IOSR Journal of Electronics & Communication Engineering (IOSR-JECE)
ISSN(e) : 2278-1684 ISSN(p) : 2320-334X, PP 39-58
www.iosrjournals.org

TRAFFIC ANALYSIS USING DISCRETE WAVELET TRANSFORM AND BAYESIAN REGRESSION

Nishidha.T, Dr. P.Janardhanan
[1] *(Electronics & Communication, KMCT College of Engineering / University of Calicut, India)*
[2] *(Electronics & Communication, KMCT College of Engineering / University of Calicut, India)*

Abstract : *Traffic analysis using Discrete Wavelet Transform and Bayesian Regression is used to estimating the size of inhomogeneous traffic, composed of vehicles that travel in different directions without using explicit object segmentation or tracking is proposed .Using the dynamic texture motion model, here the traffic is segmented into components of homogeneous motion. From each segmented region, a set of holistic low-level features are extracted using 4-level discrete wavelet transform. Using the 4 level discrete wavelet transform, I calculate the energy of wavelet coefficients and a function that map features into estimates of the number of vehicle per segment is learned with Bayesian regression. Here two Bayesian regression models are examined. The first is a 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. I addressed 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. Experimental results show that regression-based counts are accurate regardless of the traffic size. Velocity of each car can be calculated.*

Keywords – *Bayesian regression, Discrete Wavelet Transform, Gaussian processes, Poisson regression, Traffic analysis.*

1 Introduction

There is 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. Here I focus on detecting, tracking and analyzing vehicles. In a vehicle-centric approach each vehicle is depicted by a few image pixels and vehicle occlude each other in complex ways. There are many problems in monitoring that can be solved without explicit tracking of vehicles[1]. Abnormal vehicle actions could be detected as outliers with respect to the traffic behavior. The traffic-centric approach analyzes the low-level features extracted from traffic to produce accurate counts. Here I simply segment the traffic into subparts of interest, extract a set of holistic features from each segment and estimate the traffic size with a suitable regression function. By bypassing intermediate processing stages, such as vehicle detection or tracking, which is susceptible to occlusion problems, the proposed approach produces robust and accurate traffic counts, even when the traffic is large and dense. Velocity of each car can be calculated[2].

2 Previous work

Traffic analysis using discrete wavelet transform and Bayesian follow three factors 1) vehicle detection 2) visual feature trajectory clustering 3) regression.

I referred the papers "Detecting pedestrians using patterns of motion and appearance" by P.Viola and M.Jones and D.Snow,2005[2],"Pedestrain detecton in crowded scenes" by B.Leibe, E.Seemann and B.Schiele,2005[5] and paper "segmentation and tracking of multiple frames in crowded environments" by T.Zhao, R.Nevatia and B.Wu[4].In this papers vehicle detection algorithms can be based on motion features and histogram of gradients. Because they detect whole vehicles, these methods are not very effective in densely traffic scenes. This problem has been addressed to some extent by the development of part-based detectors. This method is used in the "Multiple component learning for object detection by P Dollan , B Babenko in 2008[3].

Next I referred the papers "Counting crowded moving objects" by V.Rabaud and S.J. Belongie[7] , "Unsupervised Bayesian detection of independent motion of crowds" by G.J Brostow and R.Cipolla[8] and "A survey of vision-based trajectory learning of survelliance" by B.T Morris and M.M Trivedi [12]. This paper consists of identifying and tracking visual features over time Feature trajectories that exhibit coherent motion are clustered. Counting of feature trajectories has two disadvantages 1) Handlying broken feature tracks due to occlusion [it requires sophisticated trajectory management] 2) In densely traffic environment, 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.

Then I referred "Regression analysis of count data" by A.C Cameron and P.K Trivedi [11] and "Applied regression analysis" by N.R Droper [10] here Bayesian Poisson Regression was found more accurate for denser crowds, whereas Gaussian Process Regression performed better when the crowd was less dense. Next I Referred "Estimation of number of people in crowed scenes uses perspective transformation" by S.F Lin, J.Y Chen and H.X Chao[6]. In this 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 weight given to far objects.

