Tracking Multiple Objects In Surveillance Cameras

Report Author : Shuo Zheng

KTH ROYAL INSTITUTE OF TECHNOLOGY

Abstract. Objects tracking plays an important role to the traffic management system, as accurate tracking and counting result will lead to a more efficient management system.The paper presents a method for tracking multiply moving Objects in a surveillance cameras. Firstly, Moving objects are detected and performed blob analysis. Then a kalman filter or particle filter is applied to every target to get a predicted position. Finally, We use Munkres' version of the Hungarian algorithm to compute an assignment between detected and predicted position to associates the detections corresponding to the same object. The paper also compares using kalman filter and particle filter.

Keywords: kalman filter; particle filter; vehicle tracking; moving objects tracking

1 Introduction

Vehicle tracking has been an active field of research within the past decade due to the increase in computational power and the development of video surveillance infrastructure. There is, today, a huge need for automatic traffic control and regulation, automatic video surveillance and abnormal event detection. Robust car tracking is a fundamental low-level task necessary to achieve such intelligence. There have been various techniques developed to track vehicles.

Surveillance cameras are the most common cameras in traffic system. They are used to monitor Vehicles moving on the road. One of the features of surveillance cameras is stationary, which is an advantage for analyzing the background and detecting vehicles. Generally, we want to get 2 information from camera. One is the traffic flow which can be used to analyze the road's situation ; In addition, specific Vehicles' behaviors are also what we are interested, because we can monitor if there is vehicle breaking traffic rules.

For the traffic flow, what we need is counting vehicles. In the project, our program will give a label from 1 for every vehicle appearing in the camera. Estimator and Matching algorithm is applied to make sure we give the vehicles consistent labels. For specific Vehicles' behaviors, we use the foreground detector to detect a moving object and track it. Once a vehicle is tracked, we can easily judge whether it run on the right way, whether its speed exceed the limit and so on.

In our project, we write code based on Motion-Based Multiple Object Tracking[1]. It Detects moving objects in each frame and associates the detections corresponding to the same object over time. Originally, it uses kalman filter to predict position, and use this predicted position for association. After testing, we draw the conclusion that the kalman filter has its disadvantages and it makes mistakes in some situations. So we try to use a particle filter to replace it. Finally, we compare the 2 filters' advantages and disadvantages.

1.1 Contribution

We developed a particle filter based on a program which can detect moving vehicles in a surveillance cameras and track them before they leave the range of camera. A label from 1 is given to all vehicles appearing in the camera. Kalman filter and particle filter are used separately in the program. We try both filter to predict a vehicle's position. Then, we analyze results of these 2 filters and find that in some situation Kalman filter perform better and in other situation, particle filter leads to a better result.

1.2 Outline

Chapter 2 contains some related work of tracking vehicle in a stationary camera. Chapter 3 introduces our method of tracking multiple Vehicles In surveillance cameras in detail. Chapter 4 includes the experiment result. Analyzing and comparing 2 filters is in this part. Finally, we summarize the whole project in chapter 5.

2 Related work

Video traffic surveillance is of high interest in the field of Intelligent Transportation Systems and the moving vehicle tracking is an essential technique. There have been various techniques developed to track vehicles. The most common ones undoubtedly rely on Bayesian filtering and, in particular, Kalman and particle filters.

Da-Peng Bai[2] apply particle filter for this problem. Discrete wavelet transform is used to detect the object, and a suitable model to characterize a state space.

Francois Bardet[3] presented a generic multi-vehicle real-time automatic tracking and labeling system, using a MCMC Particle Filter involving a vehicle kinematic model. However, there are still some problem such as weak foreground segmentation, tracking dark vehicle.

Bouttefroy[4]used a new particle filtering approach for object tracking in video sequences. The projective particle filter uses a linear fractional transformation, which projects the trajectory of an object from the real world onto the camera plane, thus providing a better estimate of the object position. The standard and the projective particle filters have been evaluated on traffic surveillance videos. It has been shown that the MSE on the trajectory of the vehicles is reduced with the projective particle filter.

A vehicle tracking method based on PF is presented by Mingxiu Lin.[5] It combines gray and contour feature particles using fusion algorithm to balance the weights according to the present scene. However, it cannot solve multiple vehicle tracking problems properly.

Yang Xu[6] combined Gaussian particle filtering and the variational level set method to track moving vehicle. The algorithm solves the initial location problem to the variational level set curve evolution and make Gaussian particle filtering tracking more precise.

Christiano Bouvi[7] takes into consideration the convex shape of the objects and background information to merge or split the groupings. After a vehicle is identified, it is tracked using the similarity of color histograms on windows centered at the particle locations. Very small vehicles can be missed, since the number of particles may be insufficient to generate a cluster.

