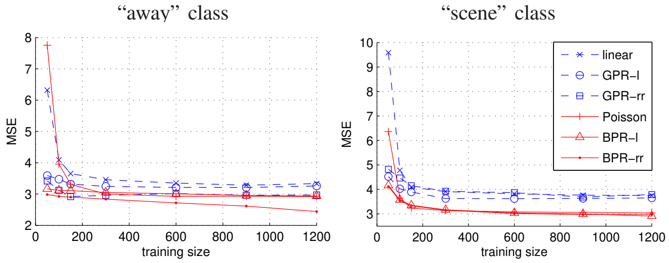

Fig. 10. Error rate for training sets of different sizes on Peds1, for the "away" (left) and "scene" (right) classes. Similar plots were obtained for the "towards" class and are omitted for brevity.

TABLE II

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

	MSE		
Method	away	towards	scene
linear	4.090 (0.609)	3.659 (0.500)	4.780 (0.818)
GPR-l	3.472 (0.288)	**1.923** (0.128)	4.029 (0.298)
GPR-rr	3.118 (0.154)	2.272 (0.604)	4.465 (0.495)
Poisson	3.956 (0.598)	3.605 (0.395)	3.643 (0.370)
BPR-l	3.118 (0.094)	2.358 (0.093)	3.569 (0.141)
BPR-rr	**2.924** (0.093)	2.320 (0.089)	**3.537** (0.127)

TABLE III

COMPARISON OF REGRESSION APPROACHES ON PEDS1 USING DIFFERENT SEGMENTATION METHODS AND $\mathcal{F}_{all}$ ("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

GPR-l and BPR-l, the initial hyperparameters were set to $\theta = [1 \ \cdots \ 1]$, while for GPR-rr and BPR-rr, the optimization was performed over 5 trials with random initializations to avoid bad local maxima. For completeness, standard linear least-squares and Poisson regressions were also tested.

For GPR, counts were estimated by the mean prediction value μ_*, rounded to the nearest non-negative integer. The standard deviation σ_* was used as 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-squared error, MSE $= \frac{1}{M}\sum_{i=1}^{M}(\hat{c}_i - c_i)^2$, and absolute error, err $= \frac{1}{M}\sum_{i=1}^{M}|\hat{c}_i - c_i|$, where c_i and $\hat{c}_i$ are the true and estimated counts for frame i, and M 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 ($\mathcal{F}_t$); segment and edge features ($\mathcal{F}_{se}$). The full set of 30 features is denoted $\mathcal{F}_{all}$. 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 a 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 non-linear 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% 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 ($\mathcal{F}_{se}$) leads to a further decrease to 1.307. Finally, adding the texture features ($\mathcal{F}_{all}$), achieves the lowest error of 1.210. This illustrates the different components of information contributed by the different feature subsets: the estimate produced from segment features is robust but coarse, the refinement by edge and texture features allows the modeling of various non-linearities. Note also that 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 $\mathcal{F}_{all}$. Compared to [13], [14], the full feature set $\mathcal{F}_{all}$ 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. Figure 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 twelve 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

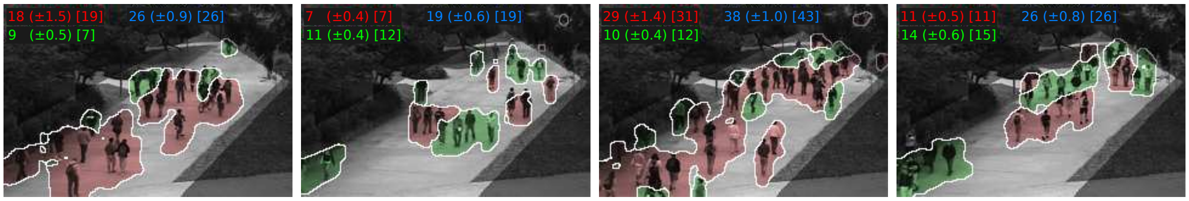

Fig. 11. Crowd counting examples: The red and green segments are the "away" and "towards" components of the crowd. The estimated crowd count for each segment is shown in the top-left, with the (uncertainty) and the [ground-truth]. The prediction for the "scene" class, which is count of the whole scene, is shown in the top-right. The ROI is also highlighted.

better performance than linear or Poisson regression when the training set is small. In practice, this means that 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 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.