In the regression based traffic counting two Bayesian regression models are examined. The first is a Gaussian Process Regression and the second is a Bayesian treatment of Poisson Regression. Here a function that map features into the estimates of the number of vehicle per segment in learned with Bayesian Regression. In the base paper "Counting People with low-level features and Bayesian regression" by Antoni B [1] pedestrians are counted from the crowd. In this project I count the number of cars from the traffic.

3 *Traffic Analysis using Discrete Wavelet Transform and Bayesian Regression*

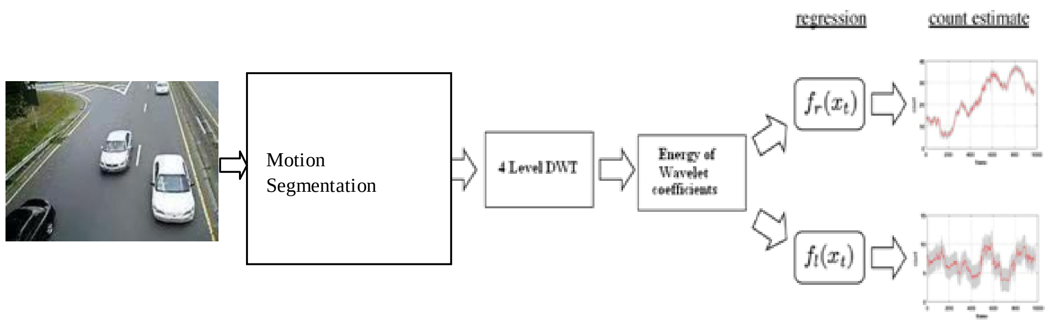


Fig. 1 Traffic Analysis using Discrete Wavelet Transform and Bayesian Regression

An outline of the proposed traffic analysis using discrete wavelet transform and Bayesian regression is shown in figure 1.The video is first segmented into traffic regions moving in different directions. Features are then extracted from each traffic 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 cars per segment is estimated from the feature vector, using the BPR module.

3.1 Traffic Segmentation

The first step of the system is to segment the scene into the traffic subcomponents of interest. The goal is to count cars moving in different directions or with different speeds. This is accomplished by first using a mixture of dynamic textures [1] to segment the traffic 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. 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.

3.2 Perspective Normalization

The extraction of features from traffic 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.

