Efficient Moving Object Detection Based On Statistical Background Modeling

Kusuma.U[1], S.T Bibin Shalini[2]
[1,2]Dept. Electronics and Communication
Address AMC Engineering College, Bannerghatta road, Bangalore

Abstract— Tracking vehicles is an important and challenging problem in video-based Intelligent Transportation Systems, which has been broadly investigated in the past. A robust method for tracking vehicles is implemented in this thesis work.

The proposed algorithm includes three stages: object detection, counting and tracking. Vehicle detection is a key step. The concept of moving object detection is built upon the segmentation method. Background subtraction method is used in this work. According to the segmented object shape, a predict method based on Kalman filter is proposed. By assuming that the vehicle moves with almost a constant acceleration from the current frame to the next, a Kalman filter model is used to tracking and predicting the trace of a vehicle.

The model can be used in the traffic analysis as it is capable of tracking and counting multiple targets in a big area hence forming an effective, efficient, practical vehicle tracking system. The proposed method has been tested on few traffic-image sequences and the experimental results show that the algorithm is robust and can meet the requirement.

1. Introduction

Object tracking is the problem of estimating the positions and other relevant information like trajectory, shape, size and number of moving objects in an image sequence. So, a tracker assigns consistent labels to the tracked objects in different frames of a video. It has several important applications such as security and surveillance, annotation of videos, traffic management, motion-based video compression and interactive games.

Object tracking, in general, is a challenging problem. Difficulties in tracking objects can arise due to abrupt object motion, changing appearance patterns of the object and the scene, nonrigid object structures, object-to-object occlusions, and camera motion. The complex task of tracking can be simplified by imposing constraints on the motion and/or appearance of objects. For the given video sequence, we assume that the object motion is smooth with no abrupt changes. The object motion is further constrained to be of constant velocity. Prior knowledge about the number and the size of objects, object appearance and shape, is also made use of to simplify the problem.

We are interested in detecting and tracking moving objects in video at low- to moderate- resolution and frame rate. We have developed a flexible tracking pipeline that allows us to investigate different combinations of foreground extraction, feature extraction and motion correspondence algorithms. In foreground extraction we have explored applications of background subtraction techniques and salient region extraction. Our background subtraction research includes the investigation of the effectiveness of popular background subtraction techniques and the development of a new technique for background subtraction with foreground validation. We have also examined the application of salient region detection to extracting moving objects.

In feature extraction, we extract features such as the centroid, the size, and the average pixel intensity of each moving object. These are then used in tracking algorithms such as Kalman filters to track the objects from one frame to the next. Additional logic is incorporated for track maintenance to determine constraints such as the number of frames over which an object must be tracked successfully for it to be assigned a track and the number of frames over which no object is assigned to an existing track, making the track disappear.

In this thesis our approach uses frame differencing for background and fore ground separation. Blob detection for counting the no of detected objects. Centroid is the feature extracted and these centroid co-ordinates are used for tracking. Tracking is achieved using Kalman filter.

2. Problem Description

Given a video sequence (recorded), our goal is to detect if there are entities in that sequence that are changing their spatial position. When these entities are detected we should be able to estimate their spatial position (within certain limits) such that if certain events occur (an entity is temporarily disappears shortly from the sequence) we still want to be able to predict its trajectory and behaviour until the event will eventually stop.

We will make any assumption regarding the type of the scene recorded in the video sequence should posses a non complex fixed background. We are in need of a model that is capable in emulating the motion of an object in that particular frame of the video sequence with respect to the background and a model which is robust enough to withstand various influences exerted by the environment. In addition to this, our model will also have to take into account the noise injected by the capturing device.

At the end of the process, our model should provide us with an estimation of the trajectory of the object whose motion it tried to emulate as well as the number of moving entities detected.

3. Proposed Solution

We use background subtraction for segmentation and detection of objects and use Kalman Filter for tracking the objects.

3.1. Background Subtraction

Background subtraction is an approach for detecting changes in the background of the scene. It assumes an initially static scene where there is no motion and builds mean and standard deviation images for the scene during this phase. Since the input video to our system has no such initial frames with no motion, we initialise the mean image by taking the median of the first 100 frames and the standard deviation image as the difference be the frame and the initial median image averaged over the first 100 frames.

$$|I(x, y) - \mathrm{MeanImage}(x, y)| > c \times \mathrm{StdDevImage}(x, y) \qquad (1)$$

Once we have our initial background images, we label a pixel as being in foreground if for any channel Eq. (1) holds true. We chose the constant $c = 2.1$ in Eq. (1).Hence we obtain a mask for the foreground pixels in the scene.

Another issue that needs to be addressed here is that the input video in our case has some camera shake and the camera moves considerably in one or two frames. The sudden camera movement results in some background objects also coming into the foreground. Our approach to solving this problem was trying to absorb static objects into the background. Hence we maintained a counter for each pixel as to if some pixel was continuously in the foreground for a threshhold number of frames, we removed it from the foreground mask. We chose this threshhold to be 15 frames. Once we removed the pixel from the foreground mask, we updated our background mean and standard deviation images.

This was done as a running average between the previous mean image and the background in the current frame.

The background standard deviation image was updated similarly as a running average between previous standard deviation image and the difference between current frame and mean background image.

3.2. Detection of Objects

Background subtraction gives us candidate foreground pixels. We further need to cluster the foreground pixels into candidate objects. We initially use morphological operations of image opening and closing. Image closing operation fills the holes, if any, in some cluster of foreground pixels. Image opening operation removes potential noise foreground pixels that connect two clusters. After these operations we remove the foreground clusters having area less than 200 pixels and use connected component labelling which gives us potential objects in the scene. We maintain areas, bounding boxes and centroids of each such object in the foreground.