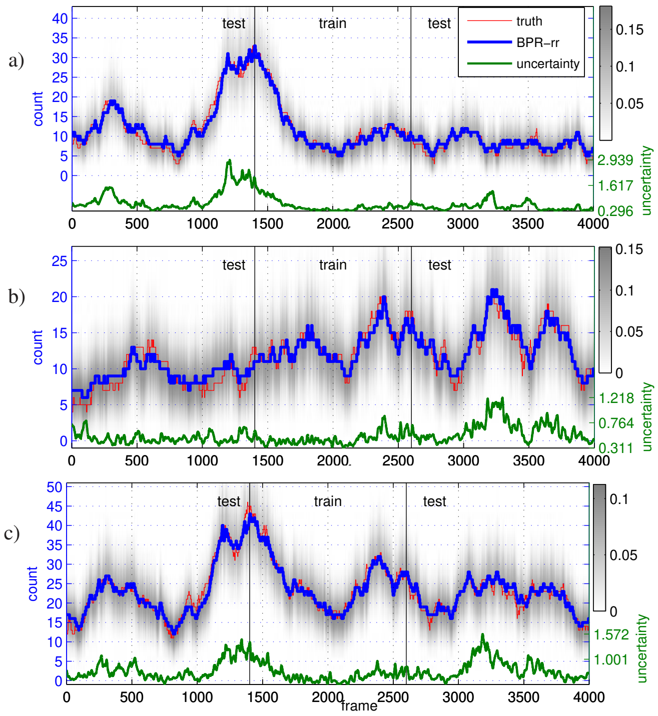

Fig. 12. Crowd counting results on Peds1: a) "away", b) "towards", and c) "scene" classes. Gray levels indicate probabilities of the predictive distribution. The uncertainty is plotted in green, with the axes on the right.

As an alternative to motion segmentation, two background subtraction methods, 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 post-processing steps to the foreground segment: 1) a spatial median filter to remove spurious noise; 2) morphological dilation (disk of radius 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, Figure 12 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. Figure 11 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 non-linear regression model, which is quite robust.

D. Crowd counting results on Peds2

The Peds2 dataset contains smaller crowds (at most 15 people). We found that the segment and edge features ($\mathcal{F}_{se}$) worked the best on this dataset. 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 a 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 $\mathcal{F}_{se}$ features tend to grow linearly with the count. This does not fit well the exponential model that underlies BPR-l. Due to the non-linear 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 $\mathcal{F}_{se}$ are more accurate than those of [13], [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.

Figure 14 shows the crowd count estimates (using $\mathcal{F}_{se}$ and GPR-l) for the five motion classes over time, and Figure 13 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"

TABLE IV
COMPARISON OF REGRESSION METHODS AND FEATURE SETS ON PEDS2.

Feat.	Method	MSE right-slow	MSE left-slow	MSE right-fast	MSE left-fast	MSE scene	MSE total	err right-slow	err left-slow	err right-fast	err left-fast	err scene	err 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

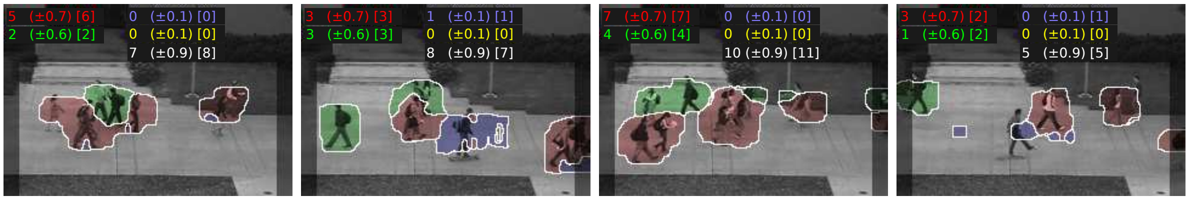

Fig. 13. Counting on Peds2: The estimated counts for the the "right-slow" (red), "left-slow" (green), "right-fast" (blue), and "left-fast" (yellow) components of the crowd are shown in the top-left, with the (uncertainty) and the [ground-truth]. The count for the "scene" class is in white text.

class, where one skateboarder is missed due to an error in the segmentation, as displayed in the last image of Figure 13. In summary, the results on Peds2, again, suggest the efficacy of regression-based crowd counting from low-level features.