Ashvini Kulkarni[8] contrast performance of Mean-Shift algorithms gradient descent based search strategy with Kalman Filter based tracking algorithm used to models the dynamic motion of target object to guide optimize objects position through time using Swarm Intelligence based Particle Swarm Optimization.Experimental results of tracking a car demonstrate that the proposed Kalman Filter for object tracking is efficient under dynamic environment.

Kenneth Tze Kin Teo[9] used GA resampling based PF algorithm has successfully tracked the target vehicle undergoing occlusion and maneuvering incidents in a parking lot because GA has the ability to converge the estimated posterior position instead of diverge to the wrong position.

S.Srilekha[10] update background by using Kalman filter to detect, track and count the vehicles. The proposed algorithm is tested with different videos and results clearly show the efficiency of the algorithm. Its major limitation is inability to manage dynamic background pixels.

Gaussian Mixture Model (GMM) method was applied for vehicle detection and Kalman Filter method was applied for object tracking by Indrabayu[11]. The results show that GMM method working properly in light traffic.

3 My method

Our program performs automatic detection and tracking of moving vehicles in a video from a surveillance camera. Generally, the program contains 2 parts: 1)Detecting moving objects in each frame , 2) Associating the detections corresponding to the same object over time. The detection of moving objects uses a background subtraction algorithm based on Gaussian mixture models. Morphological operations are applied to the resulting foreground mask to eliminate noise. Finally, blob analysis detects groups of connected pixels, which are likely to correspond to moving objects.The association of detections to the same object is based solely on motion. The motion of each track is estimated by a Kalman filter or a particle filter. The filter is used to predict the track's location in each frame, and determine the likelihood of each detection being assigned to each track.Track maintenance becomes an important aspect of this example. In any given frame, some detections may be assigned to tracks, while other detections and tracks may remain unassigned. The assigned tracks are updated using the corresponding detections. The unassigned tracks are marked invisible. An unassigned detection begins a new track.Each track keeps count of the number of consecutive frames, where it remained unassigned. If the count exceeds a specified threshold, the example assumes that the object left the field of view and it deletes the track.

Figure 1 is the flow chart of the whole program.

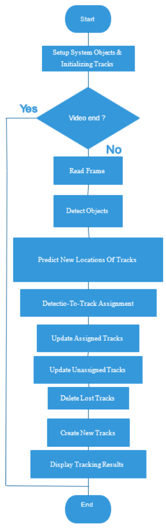

Fig. 1. The flow chart of the whole program.

Mainly, it has 9 steps.

1. **Setup System Objects and Initialize Tracks** : The program Creates System objects used for reading the video frames, detecting foreground objects, and displaying results in this step. The initialize Tracks function creates an array of tracks, where each track is a structure representing a moving object in the video.

   For each frame of video we do following steps.:
2. **Detect Objects**: The Detect Objects function returns the centroids, the bounding boxes of the detected objects and the binary mask. The mask picture example is as following figure 2.

3. **Predict New Locations of Existing Tracks** : Use the kalman or particle filter to predict the centroid of each track in the current frame, and update its bounding box accordingly.
4. **Detection-To-Track Assignment**: Assigning object detections in the current frame to existing tracks is done by minimizing cost. The cost is defined as the negative log-likelihood of a detection corresponding to a track.
5. **Update Assigned Tracks**: The update Assigned Tracks function updates each assigned track with the corresponding detection. It calls the correct method of vision.KalmanFilter to correct the location estimate.
6. **Update Unassigned Tracks**: This function marks each unassigned track as invisible, and increase its age by 1.
7. **Delete Lost Tracks**: The Delete Lost Tracks function deletes tracks that have been invisible for too many consecutive frames or be invisible for too many frames.
8. **Create New Tracks**: Create new tracks from unassigned detections.
9. **Display Tracking Results**:The Display Tracking Results function draws a bounding box and label ID for each track on the video frame and the foreground mask.

Fig. 2. The binary mask performed morphological operations.

Figure 3 shows how one track is preformed. Once a track is created, it will update in every frame before it is deleted. We will delete it when it meets one of the following conditions: 1)the fraction of the track's age for which it was visible is less than 0.6 2) It is consecutive invisible for more than 20 frames. Every track's structure has a estimator(kalman filter or particle filter) which is used for predicting position.

In track's every survival frame, track will get a predicted position by its estimator firstly. Then, the program uses these predicted positions from all tracks and detected positions to do assignment.The cost takes into account the Euclidean distance between the predicted centroid of the track and the centroid of the detection. The assignment problem is solved by the Munkres' version of the Hungarian algorithm. Finally, we can get pairs of matched predicted positions and detected positions which represents that which detected positions belong to which tracks.