3.3. Tracking of objects - Kalman filter

Once we have detected some objects, we need to find which of these objects correspond to previously detected objects and those which are detected for the first time. The tracking is done by fitting a discrete time linear dynamical system(LDS) to each object. Each object has an associated state which is a vector $[x, y, vx, vy]^T$, where x, y is taken as the center of the bounding box of the object and vx, vy are the velocities along x and y directions. We assume that the vehicles move with constant velocities and hence do not take into account the accelerations of the objects. This assumption is valid considering the fact that the velocities of objects in the video do not change considerably between two consecutive frames. The output of the LDS is the actual centers of the bounding boxes of the detected objects.

In principle, the Kalman filter may have applications both to discrete-time systems as well as continuous-time systems but the continuous domain is not to be discussed herein. Our system is treated in discrete-time due to the nature of the digital computing methods used, and the arrival of sensor data in a time-quantized manner. The purpose of the kalman filter is to estimate the state $x \in R^n$ of a discrete-time process that is perturbed by some noise. The nature of that noise is crucial in the definition of the filter, and in its performance. All sensor and process noise is presumed to be zero-mean-Gaussian (0,). The system being filtered has inherent system properties described by a model. The evolution of the state is governed by the following discrete time stochastic difference equation:

Xk = Ak Xk-1 + Bk Uk-1 + Wk-1

Sensor data is used to correct this state, but measurement of this process may not necessarily be performed in the same space as the state itself. It is also assumed to have had some noise introduced in the form of sensor noise. We define the measurement or otherwise called the observation of Zk∈ Rn as:
Zk = Hk Xk +vk

Kalman filter is divided into two distinct stages : Prediction stage and update stage The ability of these two stages to be partitioned forms an important implication. If sensor data is not available for a stretch of time, or if two or more sensors are operating at different frequencies, the filter may continue to operate, giving an estimate of the true state. The prediction stage consists of the dynamics themselves, along with the incorporation of another matrix that represents the degree to which the state can be trusted. Every time the system predicts what the state should be, it also predicts how accurate the state estimation is. On a prediction, the state covariance matrix increases, while every time sensor updates the state, it is reduced. The filtering equations themselves are presented below with annotation as to their significance

Prediction Stage:

Predicted State

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k U_k$$

Predicted estimate covariance

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k{}^T + Q_k$$

Update Stage

- Innovation (error)

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1}$$

Innovation Covariance

$$S_k = H_k P_{k|k-1} H_k{}^T + R_k$$

Kalman Gain (weighting factor)

$$K_k = P_{k|k-1} H_k{}^T S_k{}^{-1}$$

Updated state estimate x

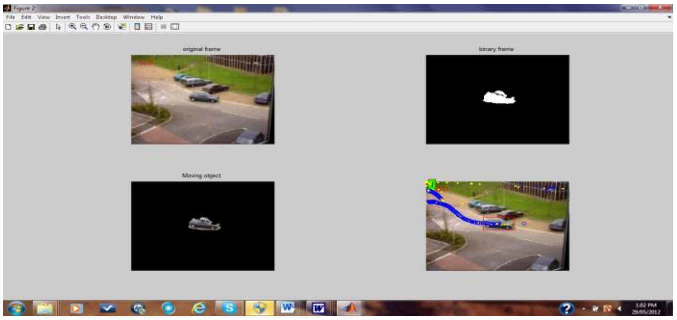
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k$$

Updated estimate covariance

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

While the derivations themselves will not be presented in this thesis, the implementation of these equations is very straightforward in MATLAB.
Certain initializations are required for the state vector and matrices defined above. If the filter began operating with the wrong assumptions, many time-steps may occur before the filter starts reflecting what the internal state truly is, due to the internal dynamics. Since the tracking example presented assumes the system is initialized while the vehicle is standing still, the state vector is initialized to zero.

4. Results



Conclusions

This work proposes an efficient algorithm for detecting a moving object using background elimination technique. Initially we compute the frame differences (FD) between frames Fi and Fi+k. The frame differences obtained are then compared with one another which help in identifying the stationary background image. The moving object is then isolated from the background. In the post processing step, the noise and shadow regions present in the moving object are eliminated using a morphological gradient operation that uses median filter without disturbing the object shape. This could be used in real time applications involving multimedia communication systems.

This paper also discusses an application system of traffic surveillance. Here we develop an algorithm to track and count dynamic objects efficiently. The tracking system is based on a combination of a temporal difference and correlation matching. Tracking system uses Kalman filter to predict the path of the moving object.

The system effectively combines simple domain knowledge about object classes with time domain statistical measures to identify target objects in the presence of partial occlusions and ambiguous poses in which the vehicles are moving. Ground clutter is effectively rejected. The experimental results show that the accuracy of counting vehicles is good although the vehicle detection was computational complexity of our algorithm is linear to the size of a video clip and the number of vehicles tracked. As a future work a combination of higher dimensional features with some additional constraints may be tried so that adverse effects of some features can be compensated by contribution of others.

References

[1] http://www.darpa.mil/grandchallenge.
[2] http://www.ieeexplore.org
[3] www.google.com

[4] L.Li I.Y.H Gu, M.K.H. Leung, Q. Tian, "Adaptive background subtraction based on feedback from fuzzy classification"

[5] Rafael C. Gonzalez, Richard E. Woods,"Digital Image Processing"

[6] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing"

[7] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russel. "Towards robust automatic traffic scene analysis in real-time."

[8] Chris Stauffer W.E.L Grimson "Adaptive background mixture models for real-time tracking"

[9] Nir Friedman and Stuart Russell. "Image segmentation in Video sequences: A probabilistic approach,"

[10] Chris Stauffer W.E.L Grimson, "Adaptive background mixture models for real-time tracking"