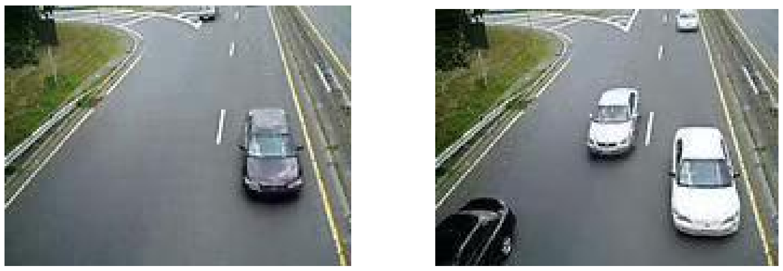


Fig.2.(a) reference car at the front of walkway, and (b) at the end

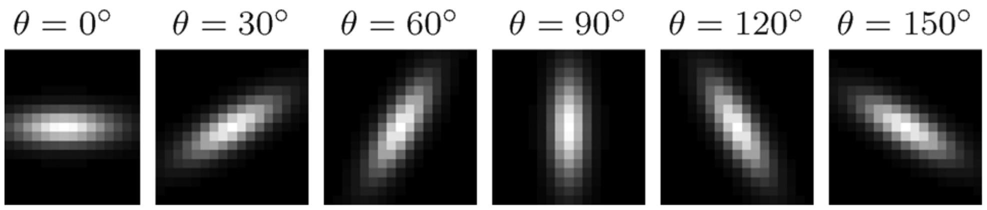


Fig. 3. Filters used to compute edge orientation

The perspective map is estimated by linearly interpolating the size of a reference car between two extremes of the scene. First, a rectangle is marked in the ground plane, by specifying points {A,B,C,D},. It is assumed that 1) {A, B, C, D} form a rectangle in 3-D, and 2) $\overline{AB}$ and $\overline{CD}$ are horizontal lines in the image plane. A reference car is then selected in the video, and heights h1 and h2 estimated as the center of the person move over $\overline{AB}$ and $\overline{CD}$. 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 $\overline{AD}$ receive weight 1and those on $\overline{CD}$ weight equal to the area ratio (h1w1)/(h2w2), where w1 is the length of $\overline{(AB)}$ and w2 is the length of $\overline{(CD}$).The remaining pixel weights are obtained by linearly interpolating the width of the rectangle and the height of the reference carat each image coordinate and computing the area ratio. In this case, objects in the foreground ($\overline{AB)}$ are approximately 2.4 times bigger than objects in the background $\overline{(CD)}$. In other words, pixels on $\overline{CD}$ are weighted 2.4 times as much as pixels on $\overline{AB}$.

a. Feature Extraction using wavelet transform

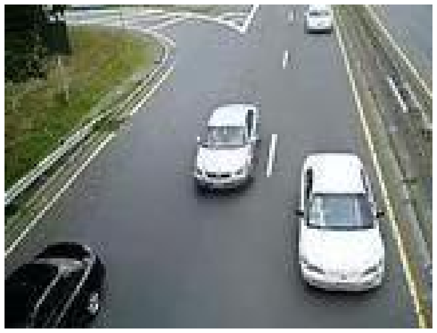


Fig. 4 Cars moving towards the camera.

3.3.1 Discrete Wavelet Transform

In numerical and functional analysis, a discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. As with other wavelet transforms, a key advantage it has over Fourier transforms is temporal resolution: it captures both frequency and location information (location in time)[3].

The first DWT was invented by the Hungarian mathematician AlfrédHaar. For an input represented by a list of 2^nnumbers, the Haar wavelet transform may be considered to simply pair up input values, storing the difference and passing the sum. This process is repeated recursively, pairing up the sums to provide the next scale: finally resulting in $2^n - 1$differences and one final sum.

The most commonly used set of discrete wavelet transforms was formulated by the Belgian mathematician Ingrid Daubechies in 1988. This formulation is based on the use of recurrence relations to generate progressively finer discrete samplings of an implicit mother wavelet function; each resolution is twice that of the previous scale. In her seminal paper, Daubechies derives a family of wavelets, the first of which is the Haar wavelet. Interest in this field has exploded since then, and many variations of Daubechies' original wavelets were developed [5].

The Dual-Tree Complex Wavelet Transform (ℂWT) is relatively recent enhancement to the discrete wavelet transform (DWT), with important additional properties: It is nearly shift invariant and directionally selective in two and higher dimensions. It achieves this with a redundancy factor of only $2^d$for d-dimensional signals, which is substantially lower than the un-decimated DWT. The multidimensional (M-D) dual-tree ℂWT is no separable but is based on a computationally efficient separable filter bank (FB).

Discrete wavelet transforms derived features used for digital image texture analysis. Wavelets appear to be a suitable tool for this task, because they allow analysis of images at various levels of resolution. The main feature of DWT is multi scale representation of function. By using the wavelet, given function can be analyzed at various levels of resolution. The DWT is also invertible and can be orthogonal. Wavelets seem to be effective for analysis of textures recorded with different resolution DWT derived texture feature is a vector, which contains energies of wavelet coefficients calculated in sub bands at successive scales[7].

In principle, features such as segment area should vary linearly with the number of people in the scene. . 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 cars within a segment). To model these nonlinearities, additional29 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.

3.3.2 Segment Features

Using wavelet transform 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.

- ❖ 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
- ❖ 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 cars contained within.
- ❖ "Blob" count—number of connected components, with more than 10 pixels, in the segment.

3.3.3 Internal Edge Features

The edges within a traffic segment are a strong clue about the number of cars in it. A Canny edge detector [2] is applied to the image, the output is masked to form the internal edge image and a number of features are extracted.

- ❖ Edge length—number of edge pixels in the segment.

