MVA2015 IAPR International Conference on Machine Vision Applications, May 18-22, 2015, Tokyo, JAPAN

Object Detection in Surveillance Video from Dense Trajectories

Mengyao Zhai, Lei Chen, Jinling Li, Mehran Khodabandeh, Greg Mori
Simon Fraser University
Burnaby, BC, Canada
mzhai@sfu.ca, chenleic@sfu.ca, Jinlingl@sfu.ca, mkhodaba@sfu.ca, mori@cs.sfu.ca

Abstract

Detecting objects such as humans or vehicles is a central problem in video surveillance. Myriad standard approaches exist for this problem. At their core, approaches consider either the appearance of people, patterns of their motion, or differences from the background. In this paper we build on dense trajectories, a state-of-the-art approach for describing spatio-temporal patterns in video sequences. We demonstrate an application of dense trajectories to object detection in surveillance video, showing that they can be used to both regress estimates of object locations and accurately classify objects.

1 Introduction

Object detection is a crucial first step in numerous visual surveillance applications. High-level human activity analysis typically builds upon this step. As such, robust solutions for detecting and classifying objects in videos is a well-studied problem. Standard approaches to the problem include background subtraction, moving point trajectory analysis, and appearance-based methods.

All of these methods have long histories in the computer vision literature. Appearance-based methods are exemplified by the histogram of oriented gradients (HOG) detector [1] and its variants. Background subtraction-based methods (e.g. [2]) find contiguous foreground regions and further classify them into object types. Point trajectory [3] or moving region [4] methods are related, finding groups of points moving together or containing a motion different from the background.

Each of these methods has its shortcomings. Appearance-based methods are sensitive to highly textured background regions and need to handle substantial intra-class variation. Background subtraction methods and moving point/region methods are sensitive to moving clutter regions. Further, there is often ambiguity when distinguishing foreground from background due to pixel-wise similarities between objects and the background.

In this paper we present an application of the recently developed dense trajectories approach [5] for combining both moving trajectories and discriminative (moving) appearance-based classification for object detection in surveillance video. The dense trajectories approach has been shown to obtain state-of-the-art performance on standard benchmarks for human activity recognition, particularly for unconstrained internet videos. Here we demonstrate their effectiveness for the tasks of human and vehicle detection in surveillance videos. We focus on detecting only moving objects – note that while stationary objects can be of interest, in a surveillance context objects generally move at some point, and a tracker or scene entry/exit point knowledge can be used to fill temporal gaps.

The contribution of this paper is the application of these descriptors to the problem of object detection in surveillance video. Our method utilizes dense trajectories in two steps. First, we detect moving regions and estimate object locations by regressing from dense trajectory descriptors to object locations. After forming candidate detections, we then score them by training classifiers upon a dense trajectory bag-of-words representation. We demonstrate empirically that this method can be effective for human and vehicle detection in surveillance video.

2 Previous Work

As noted above, object detection in surveillance video is a well-studied problem. Turaga et al. [6] provide a survey of this literature. Classic approaches mentioned above include appearance histogram-based methods [1]. More recent methods based on refined Haar-like features [7] and deep learning [8] have shown impressive results for single image person detection.

In this paper we focus on surveillance video. Analyzing the temporal domain should lead to more robust detection algorithms for this static camera setting. Classic methods such as grouping moving feature points exist [3]. Similar methods have been explored in the context of crowded videos of people [9]. The aforementioned appearance-based methods have been extended to video, e.g. [10, 11]. However, relatively fewer recent methods that focus on motion patterns exist. The main focus of this paper is revisiting the idea of trajectory-based object detection and classification based on state-of-the-art trajectory descriptors.

3 Motion-based Detection Model

A high-level overview of our method is shown in Fig. 1. First, we follow the dense trajectory pipeline of finding and tracking moving points over short time scales. From these we use regression to predict the positions of objects. We then agglomerate nearby predictions by clustering and generate bounding boxes. Finally classifiers are trained to discriminate objects of interest from other regions. In the following subsections we provide details of each of these steps.

3.1 Trajctory-aligned Motion Features

We develop an algorithm for detecting moving objects in surveillance video. The first step of our algorithm is to generate a set of candidate object locations. Analagous to Hough transform-type voting methods (e.g.. [12]) we will do this by generating an initial set of points that can vote for possible object centers via a regression step.

