Anomaly Detection

Sirisha Boothapati
Advisor: Natalia A. Schmid

July 03, 2012

1 Method

Anomaly detection is to detect any abnormal behavior of people in the video sequence. The data set used to produce results is PETS 2006 data which consists of video sequences from 4 cameras. We used video frames from 3 cameras for our experiments. Training sequence of 120 frames is selected from each video sequence. The trained parameters are tested on 1000 frames captures from each camera.

Training phase consists of computing the homography, background subtraction, tracking individuals using kalman filter and training the velocities into two groups using Support Vector Machines(SVMs). These trained parameters are applied on testing data. Testing phenomenon starts with tracking individuals from frame to frame and classifying the group to which the object belongs. Each object has decision from all the 3 cameras and the final decision is obtained as the fusion of all decisions using homography.

The process and theory of homography, background subtraction, tracking, classifying the objects is explained in Sec. 2,3,4, 5 respectively. Results for each section are shown.

2 Homography

Homography is an invertible transformation from a projective space to itself that maps straight lines. Consider a point $x = (u, v, 1)$ in one image and $x' = (u', v', 1)$ in another image, homograohy relates the pixel coordinates in the two images if

Figure 1: Frame captured by Camera 1. Points selected for homography computation are shown in blue.

$x' = Mx$. Homography is a 3×3 matrix M given by

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$$

To compute homography, the first frame from each video sequence is considered. Six common points from all the views are manually plotted as shown in Figs. 1, 2, 3. Homography is computed using these points.

3 Background Subtraction

Given a video sequence, the first thing to be done is to estimate the foreground or the background. Objects in the video sequence are represented by foreground objects in that image. A reliable and robust background subtraction algorithm should handle sudden or gradual illumination changes, high frequency, repetitive motion in the background and long term scene changes.The basic background subtraction method has to estimate the background for time t, subtract the estimated

Figure 2: Frame captured by Camera 3. Points selected for homography computation are shown in blue.

background from the input frame and apply a threshold th to the absolute difference to get the foreground mask. To estimate the background we used Mixture of Gaussians technique.

3.1 Mixture of Gaussians

Univariate Normal Gaussian Distribution is given by

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{1}$$

Multivariate Normal Gaussian distribution is given by

$$\mathcal{N}(X|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1}(X-\mu)} \tag{2}$$

The values f a particular pixel is modeled as a mixture of adaptive gaussians. Since multiple surfaces appear in a pixel, mixture of gaussians is used. Due to change in lighting conditions, the gaussians have to be adaptive to the change in lighting. At each iteration, gaussians are evaluated using a simple heuristic to

Figure 3: Frame captured by Camera 4. Points selected for homography computation are shown in blue.

determine which ones are mostly likely to correspond to the background. Pixels that do not match with the background gaussians are classified as foreground. Foreground pixels are grouped using 2D connected component analysis. At any time t, history of a particular pixel (x_0, y_0) is known as

$$X_1, ..., X_t = I(x_0, y_0, i) : 1 \leq i \leq t \tag{3}$$

This history is modeled by a mixture of gaussian distributions as

$$P(X_t) = \sum w_{i,t} \mathcal{N}(X_t | \mu_{i,t}, \Sigma_{i,t}) \tag{4}$$

An online K means approximation is used to update the gaussians. If a new pixel value, X_{t+1} can be matched to one of the existing gaussians, that gaussian's $\mu_{i,t+1}$ and ${\sigma^2}_{i,t+1}$ are updates as

$$\mu_{i,t+1} = (1 - \rho)\mu_{i,t} + \rho X_{t+1} \tag{5}$$

$$sigma^2{}_{i,t+1} = (1 - \rho){\sigma^2}_{i,t} + \rho(X_{t+1} - \mu_{i,t+1})^2 \tag{6}$$

where

$$\rho = \alpha \mathcal{N}(X_{t+1} | \mu_{i,t}, {\sigma^2}_{i,t}) \tag{7}$$

Figure 4: Background extracted from training sequence from Camera 1

and α is a learning rate. Prior weights of all gaussians are adjusted as follows:

$$w_{i,t+1} = (1 - \alpha)w_{i,t} + \alpha(M_{i,t+1}) \tag{8}$$

where $M_{i,t+1} = 1$ for the matching gaussian and $M_{i,t+1} = 0$ for all the others. If X_{t+1} do not match to any of the existing gaussians, the least probably distribution is replaced with a new one. The gaussians with the most supporting evidence and least variance should correspond to the background. Advantages of this method is that, a different threshold is selected for each pixel and these are adaptive by time. Objects are allowed to become part of the background without destroying the existing background model and also provides fast recovery.

