

# Pedestrian Detection and Tracking Based on A Novel Background Subtraction & Particle Filter

Defang Liu, Ming Deng

College of Computer&Information,Fuyang Normal College,Fuyang,Anhui 236037,China

mmludefang@yeah.net,mdeng76@163.com

**Keywords:** pedestrian; detection; tracking; background subtraction; particle filter

**Abstract.** This paper focuses on pedestrian detection of moving objects, and then tracking the detected objects. The pedestrian detection is performed using an algorithm based on a type of background subtraction technique. This technique not only uses a median-based approach for initial background estimation but also a novel two-pass approach for noise removal. The tracking of the detected object is then done using an offline particle filter. The proposed applications of the algorithm presented in this paper, could be pedestrian activity monitoring, ITS, etc. The experiment indicates that the algorithm can detect and track the pedestrian effectively.

## Introduction

Object detection and tracking is an important task within the field of computer vision, due to its promising applications in many areas, such as video surveillance, traffic monitoring, vehicle navigation etc. The availability of high quality and inexpensive video cameras and the increasing need for automated video analysis has generated a great deal of interest in the areas of motion detection, object tracking and gesture analysis. Thus on a very high-level, its possible to identify three key steps in video analysis: detection of interesting moving objects, tracking of the detected objects from frame to frame, and analysis of the object tracks to recognize their behavior. This paper focuses mainly on the problem of robust foreground segmentation of moving objects.

The paper is structured as follows. In next section, describes the working of the object detection algorithm implemented in this paper. In section 3, talks about the problem of tracking multiple objects and the proposed solution. In sections 4, describes the results of the experiment, and finally the conclusions and future work is presented in section 5.

## Object detection algorithm

Object detection methods have been classified as point detectors, segmentation, background subtraction, and supervised classifiers in [1]. The method of choice in this paper is based on the background subtraction algorithm. Here, object detection is achieved by building a representation of the scene called the background model and then finding deviations from the model for each incoming frame. A threshold difference in an image region from the background model signifies a moving object. The main drawback of this approach is the sensitiveness to dynamic scene changes due to lighting and extraneous events.

Several approaches for automatically adapting a background model to dynamic scene variations have been proposed in literature. Such methods differ mainly in the type of background model used and in the procedure used to update the model. In order to learn gradual changes in time, Fatih Porikli etc. [2] propose modeling the color of each pixel of a stationary background with a single 3D (Y, U, and V color space) Gaussian. Thus  $I(x, y) = N(\mu(x, y), \Sigma(x, y))$ . The model parameters, the mean  $\mu(x, y)$  and the covariance  $\Sigma(x, y)$ , are learned from the color observations in several consecutive frames. Once the background model is derived, for every pixel  $(x, y)$  in the input frame, the pixels that deviate from the background model are labeled as the

foreground pixels. However, a single Gaussian is not a good model for outdoor scenes since multiple colors can be observed at a certain location due to repetitive object motion, shadows, or reflectance. A substantial improvement in background modeling is achieved by using multimodal statistical models to describe per-pixel background color, such as using a mixture of Gaussians to model the pixel color [3].

The approach used here, is similar to the one suggested in [4]. Step1~ Step4 shows a simplified flow diagram of the algorithm proposed in [4].

Step1 Estimate initial background model  $\mu(x, y), \sigma(x, y)$

Step2  $P(x, y) = \max(|I_t(x, y) - I_{t-1}(x, y)|)$

Step3 If  $\text{mod}(\text{frame}, \text{update}) == 0$

then Stack-Update parameters,

$$\mu_{n+1}(x, y) = \alpha * \mu_n(x, y) + (1 - \alpha) * \mu_{n-1}(x, y)$$

$$\sigma_{n+1}(x, y) = \alpha * \sigma_n(x, y) + (1 - \alpha) * \sigma_{n-1}(x, y)$$

$$p_{n+1}(x, y) = \alpha * p_n(x, y) + (1 - \alpha) * p_{n-1}(x, y)$$

else if

$$|I_t(x, y) - \mu(x, y)| > \sigma(x, y) \text{ OR } |I_t(x, y) - I_{t-1}(x, y)| > p(x, y)$$

then Foreground Pixel

else Background Pixel

Step4 If the frame = last frame?

then End

else Step3

The proposed approach uses a training set (sequence of frames) in which there are only 'legal' moving objects, i.e. trees, leaves etc. and not human figures. The system uses this training period to determine a series of parameters which contain the information about tolerable motions. So during the normal surveillance operations, when a movement is detected it is considered legal if it is similar to the movements registered during the training period, while it is classified as illegal otherwise. Thus, according to this algorithm a pixel is marked as a foreground pixel if, at time  $t$ :

$$|I_t(x, y) - \mu(x, y)| > \sigma(x, y) \quad (1)$$

$$\text{or } |I_t(x, y) - I_{t-1}(x, y)| > p(x, y) \quad (2)$$

where  $I_t(x, y)$  and  $I_{t-1}(x, y)$  are the intensity values for the pixel  $(x, y)$  at time  $t$  and  $t-1$ ,  $\mu(x, y)$  and  $\sigma(x, y)$  are mean and variance respectively of the intensity values observed during the supervised training period, and  $p(x, y)$  is the maximum difference between intensity values that are consecutive in time, observed during the whole training period, given by

$$p(x, y) = \max_{t \in T} (|I_t(x, y) - I_{t-1}(x, y)|) \quad T = \text{training set} \quad (3)$$

The updating algorithm mentioned in [4], is given as,

$$m_{n+1}(x, y) = \alpha * m_n(x, y) + (1 - \alpha) * m_{n-1}(x, y)$$

$$\sigma_{n+1}(x, y) = \alpha * \sigma_n(x, y) + (1 - \alpha) * \sigma_{n-1}(x, y)$$

$$p_{n+1}(x, y) = \alpha * p_n(x, y) + (1 - \alpha) * p_{n-1}(x, y)$$

where the  $n+1$  indicates the new updated values,  $n$  indicates the values calculated during the last observation period, and  $n-1$  indicates the old values of the parameters. The updating procedure described above must be done only for pixels that have been classified as static for most of the observation period (i.e. 80%).

