2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)

Counting People by Clustering Person Detector Outputs

Ibrahim Saygin Topkaya
Sabanci University
Alcatel-Lucent Teletas
Istanbul, Turkey
isaygint@sabanciuniv.edu

Hakan Erdogan
Sabanci University
Istanbul, Turkey
haerdogan@sabanciuniv.edu

Fatih Porikli
Australian National University
NICTA
Canberra, Australia
fatih.porikli@anu.edu.au

Abstract

We present a people counting system that estimates the number of people in a scene by employing a clustering scheme based on Dirichlet Process Mixture Models (DPMMs) which takes outputs of a person detector system as input. For each frame, we run a person detector on the frame, take its output as a set of detection areas and define a set of features based on spatial, color and temporal information for each detection. Then using these features, we cluster the detections using DPMMs and Gibbs sampling while having no restriction on the number of clusters, thus can estimate an arbitrary number of people or groups of people. We finally define a measure to calculate the actual number of people within each cluster to infer the final estimation of the number of people in the scene.

1. Introduction

People counting is one of the most fundamental yet challenging computer vision tasks. Existing solutions can be broadly categorized into three groups; detection based, regression based and tracking methods.

Detection based methods infer the number of people in the scene from region classifiers that designed to locate humans or human body parts. For instance, [1] uses a head detector to determine the number of people by applying a classifier that is trained with color and orientation of gradients features around a set of chosen interest points. However, a direct employment of detectors is sensitive to occlusions and imperfect detector responses. Regression based methods learn a function of linear or nonlinear correspondences between image features and the number of people in the training data, and then employ the learned function to estimate the number of people. For instance, [2], computes a fixed ratio between the number of extracted foreground corners and the number of people and [3] extracts interest points, clusters them, and trains a regressor on the number of interest points and the number of people in the cluster. Regression based approaches require different sets of training data with different camera setups, thus their generality is limited. Tracking based methods draw the number of people by grouping similar trajectory segments. For instance, in [4] a model based tracker is used to generate short trajectories, which are grouped into unique tracks per person using spatial and temporal consistency heuristics. In addition to the inherent issues in tracking, these methods assume multiple trajectories for each person, thus relatively larger human regions in images.

In this work we present a novel clustering based framework that takes responses of a generic person detector [5] as its input instead of trajectories. Since even the best generic human detectors have inconsistent outputs and one person can be detected multiple times because of overlapping search windows and repetitive searches through pyramidal multi-scale schemes (e.g. blue bordered detections in Figure 2b), a post-detection bundling step is crucial to distinguish the individual people in the scene. For this we fuse different types of color, spatial and temporal features into clustering.

Our method is based on the Dirichlet Process Mixture Model (DPMM) [6], which recently has received increasing attention for computer vision applications particularly for object tracking purposes [7]. One advantage of DPMMs is their implicit nonparametric nature, allowing to determine the salient clusters even when their number is unknown apriori (as opposed to, for example, k-means). We use this property to estimate the distinct responses of each individual person and also the responses for groups of people by combining individual detection responses into groups of people where more than one person can occupy a cluster using a proposed metric.

In the next section we give an overview of DPMMs. In Section 3 we elaborate the details of our method including the choice of person detector, feature extraction stage and clustering scheme. We present the comparative evaluation results with three alternative methods at the end.

978-1-4799-4871-0/14/$31.00

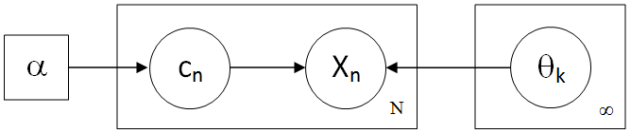


Figure 1: Graphical model for the DPMMs: the observation X_n depends on one of the infinite number of cluster parameters θ_k, assignment c_n of which is controlled by α

2. Dirichlet Process Mixture Models

DPMMs allow modeling the observations as a mixture model having unknown (as opposed to GMMs) number of mixture components [6], where $X_n; n = 1..N$ is the data that is to be modeled as a mixture of distributions having the form $F(\theta)$. For $X_n \in k$; $X_n \sim F(\theta_k)$ where θ_k denotes parameters of the kth mixture component. In our framework, each component corresponds to a cluster.