- Edge orientation—6-bin histogram of edge orientations.
- Murkowski dimension—fractal dimension of the internal edges, which estimates the degree of "space-filling" of the edges [2].

3.3.4 Texture Features

Texture features, which are based on the gray-level co-occurrence matrix, were used in [1] 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 cars in each segment. The image is first quantized into eight gray levels and masked by the segment. The joint probability of neighboring pixel values is then estimated for four orientations. A set of three features is extracted for each for a total of 12 texture features

- Homogeneity: the texture smoothness,
- .Energy: the total sum-squared energy,
- .Entropy: the randomness of the texture distribution,

Finally, a feature vector is formed by concatenating the 30 features, into the vector. Velocity of each car can be calculated.

$$\text{Velocity}=\frac{Length}{No\ of\ frames}\,x\ frame\ rats$$

4 Regression Models

4.1 Gaussian Process Regression

Fig. 1 shows examples of a traffic scene on the road. I assume that the camera is part of a permanent surveillance installation; hence, the viewpoint is fixed. The goal of traffic counting is to estimate the number of cars 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 cars in that segment. Intuitively, assuming proper normalization for the scene perspective, one such feature is the area of the traffic segment (number of segment pixels). Fig. 2(a) plots the segment area versus the traffic 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 traffic 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 *(x)* of a feature vector, $\mathbf{x} \in \mathbb{R}^d$ from a training sample. Let Ø (x) be a high-dimensional feature transformation of x, Ø: $\mathbb{R}^d \to \mathbb{R}^D$, . Consider the case where *f (x)* is linear in the transformation space and the target count *y* modelled as

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

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

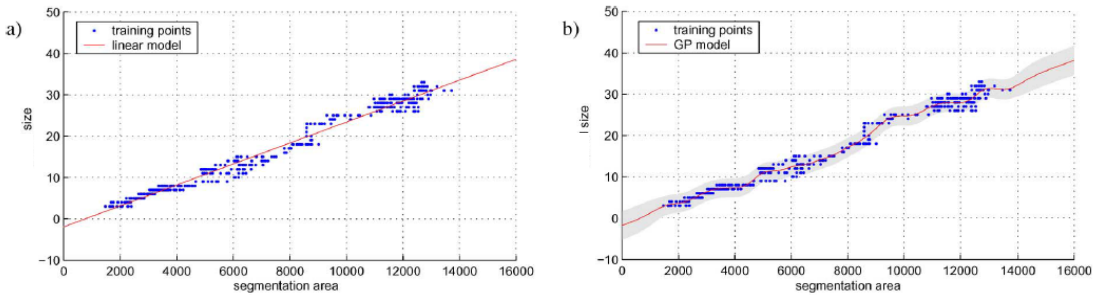


Fig. 2. Correspondence between traffic size and segment area. (a) Line learned with least squares regression. (b) Nonlinear function learned with GPR. The two standard deviations error bars are plotted (gray area).

4.1.1 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 **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]

$$p(f_*|\mathbf{x}_*, X, \mathbf{y}) = \int p(f_*|\mathbf{x}_*, \mathbf{w}) p(\mathbf{w}|X, \mathbf{y}) d\mathbf{w} \qquad (2)$$
$$= \mathcal{N}(f_*|\mu_*, \sigma_*^2) \qquad (3)$$

where the predictive mean and covariance are

$$\mu_* = \mathbf{k}_*^T \left(K + \sigma_n^2 I\right)^{-1} \mathbf{y} \qquad (4)$$
$$\sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T \left(K + \sigma_n^2 I\right)^{-1} \mathbf{k}_*. \qquad (5)$$

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 $\mathbf{x}_i$.