A key consideration is that we would like a large number of such points. In constrast, approaches such

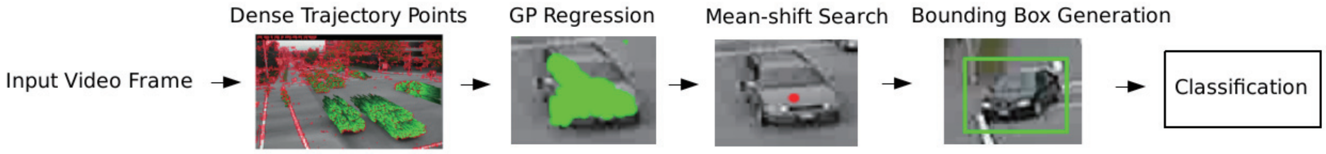


Figure 1. Overview of detection procedure. Our method first computes dense trajectories from an input video. Each dense trajectory votes for an object center via a learned regression model. These votes are aggregated using mean-shift clustering. Finally, bounding boxes are generated from the clusters and then scored using a classifier on dense trajectory features.

as Harris interest points are sparse and will have difficulty covering objects, especially given noise in the subsequent regression step. Our approach is to extract dense moving points from a given video, and use these points to vote for possible object centers. We use the dense trajectory algorithm from Wang et al. [5]. In that algorithm, dense trajectories are obtained by tracking densely sampled image points through multiple frames using optical flow, HOG descriptors are extracted around each point along a trajectory and normalized to produce a feature vector for corresponding trajectories. Features extracted from a trajectory are shown to be more robust compared to features extracted from single points.

We use these densely sampled trajectory points and their descriptors as inputs to regress object positions, as described next.

3.2 Center Prediction Based on Regression

In this section, we describe how possible object center locations are predicted using a regression model and how to detect objects given these predicted centers.

Let the number of trajectories be N and the length of each trajectory be L. Our goal is to predict the locations of object centers, given a feature vector s_i extracted along a trajectory where $\{i = 1, 2, ..., N\}$. A regression model is learned and the output of the regression model is the offset vector $o_i = (x_i, y_i)$ starting from points $p_{ij} = (x_{ij}, y_{ij})$ on a trajectory pointing to possible centers $c_{ij} = (u_{ij}, v_{ij})$ where $\{j = 1, 2, ..., L\}$. We assume all points on a trajectory should share exactly the same offset vector. Thus, the inputs of the regression model are features s_i and the outputs are o_i. The center prediction can be computed as:

$$c_{ij} = p_{ij} + o_i \tag{1}$$

Many objects have internal symmetries. For example, a car has two front lights. This requires the model to have the capability that given one feature vector the output should not be only one offset vector but several possible offset vectors. Suppose we want to produce M outputs given one input. Intuitively we can achieve this goal by clustering the input feature space into M subspaces, train a regression model for each subspace, and finally get M outputs given one input. The problem of doing so is that we cannot capture the relations between pairs of outputs. Sometimes more accurate results can be generated if the relations between outputs are also modelled. To achieve this goal, we use the dependent gaussian process model [13] where the relations between the outputs are modelled by adding a noise source which influences all outputs.

Suppose we want to produce M offset outputs $Y_k(s)$ where $s \in R^p$ is a dense trajectory feature and $k = \{1, 2, ..., M\}$, and for each output we have N_k training observations. Given M datasets $D_k = \{s_{ki}, o_{ki}\}_{i=1}^{M_k}$, we want to learn a model from the combined data $D = \{D_1, D_2, ..., D_M\}$ to predict $(Y_1(s'), Y_2(s'), ..., Y_M(s'))$ for input $s' \in R^p$. Each output is modelled as a sum of three gaussian processes U, V, and W. V is unique to each output, U shares the same noise source to ensure the outputs are not independent, and W is additive noise. Thus we have $Y_k(s) = U_k(s) + V_k(s) + W_k(s)$. Let the covariance matrix be Cov^Y, then we have $Cov^Y = Cov^U + Cov^V + \sigma^2$. Following [13]:

$$Cov^U_{kk}(d) = \frac{\pi^{\frac{p}{2}} r_k^2}{\sqrt{|A_k|}} exp(-\frac{1}{4} d^T A_k d) \tag{2}$$

$$Cov^V_{kk}(d) = \frac{2\pi^{\frac{p}{2}} w_k^2}{\sqrt{|B_k|}} exp(-\frac{1}{4} d^T B_k d) \tag{3}$$

$$Cov^U_{kk'}(c) = \frac{\pi^{\frac{p}{2}} r_k r_{k'}}{\sqrt{|A_k + A_{k'}|}} exp(-\frac{1}{2} d^T A_k (A_k + A_{k'})^{-1} A_{k'} d) \tag{4}$$

Where r, w, A, and B are parameters of gaussian kernels, and d are separation between two inputs s_i and $s_{i'}$. Given $Cov^Y_{kk'}$, the covariance matrix C is constructed as below:

$$\begin{bmatrix} Cov^Y_{11} & \dots & Cov^Y_{1M} \\ \vdots & \ddots & \vdots \\ Cov^Y_{M1} & \dots & Cov^Y_{MM} \end{bmatrix} \tag{5}$$

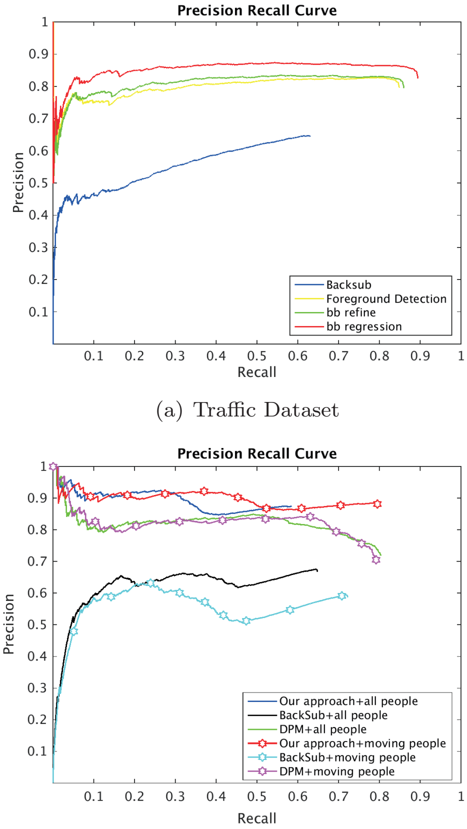
Now we can compute the log-likelihood:

$$L = -\frac{1}{2} log|C| - \frac{1}{2} \vec{o}^T C^{-1} \vec{o} - \frac{N}{2} log(2\pi) \tag{6}$$

where C is a function of parameters $\{r, w, A, B, \sigma\}$. And the mean μ of gaussian kernel is set to $\vec{0}$ in our algorithm. Learning a model corresponds to maximizing log-likelihood L, in our algorithm, the parameters are learned with gradient descent. The predictive distribution at the k^{th} output is a gaussian with mean $\hat{\mu} = \vec{q}^T C^{-1} \vec{o}$ and variance $\hat{\sigma} = \kappa - \vec{q}^T C^{-1} \vec{q}$, where $\kappa = Cov^Y_{kk}(0)$ and $\vec{q} = \{C^Y_{k1}(s' - s_{11}), ..., C^Y_{kN_1}(s' - s_{1N_1}), ..., C^Y_{kN_M}(s' - s_{N_M 1}), ..., C^Y_{kN_M}(s' - s_{N_M N_M})\}$.

3.3 Detection Based on Center Prediction

Given a set of predicted center locations, mean-shift clustering is used to generate object hypotheses. Suppose we have K object hypotheses returned by mean-shift clustering, each moving point is then assigned to



(a) Traffic Dataset

(b) VIRAT Dataset

Figure 2. Precision Recall Curves. "Moving people" is evaluation only on ground-truth people who are moving, "all people" is the entire set (moving and stationary).

the nearest cluster center. Possible foreground regions of the K objects are the convex hulls of these K clusters and these convex hulls are the initial object proposals.