Imposing $X_n \in k$ with $c_n = k$, the discrete probability distribution $p(c_n = k)$ has Dirichlet distribution as conjugate prior. If the number of mixture components are taken to infinity, the distribution becomes the Dirichlet process. For a mixture model, DPMM assumes that infinite number of mixture components $k = 1..\infty$ exist, yet only a finite number of these components have observations assigned to them. Modeling the data with DPMMs consists of finding the parameters of those finite and unknown number of mixture components. To estimate these parameters, Markov chain sampling methods such as Gibbs sampling, which iterates over all observations and samples assignment to an existing or new cluster for each observation, can be done [8]. The sampling probability, which is controlled by a single parameter α where its higher values result in more clusters, is given as:

$$p(c_n; \alpha) = \begin{cases} \frac{N_k}{N+\alpha-1}\ p(X_n|\theta_k) & \text{existing } k \\ \frac{\alpha}{N+\alpha-1}\ \int_\theta p(X_n|\theta)\ d\theta & \text{new } k+1 \end{cases} \quad (1)$$

where N_k is the number of assignments to cluster k and N is the number of all observations. The graphical model for the DPMMs is depicted in Figure 1.

We prefer DPMMs over other clustering methods such as DBSCAN, which also does not require the number of clusters. Instead of requiring a similarity metric between feature vectors, DPMMs model the data such that the probability of cluster assignments are defined with a mixture model. We go for such a probabilistic mixture model for assignments of the detections to the clusters because of the nature of the detection process. For example, spatially, detections for a single person naturally group around the correct location of the person, and other visual features, e.g. color, depict a similar behaviour as varying around an average value. In addition, the overall performance of the clustering can be controlled by a single parameter (α). For instance, its higher values generate a larger number of clusters, which is preferable for more crowded scenes.

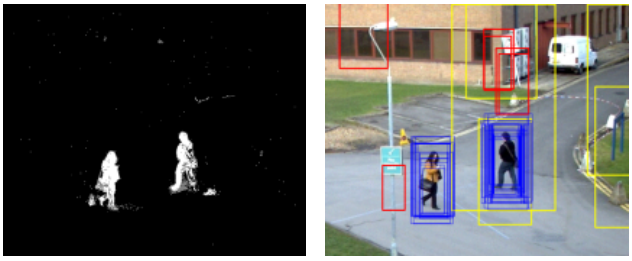
3. Counting People with DPMMs

3.1. Refining Person Detector Results

We apply a person detector at each frame. This detector is based on the histogram of oriented gradients (HOG), which first calculates gradients of the image in horizontal and vertical directions and accumulates histograms of gradient directions within small cells throughout the image [5]. For an image window, the concatenated normalized values of these histograms of cells constitute the HOG features. In training, HOG features are extracted for a large number of positive (person) and several order of magnitude more negative (non-person) images and a rejection cascade classifier is trained. During the detection process, a search window strides on the image and the classifier decides from the HOG features extracted for the position of the search window whether it contains a person or not. The size of the search window is repeatedly upscaled (or more commonly image is downsampled) at each iteration thus a pyramidal multi-scale search is performed.

We aggregate the person detection results over three consecutive frames to determine detection areas. To supplement the detection results, we compute two sets of optical flow maps [9], one between the previous frame and the current frame, and the other between the current frame and the following frame. Using these optical flow maps, we project the detection locations in the previous and the following frames to their estimated positions in the current frame. The shift vector for a detection area is taken as the average of the optical flow vectors of the keypoints that are covered by that detection area. Using detections from multiple frames improves compensation for the potential missed positives. We employ the extracted optical flow vectors and keypoints on the following steps too.

In addition, we calculate a foreground probability value of each pixel using a GMM based background representation [10] that models the previous color changes of each pixel using a mixture of Gaussians by applying an expectation maximization update. Note that, any change detection method can be used instead of GMMs. We filter out the detection results of the person detector, in case the pixels within a detection area have average foreground probability value less than a predefined threshold. We remove out the foreground areas that are larger than a predefined size. By applying these two simple heuristics, we aim to reduce the false positive detections. An example of the response of these two filters is presented in Figure 2b, where yellow bordered detections are filtered with the size threshold and



(a) (b)

Figure 2: (a) Foreground map.(b) HOG based person detection results where filtered out detections are depicted with yellow and red borders. Blue represents the final refined detection windows.

the red bordered detections are filtered with the foreground threshold using the foreground probability values of the pixels as shown in Figure 2a.