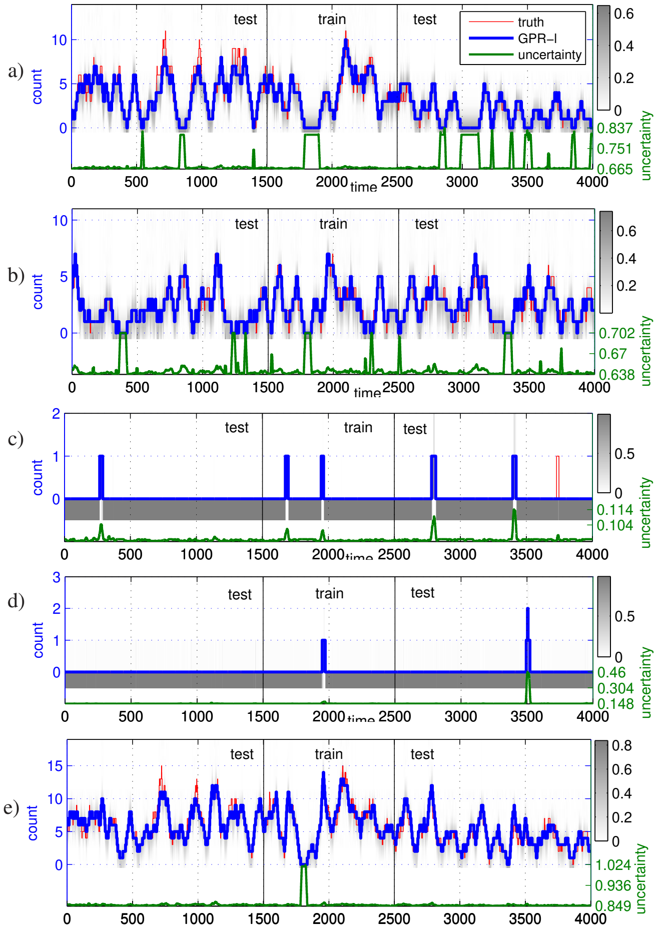

Fig. 14. Crowd counting results on Peds2 for: (a) "right-slow", (b) "left-slow", (c) "right-fast", (d) "left-fast, (e) "scene".

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. The first detects pedestrians with an SVM and the histogram-of-gradients feature [25] (denoted "HOG"). The second is based on a discriminatively-trained deformable parts model [26] (denoted "DPM"). 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. Non-maximum 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. Figure 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 datasets, with a smaller FP rate per image. However, neither algorithm is able to achieve a very high TP rate (the maximum TP rate is 74% on Peds1), due to the large number of occlusions in these scenes.

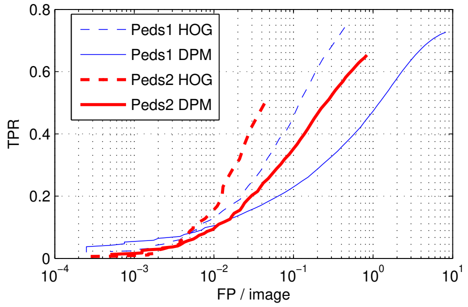

Fig. 15. ROC curves of the pedestrian detectors on Peds1 and Peds2.