4.1.2 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 traffic 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:

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

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

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

The first RBF has a larger scale parameter □2 and models the overall trend, whereas the second relies on a smaller scale parameter □4 to model local non linearities.
The kernel hyper parameters □i can be estimated from a training sample by Type-II maximum likelihood, which maximizes the marginal likelihood of the training data {X, y}

$$\begin{aligned}\log p(\mathbf{y}|X,\theta) &= \log \int p(\mathbf{y}|\mathbf{w}, X, \theta)p(\mathbf{w}|\theta)d\mathbf{w} && (8)\\ &= -\frac{1}{2}\mathbf{y}^T K_y^{-1}\mathbf{y} - \frac{1}{2}\log|K_y| - \frac{N}{2}\log 2\pi && (9)\end{aligned}$$

where $K_y = K + \sigma_n^2 I$, with respect to the parameters □ , e.g.,using standard gradient ascent methods.

4.2 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 [4]. 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 hyper parameters 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.,

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

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 **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, 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.

4.2.1 Approximate Posterior Distribution

We first derive a closed-form approximation to the posterior distribution in (11), which is based on the approximation of [5]. Consider the data likelihood of a training set {X, y} as follows:

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

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

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

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

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 [4].

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

where the parameters are related by

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

Hence, (14) can be approximated as

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

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 $\Phi = [\phi(\mathbf{x}_1) \cdots \phi(\mathbf{x}_N)]$

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

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

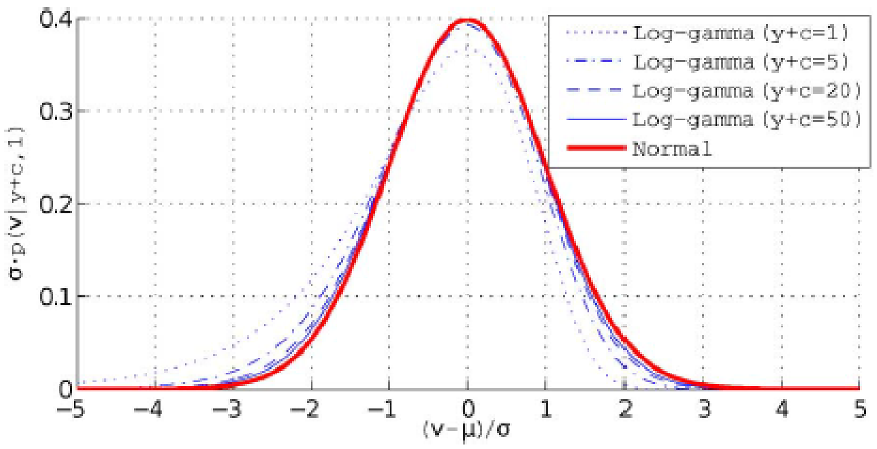


Fig. 3. Gaussian approximation of the log-gamma distribution for different values of $y+c$. The plot is normalized so that the distributions have zero mean and unit variance.

where $\Sigma_y = \text{diag}([1/(y_1+c)\cdots 1/(y_N+c)])$, and $\mathbf{s} = \log(\mathbf{y}+ \ c)$ is the element wise 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

$$\begin{aligned}
&\log p(\beta|X,\mathbf{y}) \\
&\quad\propto -\frac{1}{2}\left(\beta^T\Phi\Sigma_y^{-1}\Phi^T\beta - 2\beta^T\Phi\Sigma_y^{-1}\mathbf{s} + \mathbf{s}^T\Sigma_y^{-1}\mathbf{s}\right) \qquad (22)\\
&\qquad - c\mathbf{1}^T\Phi^T\beta - \frac{1}{2}\beta^T\Sigma_p^{-1}\beta \\
&\quad\propto -\frac{1}{2}\left[\beta^T\left(\Phi\Sigma_y^{-1}\Phi^T + \Sigma_p^{-1}\right)\beta - 2\beta^T\left(\Phi\Sigma_y^{-1}\mathbf{s} - c\Phi\mathbf{1}\right)\right] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (23)\\
&\quad= \frac{1}{2}\left(\beta^T\left(\Phi\Sigma_y^{-1}\Phi^T + \Sigma_p^{-1}\right)\beta - 2\beta^T\Phi\Sigma_y^{-1}\mathbf{t}\right) \qquad (24)
\end{aligned}$$

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.,

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

with mean and variance

$$\begin{aligned}
\hat{\mu}_\beta &= \left(\Phi\Sigma_y^{-1}\Phi^T + \Sigma_p^{-1}\right)^{-1}\Phi\Sigma_y^{-1}\mathbf{t} \qquad (26)\\
\hat{\Sigma}_\beta &= \left(\Phi\Sigma_y^{-1}\Phi^T + \Sigma_p^{-1}\right)^{-1}. \qquad (27)
\end{aligned}$$

Note that setting *c* = 0 will yield the original posterior approximation in [5]. 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.