To compute the background, we considered a sequence of 120 frames from each camera and then applied the above technique. Results obtained after background subtraction are shown in Figs. 4, 5, 6.

4 Kalman Filter

The Kalman filter is a set of mathematical equations that provide an efficient computational means to estimate the state of a process, in a way that minimizes the

Figure 5: Background extracted from training sequence from Camera 3

mean of the squared error. The filter is very powerful in estimating past, present and future states. The kalman filter addresses the general problem of trying to estimate the state $x \in R^n$ of a discrete time controlled process that is governed by the linear stochastic difference equation

$$x_k = Ax_{k-1} + bu_{k-1} + w_{k-1}, \tag{9}$$

with a measurement $z \in R^m$ that is

$$z_k = Hx_k + v_k \tag{10}$$

The random variables w_k and v_k represent the process noise and measurement noise respectively. They are assumed to be independent of each other with normal probability distributions

$$p(w) \sim N(0, Q) \tag{11}$$

$$p(v) \sim N(0, R) \tag{12}$$

We define $\widehat{x_k^-} \in R^n$ to be our apriori state estimate at step k given knowledge of the process prior to step k, and $\widehat{x}_k \in R^n$ to be our a posteriori state estimate

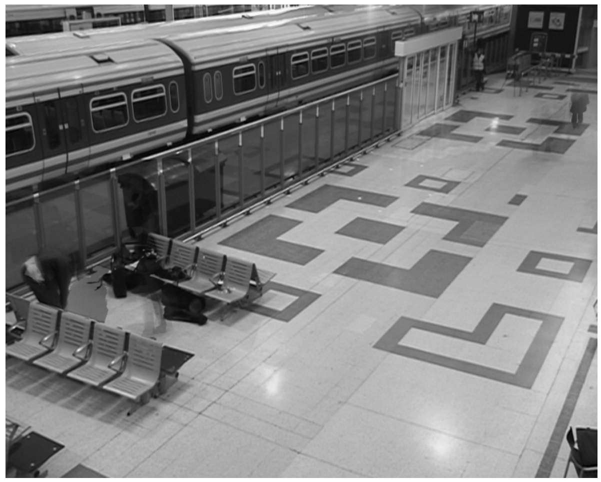

Figure 6: Background extracted from training sequence from Camera 4

at step k given meansurement z_k. We can then define a priori and a posteriori estimate errors as

$$e_k^- \equiv x_k - \widehat{x}_k^- \tag{13}$$

$$e_k \equiv x_k - \widehat{x}_k \tag{14}$$

4.1 The discrete Kalman Filter Algorithm

The kalman filter estimates a process by using a form of feedback control: The filter estimates the process state at sometime and then obtains feedback in the form of measurements. As such, the equations for the kalman filter fall into two groups: time update and measurement update equations. The time update equations are responsible for projecting forward the current state and error covariance estimates to obtain the a priori estimates for the next time step. The measurement update equations are responsible for the feedback which incorporates a new measurement into the a priori estimate to obtain an improved a posteriori estimate. The time update equations can also be thought of as a predictor equations, while the measurement update equations can be thought of as a corrector equations.

The specific equations for the time updates are

$$x_k = Ax_{k-1} + bu_{k-1} + w_{k-1}, \tag{15}$$

$$P^{-}{}_{k} = AP_{k-1}A^{T} + Q \tag{16}$$

The specific equations for the measurement updates are

$$K_k = P^{-}{}_{k}H^{T}(HP^{-}{}_{k}H^{T} + R)^{-1} \tag{17}$$

$$\widehat{x}_k = \widehat{x}_k^{-} + K_k(z_k - H\widehat{x}_k^{-}) \tag{18}$$

$$P_k = (I - K_k H)P^{-}{}_{k} \tag{19}$$

The first task is to compute the kalman gain K_k. The next step is to actually measure the process to obtain z_k and then to generate a posteriori state estimate by incorporating the measurement . The final step is to obtain a posteriori error covariance estimate. After each time and measurement update pair, the process is repeated with the previous a posteriori estimates used to project or predict the new a priori estimates. This recursive nature is one of the very appealing features of the kalman filter. The kalman filter recursively conditions the current estimate on all of the past measurements.

The measurement noise covariance R is usually measured prior to operation of the filter. The determination of the process noise covariance Q is generally more difficult as we typically do not have the ability to directly observe the process we are estimating. Sometimes a relatively simple process model can produce acceptable results if one injects enough uncertainity into the process through the selection of Q.