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

	Method	MSE	err	bias	var.
Peds1	$\mathcal{F}_{all}$ 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	$\mathcal{F}_{se}$ 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

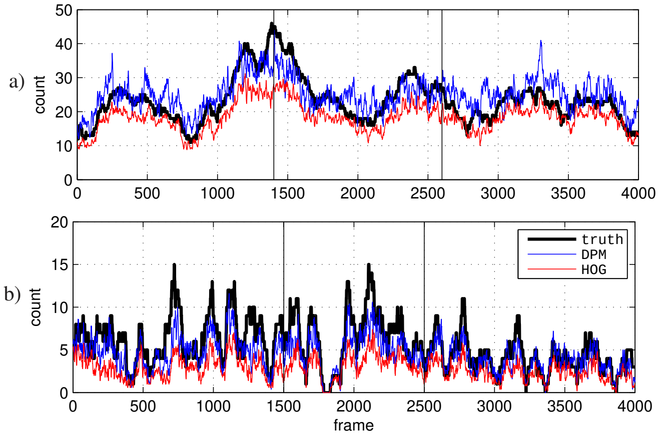

Fig. 16. Crowd counts produced by the HOG [25] and DPM [26] detectors on a) Peds1 and b) Peds2.

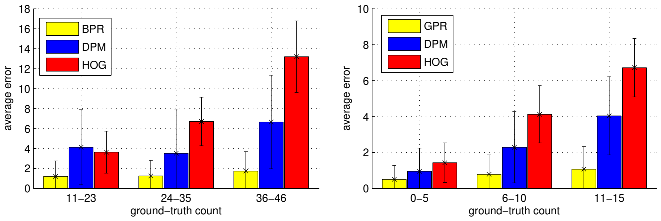

Fig. 17. Error for different crowd sizes on (left) Peds1 and (right) Peds2.

Next, each detector was used to count the number of people in each frame, regardless of 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 $= \frac{1}{M}\sum_{i=1}^{M}(c_i - \hat{c}_i)$ and var $= \frac{1}{M}\sum_{i=1}^{M}(c_i - \text{bias})^2$. The counting performance of DPM and HOG is summarized in Table V, and the crowd counts are displayed in Figure 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 even though the algorithm has a lower TP rate. On the other hand, HOG always underestimates the crowd count, as is evident from Figure 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). Figure 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 occlusions. 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 test set was then computed using the regression function. The (best) results are presented in the bottom-halves of Table V. There is not a significant improvement compared to 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 datasets, the top-performing model and feature set (BPR-rr with $\mathcal{F}_{all}$ for Peds1, and GPR-l with $\mathcal{F}_{se}$ for Peds2) were trained using 2000 frames of the annotated dataset (every other frame). Counts were then estimated on the remaining 50 minutes of each video. Examples of the predictions on Peds1 are shown in Figure 18 (top), and full video results 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 its lack of confidence in these estimates, by assigning them large standard-deviations (e.g. 3rd and 4th images of Figure 18).

A more challenging set of errors occur when bicycles, skateboarders, and golf carts travel quickly on the Peds1 walkway (e.g., 1st image of Figure 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; 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 also displayed in Figure 18 (bottom). 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, for example, three people pulling carts (7th image in Figure 18), or the small truck (final image of Figure 18). 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 detecting them as outliers.

VII. Conclusions

In this work we have proposed the use of Bayesian regression to estimate the size of inhomogeneous crowds, 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 Gaussian process and Bayesian Poisson regression. 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 non-linear log-arrival rates, and that the hyperparameters of the kernel can be estimated by approximate maximum marginal likelihood. Regression-based counting was validated on two large datasets, and shown to provide robust count estimates regardless of the crowd size.