3.2. Observation and Cluster Models

After obtaining detection areas extracted from the frame itself and its immediate, we extract feature sets for each detection window and assign them as *observations* during the DPMM clustering stage. We model each observation using the spatial center of the detection area (i.e. x and y pixel coordinates) and the mean value of the a and b foreground pixel color components in the Lab color space. In addition to the local color and spatial information, we integrate temporal information about the movement of the people by employing an additional set of features derived from the optical flow maps between the neighbouring frames. We employ Histogram of oriented optical flows [11] (HOOF) where each optical flow vector contributes to a histogram bin corresponding to its orientation weighted with its magnitude. We represent histograms with four bins and compute four additional features for each detection area.

Similarly, we model each cluster using the mean and variance of the same color and spatial components as well as HOOF bins. For computational reasons, we do not model Gaussian models with full covariance matrices for clusters but only with the covariance coefficients between the spatial components. Thus, each observation is defined with 8 parameters; $X : (\mu_x, \mu_y, \mu_a, \mu_b, \mu_{h_{1..4}})$ and each clusters with 17 parameters; $\theta : (\mu_x, \mu_y, \Sigma_{xy}, \mu_a, \sigma_a, \mu_b, \sigma_b, \mu_{h_{1..4}}, \sigma_{h_{1..4}})$. Under this model, the likelihood that an observation X_n is generated by a cluster k with parameters θ_k is

$$p(X_n|\theta_k) = \mathcal{N}(X_{xy}|\mu^k_{xy}, \Sigma^k_{xy}) \prod \mathcal{N}(X|\mu^k, \sigma^k), \quad (2)$$

where the product on the right is calculated for color and HOOF features and the parameters of the Gaussians are estimated from the observations that are assigned to the clusters. Eq.s (1) and (2) together define the assignment probability of an observation to an existing or a new cluster.

For a new cluster, the integral in Eq. 1 is calculated over the whole prior distribution. The prior for Gaussian distribution is Normal-Inverse Wishart distribution and integrating over it gives a t-distribution [12]. However, [12] shows that this can be approximated by a Gaussian with properly chosen parameters. We choose it as a Gaussian that is centered on the frame and having a variance that covers the whole frame. The color and HOOF components have a similar coverage.

3.3. Clustering and Learning the α Parameter

Using the set of extracted observations and corresponding features for a frame, we perform iterative Gibbs sampling and sample assignments for each observation. We evaluate the observations one-by-one and calculate the association probabilities of observations to an existing or to a new cluster with Eq. (1).

We enforce the clustering process to generate a tractable number of clusters. We implement this enforcement to the clustering process implicitly by modifying the Gibbs sampling probability with another probability value with respect to the cluster size.

For each frame, we calculate the following statistics for sizes (width and height) of observations (i.e. detection areas of HOG detection): $\mu_w, \mu_h, \sigma_w, \sigma_h$. Using these per-frame statistics, we update the sampling probability $p_g(c_n = k)$ of assignment of observation n to cluster k as:

$$p_g(c_n = k) = p(c_n; \alpha) \;\; p(w_k|\mu_w, \sigma_w) \;\; p(h_k|\mu_h, \sigma_h), \quad (3)$$

where w_k and h_k are the width and height of cluster k if X_n is assigned to it.

The optimal value of α is related to the density of people in the scene and can vary through time. We learn its value by taking representative video frames as the training set and running the proposed algorithm with different α values to determine a correspondence between the number of optical flow keypoints and the error rate. During testing, we assign the best α value from the number of keypoints and the learned correspondence.

3.4. Inferring the Number of People from Clusters

The ideal outcome of the clustering process described in the previous section is that every person on the scene being represented with one distinct cluster, thus the number of clusters being equal to the number of people in the scene. In practice, this may not always be achieved and people which appear close to each other in the scene and having similar appearance features may be clustered into a single cluster, so taking the cluster count itself may be misleading. In addition to controlling the cluster size with the update presented

Algorithm 1 Cluster detections $X_n; n = 1..N$ with α and infer the number of people C from clusters

