# Evaluation of Sampling-based Pedestrian Detection for Crowd Counting*

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

## Abstract

*With increases in computing power, the once computationally expensive sampling-based methods have been successfully applied to solve many hard vision problems of combinatorial nature, such as image segmentation, video tracking, and object detection. In this paper, we perform pedestrian detection using the reversible jump Markov Chain Monte Carlo (RJMCMC) sampling method. A crowd scene is viewed as a realization of a Marked Point Process (MPP) that consists of a random set of people in a bounded region. Each person is associated with a random 'mark' that governs their location and size in the image. To automatically infer the number of people in the scene and their spatial locations, RJMCMC is used to sample person hypotheses from an underlying stochastic process and evaluate them against the image observation to find the optimal configuration that best explains the image. We further extend the detector to hypothesize people not in the image plane but in 3D space, by incorporating multi-view information. The detection performance of both the single- and multi-view versions are evaluated on the crowd counting task in the PETS 2009 dataset.*

## 1. Introduction

We consider people in a crowd scene as a realization of a Marked Point Process (MPP). An MPP is a stochastic process that consists of a random set of points in a bounded region. People detection is solved by finding the configuration of a point set describing the number of people and their attributes (location and size) that best explains the image observation. The solution space is of a combinatorial nature, since for each pixel location there is a possibility for the presence or absence of a person with a certain size. We resort to the reversible jump Markov Chain Monte Carlo (RJMCMC) method to solve this NP-hard problem. RJMCMC is used as a stochastic optimization scheme to search the solution space efficiently and to avoid the traps of local minima. The method not only detects people in the image plane, it is also a general framework that can be easily modified to detect people in 3D space. We will present the implementation of both versions and the evaluation results on the PETS benchmark data for the crowd counting task to demonstrate its robustness to challenging scenarios (Figure 1).

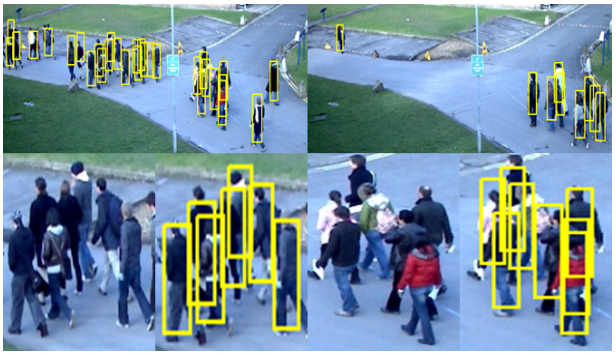

Figure 1: Sample detection results on the PETS data with different crowd densities (top). The sequence is very challenging, with frequent heavy occlusions (bottom). The detector not only generates an accurate person count but also achieves high localization accuracy for individuals.

## 2. Related Work

Crowd counting and segmentation have been approached by many different methods, such as texture or motion analysis [9, 10], and machine learning techniques that classify observation features into human or background [2]. Our work is closely related to Bayesian graphical models that have been used for object detection and recognition [13]. Taking a probabilistic approach allows us to achieve global optimization by considering contributions from all the people in the scene jointly, as opposed to local decision making at individual locations in a sliding-window or grid-based classifier [14].

Marked Point Processes have a close connection with Markov Random Fields (MRFs). The MPP framework is more general than MRFs for modeling a varying number

*This work was partially funded by the NSF under grant IIS-0729363.

of randomly located and possibly interacting objects [3]. It overcomes one major limitation of the pixel-wise MRF model [13] that only local constraints between neighboring pixels can be encoded, which makes it difficult to model long-range geometric constraints such as penalties for two overlapping rectangles. A possible remedy is to build the MRF upon high-level primitives so that instead of a pixel being a node in the graph, an image patch can serve as the base unit. However, construction of MRFs requires knowledge about the number of nodes and their relationship beforehand. Hence its applications are limited in scenarios like pedestrian detection where people are constantly moving in and out of the scene. One recent extension applicable to video is dynamic MRFs [7].

MPP models have been used to detect objects in images [1, 3, 4, 8, 11, 12, 16, 17]. Applications were often confined to pedagogical, toy "object recognition" examples where simple shapes such as disks or polygons are recognized from noisy synthetic images, due to difficulties in inference. With increases in computing power, the models began to appear in real-world applications, such as segmenting cells in confocal microscopy images [11, 12], detecting buildings from Digital Elevation Models [8], and finding people in crowds [4, 16].