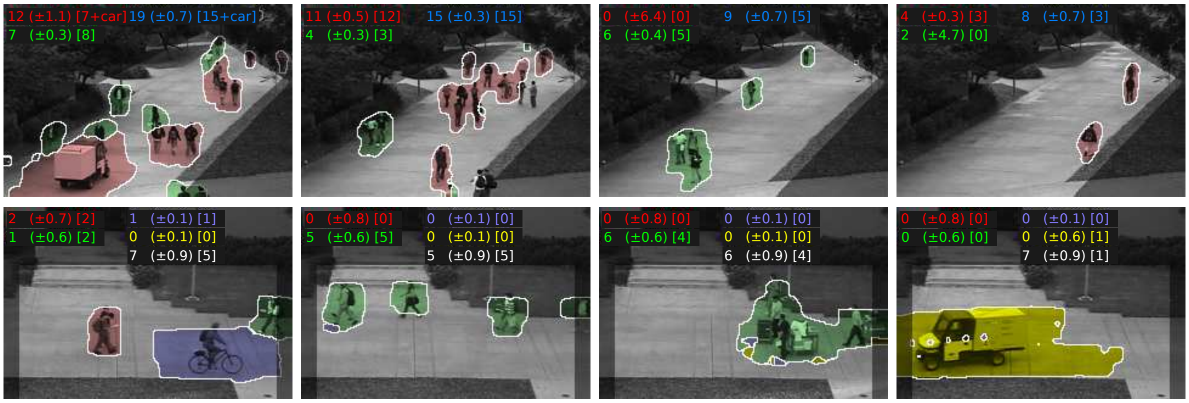

Fig. 18. Example counting results on the full videos: (top) Peds1, and (bottom) Peds2.

Comparing the two Bayesian regression methods, BPR was found more accurate for denser crowds, while GPR performed better when the crowd is 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 two 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; 3) changes in the appearance of dense crowds. Future work will be directed at improving training across viewpoints, by developing perspective invariant features, transferring knowledge across viewpoints (using probabilistic priors), or accounting for 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

$$\Phi\Sigma_y^{-1}(\Phi^T\Sigma_p\Phi+\Sigma_y) = \Phi\Sigma_y^{-1}\Phi^T\Sigma_p\Phi+\Phi \tag{39}$$

$$= (\Phi\Sigma_y^{-1}\Phi^T+\Sigma_p^{-1})\Sigma_p\Phi. \tag{40}$$

Pre-multiplying by $(\Phi\Sigma_y^{-1}\Phi^T+\Sigma_p^{-1})^{-1}$ and post-multiplying by $(\Phi^T\Sigma_p\Phi+\Sigma_y)^{-1}$ yields

$$(\Phi\Sigma_y^{-1}\Phi^T+\Sigma_p^{-1})^{-1}\Phi\Sigma_y^{-1} = \Sigma_p\Phi(\Phi^T\Sigma_p\Phi+\Sigma_y)^{-1}. \tag{41}$$

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 the kernel, $\{\Phi, \Sigma_p, \beta\}$. Using (19), the joint log-likelihood of $\{\mathbf{y}, \beta\}$ can be approximated as

$$\log p(\mathbf{y}, \beta|X, \theta) = \log p(\mathbf{y}|X, \beta, \theta) + \log p(\beta|\theta) \tag{42}$$

$$\approx -\tfrac{N}{2}\log(2\pi) - \tfrac{1}{2}\log|\Sigma_y| - \tfrac{1}{2}\|\Phi^T\beta - \mathbf{s}\|^2_{\Sigma_y} - c\mathbf{1}^T\Phi^T\beta + \sum_{i=1}^{N}\log\tfrac{\Gamma(y_i+c)}{y_i!} - \tfrac{d}{2}\log(2\pi) - \tfrac{1}{2}\log|\Sigma_p| - \tfrac{1}{2}\beta^T\Sigma_p^{-1}\beta \tag{43}$$

$$\propto -\tfrac{1}{2}(\beta^T A\beta - 2\beta^T\Phi\Sigma_y^{-1}\mathbf{s} + 2\beta^T\Phi\mathbf{1}c) - \tfrac{1}{2}\log|\Sigma_p| \tag{44}$$

$$= -\tfrac{1}{2}(\beta^T A\beta - 2\beta^T\Phi\Sigma_y^{-1}\mathbf{t}) - \tfrac{1}{2}\log|\Sigma_p|\,, \tag{45}$$