4.2.2 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:

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

(a) (b) (c) (d)

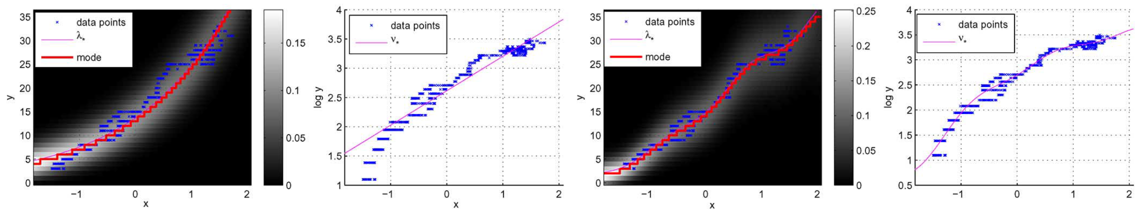


Fig. 4. BPR with (a) linear and (c) RBF kernels. The mean parameter $e^{\hat{\mu}_\nu}$ and the mode are shown superimposed on the NB predictive distribution. The corresponding log-arrival rate functions are shown in (b) and (d).

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

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

The posterior mean $\hat{\mu}_\nu$ can also be rewritten in terms of the kernel function

$$\hat{\mu}_\nu = \phi(\mathbf{x}_*)^T\Sigma_p\Phi(\Phi^T\Sigma_p\Phi + \Sigma_y)^{-1}\mathbf{t} \qquad (32)$$

$$= \mathbf{k}_*^T(K + \Sigma_y)^{-1}\mathbf{t}. \qquad (33)$$

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 $\hat{\sigma}_\nu^{-2}$ arrivals of the Poisson process is modeled as the time between 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.,

In sı

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

ollows:

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

of mean $e^{\hat{\mu}_\nu}$ and scale $\hat{\sigma}_\nu^2$, given by (29). The prediction variance is $\mathrm{var}(y_*) = e^{\hat{\mu}_\nu}(1 + \hat{\sigma}_\nu^2 e^{\hat{\mu}_\nu})$ 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., $\mathrm{unc}(y_*) = \hat{\sigma}_\nu e^{\hat{\mu}_\nu}$. Finally, the mode of y_* is adjusted downward depending on the amount of overdispersion, i.e., 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.

4.2.3 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.

4.2.4 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 identical 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 t (assuming *c* = 0), rather than the counts *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 $\{\bar{X}, \mathbf{y}\}$ 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 [5], which is a general regression framework for linear covariate regression problems. Generalized kernel machines and the associated kernel Poisson regression were proposed in [4]. The proposed BPR is a Bayesian formulation of kernel Poisson regression.

5 Algorithm

1) Start
2) Input is the traffic video.
3) Traffic video is segmented into components of homogeneous motion using dynamic texture motion model.
4) Using 4-level DWT, a set of holistic low-level features are extracted from each segmented region.
5) By using these holistic low-level features from 4-level DWT, I calculate the energy of wavelet coefficients.
6) A function that features into estimates of the number of vehicle per segment is learned with Bayesian Regression models.
7) BPR was found more accurate for denser crowds, whereas GPR performed better when the crowd was less dense.
8) Velocity of each car is calculated.
9) Output is the regression based count, which is accurate regardless of the traffic size.
10) Stop.