$H = \{\theta_0\}$
Calculate $\mu_w, \mu_h, \sigma_w, \sigma_h$
for # of Gibbs iterations **do**
  for $n = 1$ **to** N **do**
    for $t = 1$ **to** $\mid H \mid$ **do**
      if $X_n \in \theta_t$ **then**
        $X_n \notin \theta_t$, update θ_t
      end if
      $p_g(c_n = t) \leftarrow p(c_n = t; \alpha)\ p(w_k|\mu_w, \sigma_w)\ p(h_k|\mu_h, \sigma_h)$
    end for
    Sample $t \propto p_g(c_n = t)$
    if $t = 0$ **then**
      Init θ_{t+1} with X_n , $H \leftarrow \{H, \theta_{t+1}\}$
    else
      $X_n \in \theta_t$, update θ_t
    end if
  end for
end for
$C \leftarrow 0$
for $t = 1$ **to** $\mid H \mid$ **do**
  $C \leftarrow C + N_t$
end for

in Eq. (3), we present an additional measurement to infer the number of people in a cluster.

Following [2] and [3], we also propose a keypoint based measure since the keypoints for optical flow are already available. In practice, a person is usually covered by many overlapping detections and the number of keypoints within those overlapping detections do not change heavily–as well as the cluster they constitute if they cover the same person. On the other hand, if the detections that constitute a cluster are related to different people, the number of total keypoints in the cluster will be much more than the number of keypoints within the separate detection areas, since the union of keypoints come from different detection sources.

On top of these assumptions we propose to use the following measure to estimate the number of people N_k in a cluster k:

$$N_k = \left[\frac{p_k}{\overline{p}_{n \in k}} \right], \tag{4}$$

where p_k is the total number of keypoints in cluster k and $\overline{p}_{n \in k}$ is the average number of keypoints in the detection areas that constitute the cluster k. In summary, the overall clustering algorithm that works on filtered HOG detections is depicted in Algorithm 1.

4. Experiments and Results

We present our experiments on PETS 2009 [13], Peds2 [14] [1] and BEHAVE [15] datasets. To extract the foreground pixels, we applied the GMM implementation; for person detection, the raw output of the HOG implementation and to extract optical flows, the optical flow implementation of EmguCV library. We did not train specific HOG models for the video sequences and used a generic HOG model [16] trained on completely separate set of videos and shipped with EmguCV; by giving manual HOG parameters, like the upscaling of the video frames or classifier thresholds, for each dataset. As in [2] and [3] we apply a final low-pass filter to the number of people to smooth out the number of countings.

[1] Since Peds2 is grayscale, we used gray values as color features.

4.1. Visual Results

We present some sample scenes in Figures 3, 4 and 5 for PETS 2009, BEHAVE and Peds2 datasets respectively. The scenes depict examples of cases where detections for each person are clustered into a single cluster successfully because of the actual spatial distance (e.g.Figure 3h) or color variations (e.g. Figure 4f) between the people being high enough. There are also cases where a cluster contains more than one person, because of very high overlap between detections (e.g. Figure 3n) or nearby detections with similar color values (e.g.Figure 4g), and the proposed measure in Eq. (4) can successfully estimate the number of people in the cluster in such cases. Figures 3l and 3m depict a case when two people begin to be clustered as one while coming closer, because of having similar colors and the number of detections for one (on the lower) being much higher than the other–since the assignment probability for a cluster increases with the number of detections assigned to it (by the N in the numerator in Eg. (1)). Even so, the number of people is again inferred successfully with Eq. (4) in Figure 3m.

4.2. Quantitative Results

In Table 1, we compare the error values of the people count estimations of the proposed method with the results of [1] and [3], as well as results obtained by applying mean-shift clustering to the extracted detections and features for the two of *View 1* video sequences of the PETS 2009 dataset. In Table 2, we compare error values of the proposed clustering method with the results obtained by applying the mean-shift clustering for the first sequence of Peds2 dataset. We report two error values; *mean absolute error (MAE)* which is the average value of the absolute error per frame and *mean relative error (MRE)* which is the average value of the ratio of the absolute error to the ground truth value per frame, i.e.:

$$MAE = \frac{1}{F} \sum_{f=1}^{F} |G_f - C_f| \ , \ MRE = \frac{1}{F} \sum_{f=1}^{F} \frac{|G_f - C_f|}{G_f}, \tag{5}$$

where F is the total number of frames in the sequence, G_f is the ground truth count for frame f and C_f is the counting result obtained by the relevant method for frame f.