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,

$$\log p(\mathbf{y}|X, \beta, \theta) + \log p(\beta|\theta) \approx -\tfrac{1}{2}(\left\|\beta - A^{-1}\Phi\Sigma_y^{-1}\mathbf{t}\right\|^2_{A^{-1}} - \mathbf{t}^T\Sigma_y^{-1}\Phi^T A^{-1}\Phi\Sigma_y^{-1}\mathbf{t}) - \tfrac{1}{2}\log|\Sigma_p| \tag{46}$$

$$\propto -\tfrac{1}{2}(\left\|\beta - A^{-1}\Phi\Sigma_y^{-1}\mathbf{t}\right\|^2_{A^{-1}} + \mathbf{t}^T\Sigma_y^{-1}\mathbf{t} - \mathbf{t}^T\Sigma_y^{-1}\Phi^T A^{-1}\Phi\Sigma_y^{-1}\mathbf{t}) - \tfrac{1}{2}\log|\Sigma_p| \tag{47}$$

$$= -\tfrac{1}{2}(\left\|\beta - A^{-1}\Phi\Sigma_y^{-1}\mathbf{t}\right\|^2_{A^{-1}} + \mathbf{t}^T(\Sigma_y + \Phi^T\Sigma_p\Phi)^{-1}\mathbf{t}) - \tfrac{1}{2}\log|\Sigma_p|\,, \tag{48}$$

where in (48) we use the matrix inversion lemma. The marginal likelihood can thus be approximated as,

$$p(\mathbf{y}|X, \beta, \theta) = \int p(\mathbf{y}, \beta|X, \theta)d\beta \tag{49}$$

$$\approx |\Sigma_p|^{-\frac{1}{2}}\, e^{-\frac{1}{2}\mathbf{t}^T(\Sigma_y+\Phi^T\Sigma_p\Phi)^{-1}\mathbf{t}} \int e^{-\frac{1}{2}\left\|\beta - A^{-1}\Phi\Sigma_y^{-1}\mathbf{t}\right\|^2_{A^{-1}}} d\beta$$

$$\propto |\Sigma_p|^{-\frac{1}{2}}\left|A^{-1}\right|^{\frac{1}{2}} e^{-\frac{1}{2}\mathbf{t}^T(\Sigma_y+\Phi^T\Sigma_p\Phi)^{-1}\mathbf{t}} \tag{50}$$

$$= (|\Sigma_p|\,|A|)^{-\frac{1}{2}} e^{-\frac{1}{2}\mathbf{t}^T(\Sigma_y+K)^{-1}\mathbf{t}}. \tag{51}$$

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

$$|A| = |\Sigma_p^{-1} + \Phi\Sigma_y^{-1}\Phi^T| = |\Sigma_p^{-1}||-\Sigma_y^{-1}||-\Sigma_y - \Phi^T\Sigma_p\Phi| = |\Sigma_p^{-1}||\Sigma_y^{-1}||\Sigma_y + K|. \tag{52}$$

Substituting into the log of (51) yields

$$\log p(\mathbf{y}|X, \beta, \theta) \approx \tfrac{1}{2}\log|\Sigma_y| - \tfrac{1}{2}\log|\Phi^T\Sigma_p\Phi + \Sigma_y| - \tfrac{1}{2}\mathbf{t}^T(\Phi^T\Sigma_p\Phi + \Sigma_y)^{-1}\mathbf{t}. \tag{53}$$

Finally, dropping the term that does not depend on the kernel hyperparameters θ yields (38).

Acknowledgments

The authors 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], and P. Dollar for running these algorithms. The authors also thank the anonymous reviewers for their helpful comments. This work was supported by NSF CCF-0830535, IIS-0812235, and the Research Grants Council of the Hong Kong SAR, China [9041552 (CityU 110610)].

References

[1] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *Intl. J. Computer Vision*, vol. 63, no. 2, pp. 153–61, 2005.

[2] T. Zhao and R. Nevatia, "Bayesian human segmentation in crowded situations," in *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, 2003, pp. 459–66.

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

[4] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 875–85.

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

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

[7] V. Rabaud and S. J. Belongie, "Counting crowded moving objects," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2006.

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

[9] B. Leibe, K. Schindler, and L. Van Gool, "Coupled detection and trajectory estimation for multi-object tracking," in *IEEE Intl. Conf. Computer Vision*, 2007.

[10] N. Paragios and V. Ramesh, "A MRF-based approach for real-time subway monitoring," in *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 1034–40.

[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.*, vol. 29, pp. 535–41, 1999.

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

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

[14] D. Kong, D. Gray, and H. Tao, "Counting pedestrians in crowds using viewpoint invariant training," in *British Machine Vision 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. Computer Graphics, Image Processing, and Vision*, 1998, pp. 354–61.