Kalman filter tracking is applied on the training data of 120 frames. Tracking points of all objects is obtained.

5 SVM Classifier

Given training data (X_i, y_i) for $i = 1....N$, with $X_i \in R^d$ and $y_i \in \{-1, 1\}$, learn a classifier $f(X)$ such that

$$y_i = \begin{cases} +1 & \text{if } f(X_i) \geq 0 \\ -1 & \text{if } f(X_i) < 0 \end{cases} \tag{20}$$

Support vector machines are motivated to classify two sets of data by training linear machines with margins. SVMs reply on preprocessing the data to represent patterns in a high dimension typically much higher than the original feature space. A linear classifier is in the form of a line represented by

$$f(X) = w^{T}X + b \tag{21}$$

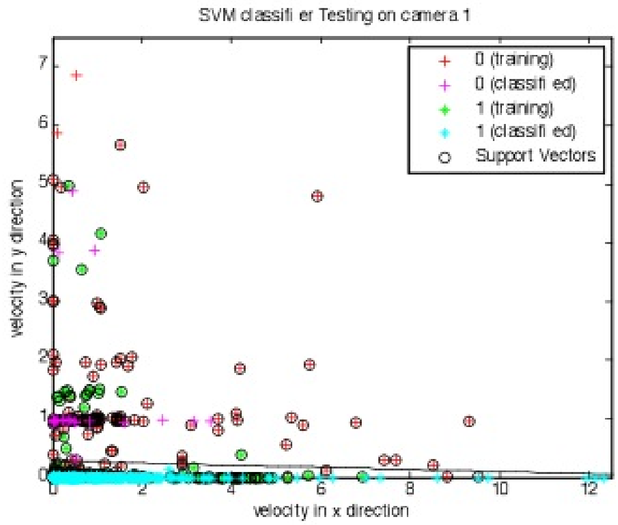

Figure 7: SVM classifier testing for Camera 1

where w is normal to the plane and b is the bias. w is also known as the weight vector. For a given training set of two classes, there exist multiple linear classifiers. Inorder to generalize the classifier, margin between the training classes data is used to compute a single classifier. Since $w^T X + b = 0$ and $c(w^T X + b) = 0$ define the same plane, we have the freedom to choose the normalization of w. Choose normalization such that $w^T x_+ + b = +1$ and $w^T x_- + b = -1$ for the positive and negative support vectors respectively. The margin is given by

$$\frac{w^T(x_+ - x_-)}{\|w\|} = \frac{2}{\|w\|} \tag{22}$$

Learning the SVM can be formulated as an optimization $max_w \frac{2}{\|w\|}$ subject to

$$w^T X_i + b \begin{cases} \geq +1 & \text{if } f y_i = +1 \\ \leq -1 & \text{if } y_i = -1 \end{cases} \tag{23}$$

The data used to train the classifier is a two class data. One belongs to stationary objects and the other belongs to moving objects. SVM classifier is trained using this data. The classifier is used to test on the testing velocities obtained from the tracking points. Figs. 7, 8 shows the classifier training and testing results.

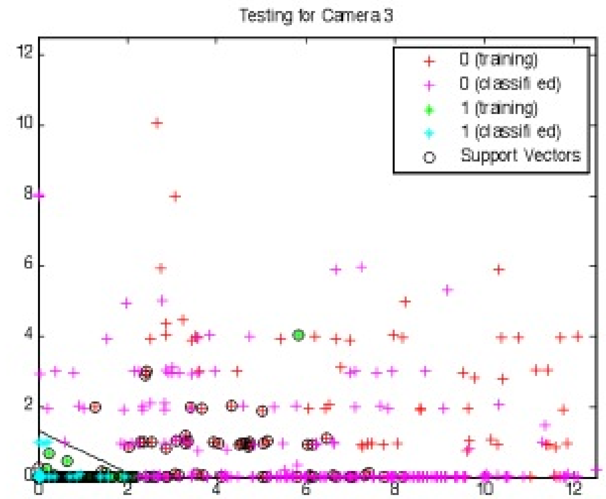

Figure 8: SVM classifier testing for Camera 3

Each object has a decision for each view. The decision is combined using homography. Points detected as anomaly are projected onto other views using homography. Final decision of an object is based on the majority rule.

Results of three camera views for a corresponding frames is shown in Fig. 9, 10, 11. Abnormality is shown in red circle.

Figure 9: Camera 1 frame

Figure 10: Camera 3 frame

Figure 11: Camera 4 frame