This paper is organized as follows. We first review the general formulation of the pedestrian detection MPP framework in Section 3. Section 4 describes the RJMCMC procedure that searches for the optimal solution in the image plane and an extension of the detector to 3D space using multiple camera views. Section 5 presents experimental results on the PETS benchmark data for the crowd counting task. Finally, we conclude the paper in Section 6.

## 3. Problem Formulation

The task of pedestrian detection is illustrated in Figure 2. The algorithm takes a binary foreground mask as input, generated by a two-pass background subtraction based on adaptive gaussian mixture models (details in Section 5.1). Following previous authors [4, 16], the goal is to place some rectangle set that fits the foreground mask best, that is, the set covers the most foreground pixels and the least background pixels. To find such a rectangular covering, we need to determine the number of rectangles and their spatial placement, which we refer to as a *configuration*. Spatial point processes are suitable for modeling the spatial distribution of an *unknown* number of objects. An MPP couples a spatial point process $Z$ with a second process defined over a "mark" space $M$ such that each point $p \in Z$ is associated with a random mark $m_p \in M$. Elements of our MPP consist of an image location $p$ defined on a bounded subset of $R^2$, together with a mark $m = (w, h)$ defining the width and height of the rectangle to be placed at point $p$. More

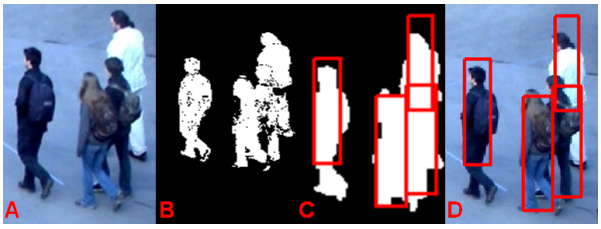


Figure 2: **A**: original image. **B**: binary foreground mask obtained by adaptive background subtraction. After morphological postprocessing, it becomes the input to the detector. **C**: the detector finds the best configuration of a rectangle set that covers the most foreground pixels and the least background pixels by RJMCMC. **D**: the best rectangular cover overlaid on the original image.

formally, the MPP can be written as

$$\pi(\mathbf{o}) = \prod \pi(\mathbf{o}_i) = \prod \pi(p_i)\pi(w_i, h_i|p_i). \qquad (1)$$

The factorized form comes from the assumption that the placement of each individual is independent from others. The second term is a conditional mark process that models the size of a 2D detection box, conditioned on its spatial location. It encodes the relationship between the size of a projected human body at different image locations, which we can estimate beforehand for static cameras by computing the scene geometry: $\pi(w, h) = \mathcal{N}_w(\mu_w, \sigma_w) * \mathcal{N}_h(\mu_h, \sigma_h)$, where $\mathcal{N}_w$ and $\mathcal{N}_h$ are normal distributions. The mean parameters are estimated by assuming an average person size and using the provided camera calibration data to compute the width and height of the projected person at each pixel.

Under the MPP framework, we assume the foreground mask is generated by this underlying stochastic process. Finding the best configuration can be done through repeatedly drawing a sample from the MPP and comparing it against the observed foreground mask. First, a synthetic 'label' image is computed from the sampled configuration by setting pixels that are covered by at least one rectangle in the set as foreground pixels. Since both the 'label' image and the foreground image are binary masks, comparing the two can be done using Bernoulli tests. Denote $Y = \{y_i\}$ as the foreground mask and $X = \{x_i\}$ the label image, $x, y \in \{0, 1\}$. At each image location $p_i$, $f(p_i) = q^{I(\cdot)}(1-q)^{1-I(\cdot)}$, where $I(\cdot)$ is an indicator function that equals 1 if $x_i == y_i$ and 0 otherwise. Assuming independence among pixels, the likelihood function is defined as follows:

$$\log \mathcal{L}(Y|X) = \log \prod_{i=1}^{N} f(p_i) = \log \prod_{i=1}^{N} q^{I(\cdot)}(1-q)^{1-I(\cdot)}$$
$$= N_0 \log q + \tilde{N}_0 \log(1-q) \qquad (2)$$