[16] L. Dong, V. Parameswaran, V. Ramesh, and I. Zoghlami, "Fast crowd segmentation using shape indexing," in *IEEE Intl. Conf. Computer Vision*, 2007.

[17] A. B. Chan, Z. S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

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

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

[20] A. C. Cameron and P. K. Trivedi, *Regression analysis of count data*. Cambridge Univ. Press, 1998.

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

[22] C. J. Paciorek and M. J. Schervish, "Nonstationary covariance functions for Gaussian process regression," in *Neural Information Processing Systems*, 2004.

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

[24] A. B. Chan and N. Vasconcelos, "Bayesian Poisson regression for crowd counting," in *IEEE Intl Conf. Computer Vision*, 2009.

[25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection." in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 886–893.

[26] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

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

[28] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2004, pp. II–406–13.

[29] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: HybridBoosted multi-target tracker for crowded scene," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2953–60.

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

[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(8), pp. 1114–1127, August 2008.

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

[33] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems*, 2010.

[34] R. P. Adams, I. Murray, and D. J. C. MacKay, "Tractable nonparametric bayesian inference in poisson processes with gaussian process intensities," in *Intl. Conf. Machine Learning*, 2009.

[35] G. M. El-Sayyad, "Bayesian and classical analysis of poisson regression," *J. of the Royal Statistical Society. Series B (Methodological).*, vol. 35, no. 3, pp. 445–51, 1973.

[36] W. Chu and Z. Ghahramani, "Gaussian processes for ordinal regression," *Journal of Machine Learning and Research*, pp. 1–48, 2005.

[37] M. S. Bartlett and D. G. Kendall, "The statistical analysis of variance-heterogeneity and the logarithmic transformation," *Supplement to the J. of the Royal Statistical Society*, vol. 8, no. 1, pp. 128–38, 1946.

[38] R. L. Prentice, "A log gamma model and its maximum likelihood estimation," *Biometrika*, vol. 61, no. 3, pp. 539–44, 1974.

[39] J. A. Nedler and R. W. M. Wedderburn, "Generalized linear models," *J. of the Royal Statistical Society, Series A*, vol. 135, pp. 370–84, 1972.

[40] G. C. Cawley, G. J. Janacek, and N. L. C. Talbot, "Generalised kernel machines," in *Intl. Joint Conf. on Neural Networks*, 2007, pp. 1720–25.

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

[42] A. Chan, M. Morrow, and N. Vasconcelos, "Analysis of crowded scenes using holistic properties," in *11th IEEE Intl. Workshop on Performance Evaluation of Tracking and Surveillance (PETS'09)*, June 2009.

[43] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–714, 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 Processing*, vol. 6, 1999, pp. 3521–4.

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

[46] http://www.svcl.ucsd.edu/projects/peoplecnt/journal/

Antoni B. Chan 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), in 2008. From 2001 to 2003, he was a Visiting Scientist in the Vision and Image Analysis Lab, Cornell University, and in 2009, he was a Postdoctoral Researcher in the Statistical Visual Computing Lab at UCSD. In 2009, he joined the Department of Computer Science at the City University of Hong Kong, as an Assistant Professor. From 2006 to 2008, he was the recipient of a NSF IGERT Fellowship. His research interests are in computer vision, machine learning, pattern recognition, and music analysis.

Nuno Vasconcelos received the licenciatura in electrical engineering and computer science from the Universidade do Porto, Portugal, in 1988, and the MS and PhD degrees from the Massachusetts Institute of Technology in 1993 and 2000, respectively. From 2000 to 2002, he was a member of the research staff at the Compaq Cambridge Research Laboratory, which in 2002 became the HP Cambridge Research Laboratory. In 2003, he joined the Electrical and Computer Engineering Department at the University of California, San Diego, where he heads the Statistical Visual Computing Laboratory. He is the recipient of a US NSF CAREER award, a Hellman Fellowship, and has authored more than 75 peer-reviewed publications. His work spans various areas, including computer vision, machine learning, signal processing and compression, and multimedia systems. He is a senior member of the IEEE.