6 Result

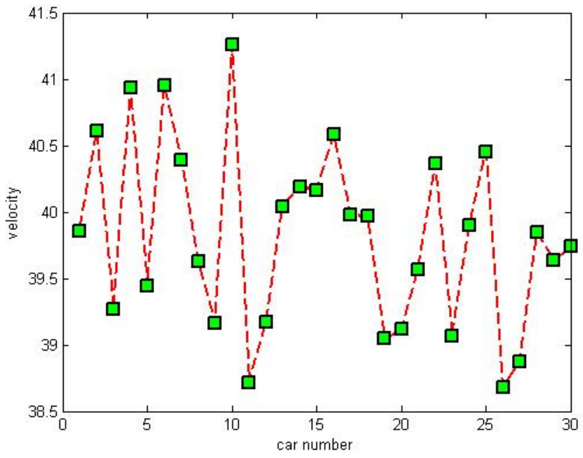
Input video frames

The traffic video contains 30 cars travel with a uniform velocity 40km/hour. A simulated output is got after running the program in MATLAB. Velocity of each car is estimated. A graph is plotted between car number and velocity.

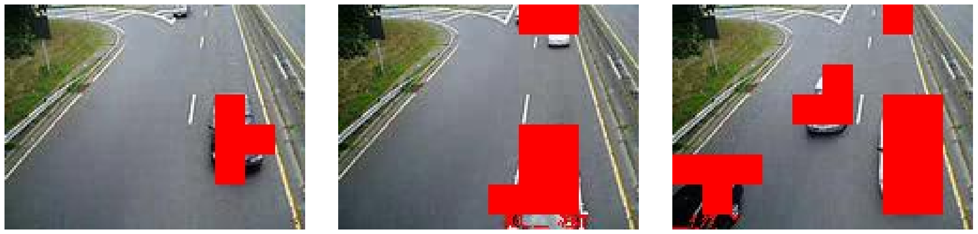
Min. value of velocity = 39.8192

Mean Value of Velocity = 38.75

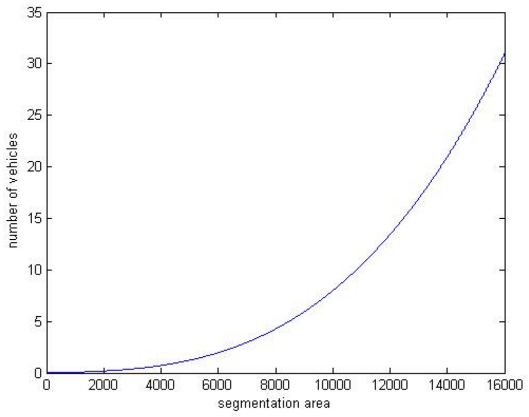
Higher value of Velocity = 41.25



Output video frames



A graph is plotted between number of car and segmentation area. Traffic video contains 30 cars. When segmentation area increases the number of cars also increases and got the graph.



7 Conclusion

In this paper, I have proposed the use of Bayesian regression to estimate the size of inhomogeneous traffic, which are composed of cars 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 hyper parameters of the kernel can be estimated by approximate maximum marginal likelihood. Regression-based counting was validated on large data sets and was shown to provide robust count estimates regardless of the traffic 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 traffic counting approaches. The regression-based count estimates were also shown substantially more accurate than those produced by state-of-the-art vehicle detectors. Finally, regression-based counting was successfully applied to the video, suggesting that systems based on the proposed approach could be used in real-world environments for long periods of time. Velocity of each car is calculated so abnormal actions of the cars are detected. Here traffic is analysed using discrete wavelet transform so the output is more accurate.

One limitation, for traffic 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 traffic counting system . 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 traffic. 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 traffic 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 traffic. This derives from the Poisson noise model and diminishes the ability of BPR to model local variations of sparse traffic (in the presence of count uncertainty, Bayesian regression tends to smoothen the regression mapping). Future work will study noise models without this restriction.

8 Acknowledgement

I would like to thank my project guide Dr. P. Janardhanan. Principal of KMCT college of Engineering for his helpful comments for completing this project.

9 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] *Counting People with Low-Level Features and Bayesian Regression*. Antoni B. Chan, *Member, IEEE,* and Nuno Vasconcelos, *Senior Member, IEEE*

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

[5] 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.*

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

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