As is often the case, objects have shadows and points belonging to shadows also move together with objects. Beyond this, the generation of moving points may be noisy – e.g. a few points far from real moving regions may be generated. Thus the convex hulls may also cover certain areas of the background. To shrink the convex hull, we compute tight bounding boxes which enclose the convex hulls and grid the bounding boxes. Grid cells are pruned if there are too few moving points inside. The convex hull of the remaining grids cells is used as the final bounding box.

3.4 Classification

Given all candidate bounding boxes produced by our method, the final step is to score them according to whether they contain an object of interest. For example, we might be interested in detecting the people in a scene or the vehicles in a scene. We do this by training a classifier based on dense trajectory features.

We consider all dense trajectory features that pass through a candidate bounding box in one frame. Follow the standard approach, we vector quantize dense trajectory features into a bag of words representation using 100 words. A discriminative classifier (SVM) is trained from a labeled training data set to classify each candidate bounding box as containing an object of interest or not.

4 Experiments

We test our model on two object detection problems in surveillance video: vehicle detection and human detection.

4.1 Traffic Dataset

Our model is first tested on a traffic dataset, where the focus is on detecting vehicles. Example frames can be seen in Fig. 3(top). The training set contains 500 frames of size 480×704 pixels. The test set contains 2258 frames with 2501 vehicles; ground-truth is obtained via manual labeling.

For training the regression model, all features extracted from the training set are clustered into 10 subsets to generate 10 outputs. 50 trajectories from each cluster that pass through ground truth bounding boxes are randomly picked to form the inputs of the regression model.

In testing, we mark a region of interest corresponding to regions of the video frame where vehicles are of sufficient size. For this dataset, the scale changes are very large: 250 pixels diagonal to 20 pixels diagonal. It is difficult to find a perfect bandwidth for entire images: if the bandwidth is too small, vehicles in larger scale will be over-segmented and if the bandwidth is too large, vehicles in smaller scale will be grouped into one cluster. To address this problem, the image coordinates are manually divided into two regions. For near-field regions, the vehicles have relatively large scale and thus mean-shift clustering is performed in world coordinates with bandwidth 3.5. For far-field regions, mean-shift clustering is performed in image coordinates with bandwidth 20. Note that in a fixed-camera surveillance setting, these parameters could be obtained with camera calibration.

To further improve the quality of the bounding boxes, we refine bounding boxes in two steps. We merge spatially nearby clusters constructed from dense trajectory points moving in the same direction, and learn a simple model to adjust bounding box sizes based on their image coordinates. For the merging step, we compute the average length of vehicles in the world coordinates, which is 20, and if two blobs are close enough and both lengths are less than 10, these two blobs will be merged into one blob. Given the dense trajectories points and features, training takes 4 hours to converge and during test time, every 10K trajectories take 23 minutes to process.

We compare every step of our method with a baseline which is background subtraction plus the same classifier used in our approach. We summarize the comparison of our model with the baseline in precision-recall curves shown in Fig. 2(a). The red curve corresponds to our final detection result and the blue curve corresponds to the result of background subtraction.

4.2 VIRAT Dataset

The second dataset we use is the VIRAT dataset [14]. We focus on detecting moving people in the video sequences. The VIRAT dataset contains a large amount of static surveillance camera video. We use Scene 0000, a parking lot scene containing people along moving background clutter such as vehicles.

We define a training set containing 500 frames with frame size 1080×1920 pixels. Again, for regression all