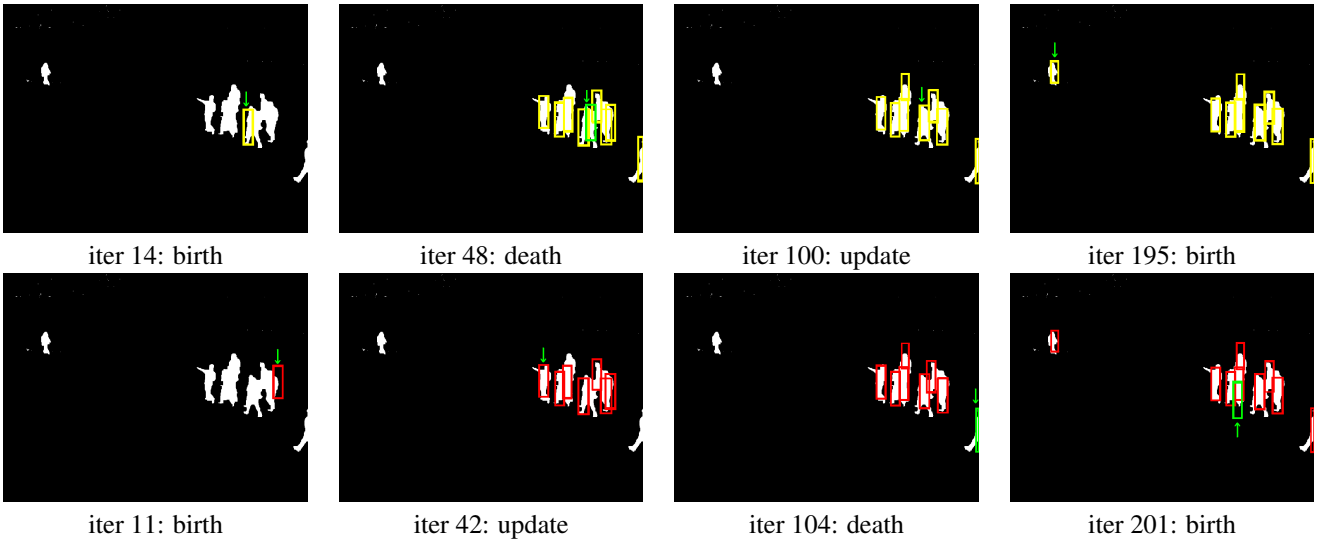


Figure 3: Intermediate results during the RJMCMC iterations in the image plane. **Top**: accepted proposals. **Bottom**: rejected proposals. In a birth or update move, the green arrow indicates the rectangle being added or modified that makes the current configuration $o^*$ differ from the configuration $o^t$ at the previous iteration. In a death move, the green box indicates the rectangle that is being removed from $o^t$, and the remaining rectangles constitute $o^*$.

where $N_0$ is the total number of pixels that have the same value in the foreground mask and the label image and $\tilde{N}_0 = N - N_0$ is the number of pixels whose value differ in the two masks. With parameter $q \in (0.5, 1)$, the likelihood function favors as many successes (agreement between foreground and label image) as possible in a series of independent Bernoulli tests at each image location. The best configuration is thus the one with the highest likelihood score. However, the search space for all possible configurations is huge and may have many local minima, especially when there is an unknown number of people in the scene. We resort to reversible jump Markov Chain Monte Carlo (RJMCMC) to search for the global optimal configuration efficiently.

# 4. Configuration Search by RJMCMC

The Markov Chain Monte Carlo method is a class of algorithm originally designed to generate samples from complicated distributions by constructing a Markov Chain with the target distribution $\pi(\theta)$ as its equilibrium distribution [5]. The Metropolis-Hastings algorithm is one popular method to simulate the desired Markov Chain. Starting with some initial state, the algorithm proposes a new state $\theta^*$ from a proposal distribution $Q(\theta^*|\theta)$ that depends on the current state $\theta$. The proposed state is accepted as the next state in the Markov Chain with acceptance probability

$$\alpha(\theta^t, \theta^*) = \min(1, \frac{\pi(\theta^*)}{\pi(\theta)} \frac{Q(\theta|\theta^*)}{Q(\theta^*|\theta)}). \quad (3)$$

RJMCMC [6] is an extension of the basic Metropolis-Hastings algorithm that can explore configurations of different dimensionality, in our case, different numbers of people. The proposal function is the key to efficient search of the solution space. We adopt a data-driven approach and design highly informative proposals based on the observed foreground mask. Hence, the RJMCMC iterations can be viewed as a stochastic hypothesis and testing procedure [17].