When the track is assigned successfully, it will correct its estimator's position, store new bounding box, add total visible count by 1, set consecutive invisible count as 0 and add age by 1. When the track is not assigned successfully, it add consecutive invisible count by 1 and add age by 1.

When total visible count is greater than minimum visible count(we set 8 here), the bounding box and label will be displayed. If an object has not been detected in this frame, its predicted bounding box will be displayed.

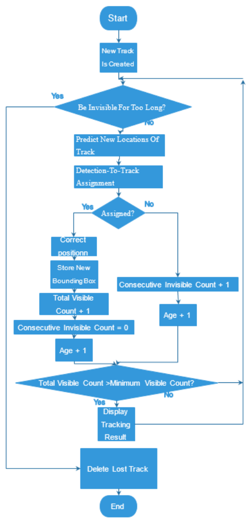

Fig. 3. The flow chart of one tracks.

3.1 Particle Filter

The original program use vision.KalmanFilter for perdition. We change it to a particle filter.

We save 10 or 100 particles in every tracks' structure. The program have 3 function for particle filters: pfDiffusion, pfResample and pfCorrect. pfDiffusion and pfResample is called in Predict New Locations of existing tracks function for predicting position. pfCorrect is called in Update Assigned Tracks function.

1. **pfDiffusion**: In this function, we apply diffusion to the state. R is model noise. And we should make sure the diffusioned positions are limited in the mask.
$$x_t = x_{t-1} + N(0, R) \tag{1}$$
2. **pfResample**: Measurements are applied on the binary mask.
$$q(u,v) = \begin{cases} 0, background \\ 1, detectedmovingobject \end{cases} \tag{2}$$

q is pixel of mask, u and v are the position of mask. So the measurements are binary. Systematic Re-Sampling[12]is applied.The algorithm is as following figure 4.

$S_t = \emptyset$
for m = 1 to M **do**
  $CDF(m) = \sum_{i=1}^{m} w_t^i$
end for
$r_0 = rand\{0 \leq r_0 \leq \frac{1}{M}\}$
for m=1 to M **do**
  $i \;=\; \min j : CDF(j) \geq r_0 + \frac{m-1}{M}$
  $S_t \;=\; S_t \cup < x_t^i, \frac{1}{M} >$
end for
return S_t

Fig. 4. Systematic Re-Sampling.

3. **pfCorrect**: If assignment is successful, one track's estimated position will be correct. That mean all particles of this track will be set to the same position. The position is the centroid of matched detected object.

4 Experimental results

We test the program by using surveillance videos. Following figure 5 shows how

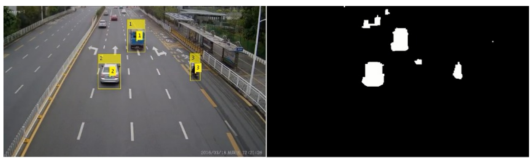

Fig. 5. Result when particles centered on the center of boundary boxes

the program runs. There are some particles centered on the center of boundary boxes. Due to the successful assignment and pfCorrect function ,the 3 track's particle filters' particles are maintained as centroids.

Following figure 6 shows other situation. In the binary picture, we can find that object labeled 14 and 15 are connected, which leads to one of these 2 objects disappeared because these 2 objects are considered as one and detect Objects function only returns one centroid for them. In this case, the returned centroid is closer to label 15 track, so label 14 track is not assigned. The position is predicted without correction, so the particles spread out.

In order to find a better method for tracking, we test both kalman filter and particle filter with different number of particles

Fig. 6. Result when particles do not centered on the center of boundary boxes

4.1 Execution time

We test the program by using 3 samples. For 3 samples, we test the program with 3 conditions: 1) 100 particles;2) 10 particles;3) kalman filer. Result is shown as following table. We can find that the kalman filter runs fastest and particle filter with 100 particles runs slowest.

Method	Execution time		
	1	2	3
100 Particles	103.412583	27.1828	68.53584
10 Particles	66.642374	19.46634	49.90434
Kalman Filter	47.794069	17.59786	39.98926

Fig. 7. Execution time.(unit: seconds)

4.2 Prediction when targets lost

When the targets are lost , boundary boxes are given by estimator of each tracks. In this situation, we find particle filter performs better than kalman filter. Since kalman filter considers all objects move in a straight line with constant speed. The prediction made by it may be wrong. To explain it, we use the video 'atrium.mp4' form original code[1].