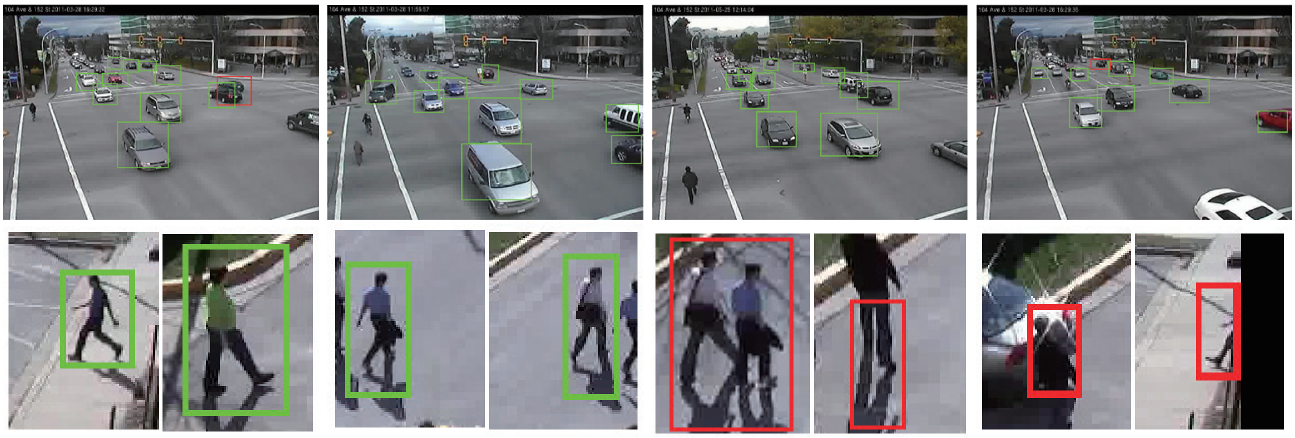


Figure 3. Visualization of detection results. Top row shows vehicle detection results on the Traffic dataset, bottom row shows human detection results on the VIRAT dataset. The green bounding boxes are true positives and the red bounding boxes are false positives. For the second row, the first 4 examples are top scoring true positives, the last four examples are the top scoring false positives.

features extracted from the training set are clustered into 10 subsets to generate 10 outputs. 50 trajectories of 5 people from each cluster going through ground truth bounding boxes are randomly picked to form the inputs of the regression model.

Our test set contains 1900 frames and has 1571 manually labelled ground truth person locations. The image coordinates are manually divided into two regions, mean-shift clustering is performed in these two regions separately. The bandwidth of the near-field and far-field regions are 50 and 25, respectively. Detections with size smaller than a threshold are removed. The two verification steps used in traffic dataset are not used on this dataset. We compare our method to baselines of DPM [15], trained on the same positive data and including hard negative mining, and background subtraction. We consider performance both on all people in the test set ("all people") and only those that are moving ("moving people"). The comparisons of final detection results with baselines are shown in Fig. 2 (b), using the standard $i/u > 0.5$ criterion. Note that our methods (red and blue curves) achieve higher precision than the baselines. As expected, the static-person DPM detector achieves higher recall for the "all people" setting.

5 Conclusion

A detection algorithm based on multi-output gaussian process regression using dense trajectories is proposed in this paper. The regression method can generate candidate detection bounding boxes superior to a baseline based on background subtraction. This is possible because regression from moving points to possible centers are advantageous in that moving points densely cover the moving objects. Another factor to the success of our method is that the features extracted from trajectory are quite stable. Limitations to our method include that since it is based on motion, only moving objects can be detected. Objects covering few pixels or overlapping in the image plane also present challenges. In summary, our method demonstrates that dense trajectories are effective for object detection in surveillance video.

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[2] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *PAMI*, vol. 19, pp. 780–785, July 1997.

[3] B. Coifman, D. Beymer, P. Mclauchlan, and J. Malik, "A realtime computer vision system for vehicle tracking and traffic surveillance," *Transportation Research C 6C*, vol. 4, pp. 271–288, Aug 1998.

[4] R. Cutler and L. Davis, "Robust real-time periodic motion detection, analysis, and applications," *PAMI*, vol. 22, no. 8, 2000.

[5] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *CVPR*, 2011.

[6] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *TCSVT*, October 2008.

[7] S. Zhang, C. Bauckhage, and A. Cremers, "Informed haar-like features improve pedestrian detection," in *CVPR*, 2014.

[8] P. Luo, Y. Tian, X. Wang, and X. Tang, "Switchable deep network for pedestrian detection," in *CVPR*, 2014.

[9] V. Rabaud and S. Belongie, "Counting crowded moving objects," in *CVPR*, 2006.

[10] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *ECCV*, 2006.

[11] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *ICCV*, pp. 734–741, 2003.

[12] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *CVPR*, 2005.

[13] P. Boyle and M. Frean, "Dependent gaussian processes," in *NIPS*, 2005.

[14] S. Oh, A. Hoogs, A. Perera, *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR*, 2011.

[15] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *PAMI*, vol. 32, no. 9, 2010.