To generate configuration hypotheses, we have designed three simple proposals: birth, death and update. Among the three, birth and death proposals are a pair of reversible jump moves because adding or deleting a rectangle from the current configuration causes a dimension change. Denote $o^t = \{o_1^t, \ldots, o_m^t\}$ as the current state at iteration $t$ that consists of $m$ rectangles at different locations and of possibly different sizes, parameterized by $(p_i, w_i, h_i)$. A new configuration $o^*$ is proposed by first choosing one move from all three candidates according to a move probability distribution that is essentially a uniform distribution, but adapted to the current configuration for better efficiency. For example, if the current person count is zero ($m = 0$), only the birth move is allowed. We now describe each proposal in detail.

**Birth/Death proposal.** During each birth move, a point and mark are added to the current configuration by sampling a centroid location according to the foreground mask, which makes it a data-driven proposal. The width and height are Gaussian distributions sampled from the conditional mark process indexed by image locations. The proposed config-

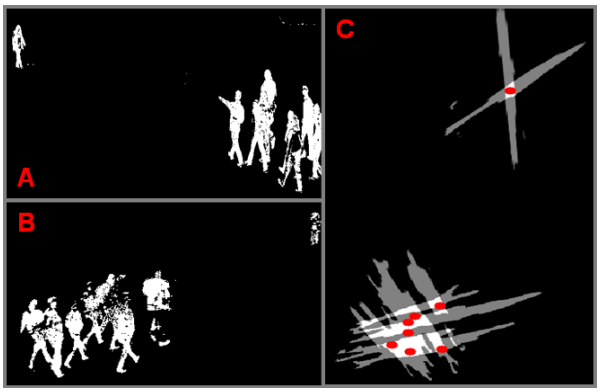


Figure 4: Foreground masks from View1 (**A**) and View2 (**B**) are projected to the centroid-plane to create a fused proposal map (**C**). Red dots are proposed locations in one iteration of RJMCMC. Locations in the brighter areas are more likely to give rise to new human hypotheses.

uration hypothesis is $o^* = \{o^t, o_{m+1}\}$. The reverse move of birth is death. The death proposal chooses one rectangle uniformly at random (u.a.r.) from the current rectangles and removes it from the configuration. The dimension of the proposed hypothesis configuration increases by one for a birth and decreases by one for a death.

**Update proposal.** The update proposal keeps the dimension of the current configuration untouched but generates a new hypothesis by making local adjustments to an existing detection. Specifically, a rectangle is chosen u.a.r. from $o^t$ and either its location or mark parameters are modified. Location update is done as a random walk around the current center. Modification of the mark is done by sampling from the corresponding conditional mark process associated with the current location. The update proposal is its own reverse move.

Each newly proposed configuration is accepted with probability $\alpha$ as defined in Eqn. 3, where $\pi(o^*) \propto \log \mathcal{L}(Y|X^*)$. Here again, $Y$ is the foreground mask and $X^*$ is the synthetic label image generated from the proposed configuration $o^*$. Notice that the proposed hypothesis $o^*$ only differs from the current state $o^t$ by one rectangle $r$. Although this limits the algorithm to explore the search space locally between successive steps, it greatly simplifies the computation of the acceptance probability. Because the label image $X^*$ only differs from the previous label image $X^t$ at the region covered by $r$, only those terms associated with pixels within that region will contribute to the computation of the log-likelihood ratio (Eqn. 3). Other terms are the same for both the numerator and the denominator so that they cancel out. Figure 3 illustrates sample birth, death, and update proposals at different iterations during one round of RJMCMC.

**Multi-view extension.** Our stochastic approach provides a uniform framework for detection using single-view and multi-view information. We now describe a straightforward extension from the aforementioned proposals to incorporate multi-view observations. We only use View1 and View2 from the sequence for illustration. One does not benefit much from the other two views as people are rather small in those images and not well-separated.