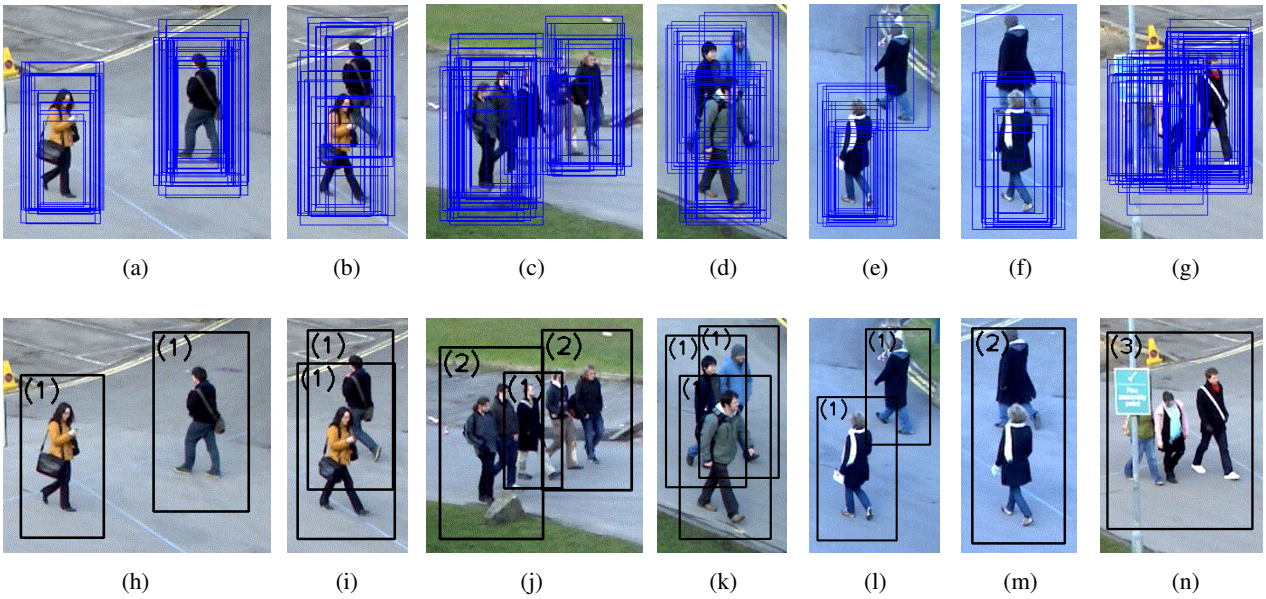


Figure 3: HOG detections (first row) and clusters with estimated number of people (second row) for PETS2009 dataset

Figure 4: HOG detections (first row) and clustering results with number of people (second row) for BEHAVE dataset

4.3. Running Time

On a PC with 2.50 GHz dual-core CPU, extracting detections and features for DPMM clustering took $\sim$ 6 sec.s per frame in average, where most tasks (i.e. foreground extraction, HOG detection and optical flow calculation) implicitly benefited from CPU parallelization–thanks to EmguCV's multi-threaded nature. DPMM clustering with Gibbs sampling and the rest of the steps took $\sim$ 1 sec. per frame in average, with no special parallelization employed.

	S1.L1.13-57	**S1.L1.13-59**
Subburaman, et al. [1]	5.95 (30.00%)	2.08 (11.00%)
Conte, et al. [3]	1.92 (8.70%)	2.24 (17.30%)
Mean-Shift Clustering	3.05 (13.57%)	4.20 (29.92%)
Proposed Method	1.47 (7.35%)	1.50 (10.74%)

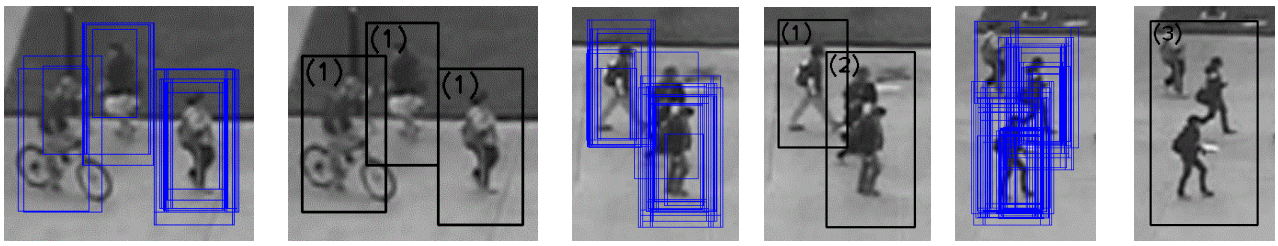
Table 1: MAE and MRE values for PETS 2009 [13].