The above algorithm assumes that a training set of frames containing only 'legal' objects is available. In actual surveillance applications, it is difficult to guarantee the availability of such a training sequence. This means the initial background and background parameters would need to be estimated from a training sequence consisting of illegal objects. Moreover, even a slight displacement in the camera between the tracking and testing period can result in inaccurate initial background parameters and can have a disastrous effect.

To avoid dependency on a training set, a mode-based approach [6] or a median-based approach [5] can be used. This paper uses the median-based approach which allows background estimation from a training sequence consisting of even illegal objects. Thus, for a training set of N frames, the background can be constructed simply as

$$Bg^R(x, y) = \text{median}_{t \in T} \{I^R(x, y, t_i), \dots, I^R(x, y, t_{i-N-1})\}$$

$$Bg^G(x, y) = \text{median}_{t \in T} \{I^G(x, y, t_i), \dots, I^G(x, y, t_{i-N-1})\}$$

$$Bg^B(x, y) = \text{median}_{t \in T} \{I^B(x, y, t_i), \dots, I^B(x, y, t_{i-N-1})\}$$

Another major problem when using background subtraction algorithms is occurrence of false positives, where certain background regions get erroneously classified as foreground objects. A simple approach to overcome this problem is to compare the area of an object detected as foreground, with a fixed threshold. All the objects with areas smaller than the threshold are merged with the background. However, this approach by itself is generally not enough to remove all the false positives, because the noisy objects can sometimes have areas comparable to the areas of the actual object of interest. A more sophisticated approach to overcome this problem is to use a two-pass approach. In the first pass, the equations (1) and (2) are employed to get a 'loose' estimation of the foreground objects. Under the assumption that these objects mostly contain true objects but also some false positives, an approach (somewhat similar to hypothesis testing) can be employed to get rid of the false positives. This gives us an 'strict' estimate of the foreground objects. Combining the two, we can get a near, noise free object detection for each frame.

## Tracking

The aim of an object tracker is to generate a trajectory of the path followed by the moving object over time by locating its position in every frame of the video. The tasks of detecting the object and establishing correspondence between the detected object across frames can either be performed separately or jointly. We have used the second approach in this paper. Thus the object detection algorithm described in the previous section identifies regions in every frame corresponding to a moving object and then the tracker tracks the detected objects across frames.

Plotting the path of a single moving object in the scene is a trivial problem, since the path followed by the object can be simply the centroid of the foreground regions detected by the object detection algorithm. However, following trajectories for multiple objects is more involved, requiring the use filters like the alpha-beta filter, Kalman filter etc. In this method, we have used a particle filter tracker to track multiple objects from frame-to-frame. The particle filter works offline, and needs the centroids of all the detected objects as input, and gives the tracked paths (or tracks) as output.

## Results and discussion

Most of the videos used for the experiments are 30 frames/sec videos of size 640 by 480 pixels. They are taken in an outdoor environment, using a still video camera. A typical test clip consists of one or multiple persons walking across a cross-walk. The algorithm was implemented entirely in MATLAB, and works offline as of now. It was found that the high temporal resolution is not necessary for proper working of the algorithm, and hence the video sequence was sub-sampled (2:1) before feeding to the algorithm.

The left part of Fig.1 shows the output of the object detection algorithm. The trajectory followed by the moving objects can be clearly seen in the right part of the Fig. 1.



Fig. 1. Figures showing the detected object, and the trajectory followed by the moving object

### Conclusions and future work

The algorithm can detect and track multiple objects in the outdoor environment. The two-pass object detection mechanism, guarantees that only the objects of interest (pedestrians) are detected in each frame. The particle filter tracker ensures that the path followed by the multiple objects as long as they are in the scene is followed.

As of now, the particle filter works offline; hence the algorithm cannot be implemented in real-time. Also, using the median-based approach for background estimation, and the two-pass approach for noise removal, though effective; is computationally expensive. Future work should concentrate on implementing a filtering approach which would work real-time, and detection algorithms which are more computationally efficient.

### Acknowledgement

This work was financially supported by the Natural Science Fund of Anhui Province Education Ministry under grant NO.KJ2013A201.

### References

- [1] M. T. Gopala krishna, M. Ravishankar, D. R. Rameshbabu, Ten-Lopp: Tensor Locality Preserving Projections Approach for Moving Object Detection and Tracking . Advances in Intelligent Systems and Computing Volume 209, 2013, pp 291-300
- [2] Fatih Porikli, Alper Yilmaz: Object Detection and Tracking, Studies in Computational Intelligence Volume 409, 2012, pp 3-41
- [3] Varun Santhaseelan, Vijayan K. Asari: Moving Object Detection and Tracking in Wide Area Motion Imagery, Augmented Vision and Reality Volume 6, 2014, pp 49-70
- [4] M. T. Gopala Krishna, M. Ravishankar, D. R. Rameshbabu : ISSR: Intensity Slicing and Spatial Resolution Approaches for Moving Object Detection and Tracking under Litter Background , Communications in Computer and Information Science Volume 361, 2013, pp 525-536
- [5] E. Jayabalan, A. Krishnan: Object Detection and Tracking in Videos Using Snake and Optical Flow Approach, Communications in Computer and Information Science Volume 142, 2011, pp 299-301