Firstly, the foreground masks from both views are projected to the centroid-plane, the plane that is half of an average-person-height above the ground plane. We did not choose the ground plane as the reference plane because we found feet do not get detected easily from background subtraction and even when they came out nicely in the foreground mask, they are usually separated when people are walking. To create the fused proposal map, we first use backward bilinear interpolation to warp the individual foreground masks into the centroid plane and average them together. During the birth proposal, we hypothesize person locations in the centroid-plane based on the fused foreground mask (Figure 4). Similarly, the location update also happens in the centroid-plane. To compute the acceptance probability, we backproject the hypothesized person to individual views and compute the log-likehood ratio as the summation of ratios from two image views.

# 5. Experimental Results

We evaluate our method on person counting task from the PETS 2009 dataset. Before we present our evaluation results, we first briefly describe how we generate the input foreground mask by a two-pass adaptive background subtraction.

## 5.1 Background Subtraction

Adaptive background subtraction is a well-studied area. We use a standard approach that maintains a mixture of Gaussians color model at each pixel [15], based on publicly available code due to Zivkovic [18]. However, it is well known that this approach does not handle sudden global illumination changes well, such as the sun coming out from behind clouds. For a pre-recorded sequence such as the PETS dataset, we use a forward-backward approach to compute foreground masks. Adaptive background subtraction is run forward in time from the first frame to last of the sequence, and a second process runs backward in time from the last frame to first. Assume a rapid change in illumination occurs at time T, with gradual illumination changes before and after that. The forward background subtraction pass will produce clean foreground masks for times less than T and then suffer degraded performance at time T before gradually recovering. Likewise, the backward pass will produce clean foreground masks for times greater than T, but suffer

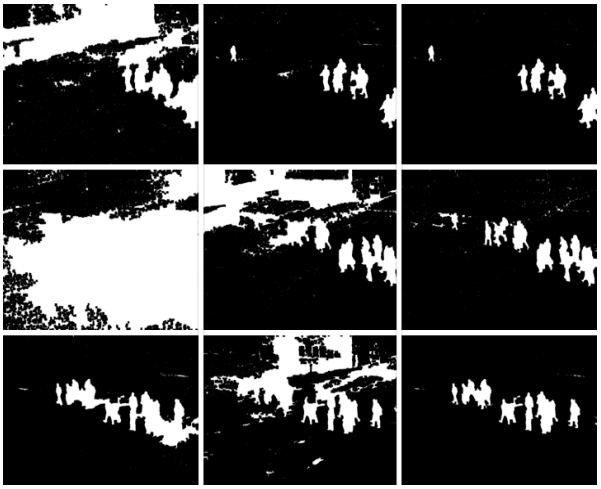

Figure 5: Example showing that forward-backward processing produces improved foreground masks during a sudden, global change in scene illumination. Row 1: frame 20, near beginning of an illumination change. Row 2: frame 40, near middle of the change. Row 3: frame 60, near end of the global illumination change. Column 1: foreground mask produced by running adaptive background subtraction forward in time. Column 2: foreground mask produced by running background subtraction backwards in time. Column 3: Results produced by combining both forward and backward masks.

degraded results for a short period of time before T. Combining both the foreground and background masks with an AND operator tends to produce greatly improved masks both before, during and after the illumination change (see Figure 5). Although it may seem at first glance that this strategy is limited to batch processing, if a one to two second delay is acceptable then forward-backward processing methods can be performed on real-time camera streams by using a sliding temporal window approach [19].

## 5.2 Quantitative Evaluation

We evaluate the single- and multi-view detectors on the PETS crowd counting task. Person counts for the S1.L1.13-57 sequence are shown in Figure 6. The detected count captures the trend of count change very well and is quite close to the ground truth count for individual frames. The multi-view version performed worse in this case. Similar observations were reported in a previous paper [14], partly because the camera setting does not provide a full, wide-angle coverage like in a smart room, and because all people are densely clustered together within the field of view. Furthermore, our current implementation of multi-view proposals is very primitive. Because we propose detections in the centroid-