Figure 8 shows how kalman filter works. When the person is hided by the tree, the predicted position goes straightly. The direction is calculated by last object detections. Finally, when the person shows up again, the predicted position goes too far away form the the person, so they fail to be assigned. The person is considered as a new one and labeled as 4.

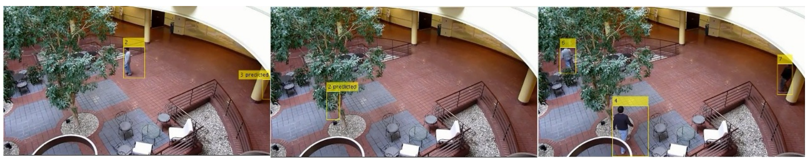

Fig. 8. Prediction by Kalman filter

Figure 9 shows how particle filter works. When the same situation happened, the predicted position can roughly follow the person. From figure 10 ,we can see the mask of the second picture. The Detect Objects function can find some moving object, and then particle filter spread out, and its particles center to this connected blocks due to re-sample. Finally, when the person shows up again, the predicted position is not too far and they are successfully assigned.

Fig. 9. Prediction by Particle filter

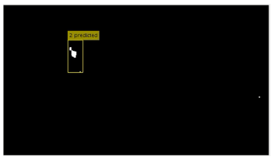

Fig. 10. Prediction mask by Particle filter

When the targets' movement model are not linear and they are not detected clearly for some reason, particle filter predict position better than kalman filter.

4.3 Prediction when targets run as closely opposite direction

When this situation happens, as figure 6 shows, 2 objects are considered as one. So one of them need a predicted position. Kalman filter may have a good result here, because targets run as opposite direction and Kalman filter will give a right direction estimation. However, particle filter may produce tracking errors here. Due to the diffusion, some particles may move to the object close to it. Finally, predicted position will be greatly effected by the object which runs closely as opposite direction.

5 Summary and Conclusions

In this project, we developed a particle filter for a program which can detect, label and track moving objects in a video from a surveillance camera. Program generally achieves goal of detecting, labeling and tracking moving objects, although it may produce tracking errors in some conditions. We compare kalman filter and particle filter in some aspects. Kalman filter perform faster and has better prediction while objects' movement motion is linear. Particle filter give a following result when the target is not detected well. If we can apply particle filters a good motion model, particle filter may perform better in some condition.

References

1. Motion-Based Multiple Object Tracking, `https://www.mathworks.com/help/vision/examples/motion-based-multiple-object-tracking.html`
2. Da-Peng Bai, Bing-Bing Lee: Based-on Particle Filter For Vehicle Detection And Tracking In Digital Video : Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 12-15 July 2008
3. Francois Bardet and Thierry Chateau: MCMC Particle Filter for Real-Time Visual Tracking of Vehicles : Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems Beijing, China, October 12-15, 2008
4. P.L.M. Bouttefroy, A. Bouzerdoum, S.L. Phung , A. Beghdadi : Vehicle Tracking using Projective Particle Filter : 2009 Advanced Video and Signal Based Surveillance
5. Mingxiu Lin, Feng Pan, Jingjing Wang, Shuai Chen : Vehicle tracking based on particle filter using double features : 2009 International Conference on Information Engineering and Computer Science
6. Yang Xu, Chengdong Wu , Xuedong Zhang , Ying Zhao : Tracking Moving Vehicles Using Particle Filtering and Level Set : 2011 Chinese Control and Decision Conference (CCDC)
7. Christiano Bouvi, Jacob Scharcanski , Pablo Barcellos , Fabiano Lopes Escouto : Tracking and Counting Vehicles in Traffic Video Sequences Using Particle Filtering : Instrumentation and Measurement Technology Conference (I2MTC), 2013 IEEE International
8. Ashvini Kulkarni, Manasi Vargantwar , Sujata Virulkar : Video Based Tracking and Optimization Using Mean-Shift,Kalman Filter and Swarm Intelligence : 2015 International Conference on Computing Communication Control and Automation
9. Kenneth Tze Kin Teo, Renee Ka Yin Chin, N.S.V. Kameswara Rao, Farrah Wong, Wei Leong Khong: Vehicle Tracking Using Particle Filter for Parking Management System : 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology
10. S.Srilekha, G.N.Swamy, A.Anudeep Krishna: A Novel Approach for Detection and Tracking of Vehicles using Kalman Filter : 2015 International Conference on Computational Intelligence and Communication Networks
11. Indrabayu, Rizki Yusliana Bakti, Intan Sari Areni, A.Ais Prayogi: Vehicle Detection and Tracking using Gaussian Mixture Model and Kalman Filter: Computational Intelligence and Cybernetics (CYBERNETICSCOM), 2016 International Conference
12. Lab 2