	Mean-Shift Clust.	**Proposed Method**
UCSD Peds2	4.58 (16.83%)	1.30 (4.90%)

Table 2: MAE and MRE values for UCSD Peds2 [14].

5. Conclusions and Future Work

We present a people counting system by applying DPMM clustering on person detector outputs using different features like color, spatial and temporal. The proposed algorithm benefits from the nonparametric nature of the DPMMs to handle unknown number of clusters. In our work we used HOG detectors, however the proposed algorithm is neutral to the detector being used and can be applied to any person detector which generate similar outputs. While inferring the number of people in a cluster, only the neighbourhood of the cluster is taken into consideration, since the overgrowth of clusters is prevented with Eq. (3). Thus the proposed measure in Eq. (4) is perspective invariant and different than [2] which assumes an overall ratio for the whole scene. The advantage of our method over [3] is that we do not need to train regressors and instead employ Eq. (4) to



(a) (b) (c) (d) (e) (f)

Figure 5: HOG detections (a, c, e) and clustering results with estimated number of people (b, d, f) for UCSD Peds2 dataset

infer the number of people in a cluster. Only the number of people in a few number of frames is required, which is used while learning the α value.

The success of the overall algorithm relies on the success of the baseline detector. The proposed method is suitable for sparsely or moderately crowded scenes. In overcrowded scenes, HOG detector may fail to distinguish targets and long term target occlusions occur. More competent detectors, e.g. deformable part models, can be employed to improve detector accuracy and long term tracking methods, e.g. belief nets, can be incorporated to handle persistent occlusions. DPMM clustering step can be time-optimized by parallelization as shown in [17] where the cluster assignment probabilities are calculated in parallel, which results reported clustering tasks on running 4 times faster with 8 processors.

References

[1] V. Subburaman, A. Descamps, and C. Carincotte, "Counting people in the crowd using a generic head detector," in *IEEE Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pp. 470–475, 2012.

[2] A. Albiol, M. J. Silla, A. Albiol, and J. M. Mossi, "Video analysis using corner motion statistics," *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, pp. 31–38, 2009.

[3] D. Conte, P. Foggia, G. Percannella, F. Tufano, and M. Vento, "A method for counting moving people in video surveillance videos," *EURASIP Journal on Advances in Signal Processing*, no. 1, 2010.

[4] G. Antonini and J. P. Thiran, "Counting pedestrians in video sequences using trajectory clustering," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 16, pp. 1008–1020, Aug. 2006.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 886–893, 2005.

[6] C. E. Antoniak *The Annals of Statistics*, vol. 2, no. 6, pp. 1152–1174, 1974.

[7] W. Neiswanger, F. Wood, and E. P. Xing, "The dependent dirichlet process mixture of objects for detection-free tracking and object modeling," in *AISTATS*, pp. 660–668, 2014.

[8] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *Journal Of Computational And Graphical Statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[9] J. J. Gibson, *The Perception of the Visual World*. Boston, MA: Houghton Mifflin, 1950.

[10] C. Stauffer and E. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2002.

[11] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1932–1939, 2009.

[12] E. Sudderth, *Graphical Models for Visual Object Recognition and Tracking*. PhD thesis, Massachusetts Institute of Technology, 2006.

[13] PETS2009, "Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance." ftp://ftp.pets.rdg.ac.uk/pub/PETS2009/.

[14] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1975–1981, 2010.

[15] S. Blunsden and R. B. Fisher, "The behave video dataset: ground truthed video for multi-person behavior classification," *Annals of the BMVA*, vol. 2010, pp. 1–11, Jan. 2010.

[16] M. Enzweiler and D. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2179–2195, 2009.

[17] S. Williamson, A. Dubey, and E. P. Xing, "Parallel markov chain monte carlo for nonparametric mixture models.," in *International Conference on Machine Learning (ICML)*, vol. 28, pp. 98–106, 2013.