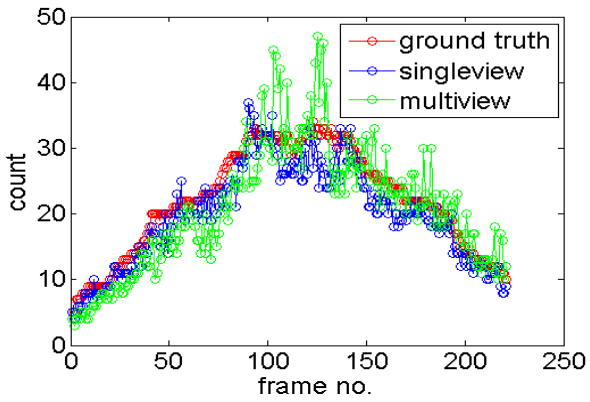


Figure 6: Person counts in the test region R0 for individual frames from PETS sequence S1.L1 using both single- and multi-view.

plane, a small error in the centroid estimate will cause big displacement in the image plane during backprojection, and together with noisy foreground masks, causes imprecision in the localization results as compared to the single-view detector. We plan to develop more sophisticated schemes

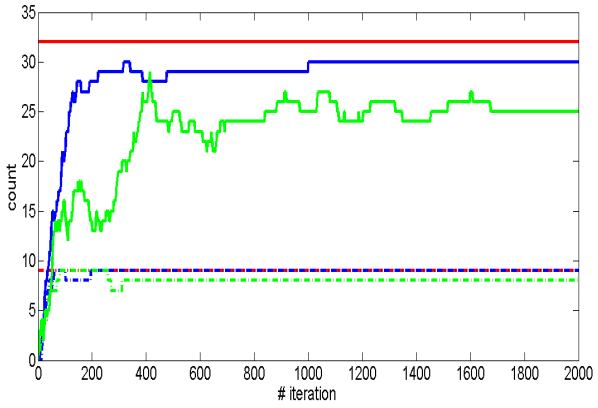


Figure 7: Empirical convergence analysis for RJMCMC on two frames of 9 (dashed line) and 32 (solid line) people. The ground truth, single-view, and multi-view counts are colored in red, blue, and green respectively. The person count estimates converge quickly.

that take better advantage of multi-view constraints such as proposing update moves along the epipolar line. We caution that the overall count in a test region is a very crude performance measure as an algorithm that overestimates the count in one part of the region and underestimates in another can still get high overall accuracy. Hence we also show sample localization results in Figure 8.

We also conducted empirical convergence analysis of our

sampling procedure. As shown in Figure 7, the estimated person count converges quickly, usually under 1000 iterations, especially for low density crowds. In general, dense crowds need more iterations. As expected, multi-view takes a longer time to converge than single-view because the best decision has to be made by balancing observations from multiple information sources.

# 6. Conclusions

We have presented a pedestrian detection method based on inferencing from a Marked Point Process that optimizes estimates of the number of people and their locations jointly, and have extended it to work for multi-view data. The sampling-based approach provides a uniform framework for detection within single and multiple views. Quantitative evaluation on the PETS crowd counting sequence demonstrates it is a viable method for handling the challenges of person detection under occlusion and for fusing multi-sensor information. It produces relatively accurate person counts and localization, and is robust against error in background subtraction. We plan to test the method on other PETS tasks and to further investigate alternative 3D proposals that exploit the multi-view geometry constraint more effectively.

# References


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

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

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

[4] W. Ge and R. T. Collins. Marked point processes for crowd counting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

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

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

[7] P. Kohli and P. Torr. Dynamic graph cuts for efficient inference in markov random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29:2079–2088, 12 2007.

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

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

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

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

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

[13] Y. Sheikh and M. Shah. Bayesian object detection in dynamic scenes. In *IEEE Computer Vision and Pattern Recognition*, pages 74–79, 2005.

[14] S. Stalder, H. Grabner, and L. V. Gool. Exploring context to learn scene specific object detectors. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 63–70, June 2009.

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

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

[17] S. Zhu, R. Zhang, and Z. Tu. Integrating bottom-up/top-down for object recognition by data driven markov chain monte carlo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 738–745, 2000.

[18] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *International Conference on Pattern Recognition*, volume 2, pages 28–31, 2004.

[19] Z.Yin and R.Collins. Moving object localization in thermal imagery by forward-backward mhi. In *IEEE Workshop on Object Tracking and Classification in and Beyond the Visible Spectrum*, 2006.

Figure 8: Sample detection results on PETS data in both image plane(left) and 3D space (right) using View1 and View2. Detections are overlaid with the original frame and the foreground mask. In View1, detections in the three test regions R0, R1 and R2 are colored in yellow, red and green respectively.