AN IMPROVED OCCLUSION HANDLING FOR APPEARANCE-BASED TRACKING

Gwo-Cheng Chao[1], *Shyh-Kang Jeng*[1], *Shung-Shing Lee*[2]

[1]Graduate Institute of Networking and Multimedia, National Taiwan Univ., Taipei, Taiwan
[2]Department of Electrical Engineering Ching Yun University, Jung-Li, Taiwan

ABSTRACT

The object occlusion is one serious issue in object tracking, especially when objects merge or split. The tracker may fail if it has no adaptability to such variations. In this paper, we present an improved solution of the appearance-based tracking for occlusion handling. The proposed method first recognizes the motion situations (merging and splitting of the moving objects), and then applies different template finding approach on each motion to get an association between detected blobs and targets. By this way, the accuracy of the data association can be improved when object occlusion occurs. Experiments were conducted to compare our results with those by other popular tracking algorithms. The proposed method is found robust for short-time complete occlusion and partial occlusion.

Index Terms— Object tracking, Occlusion handling, Appearance tracking, Motion recognition, Template finding

1. Introduction

Object tracking is to identify and track all relevant moving objects in a scene and to generate exact one track per object in computer vision. To track multiple targets in a video, the blob-based tracking approaches are widely used [1]. This kind of tracking approaches is preceded by motion detection that provides the locations of blobs (connected regions of moving pixels) to create a candidate list of observations of the current active scene objects, and then associates the blobs with known target trajectories. However, several problems need to be addressed to prevent the track being lost or taken away by other objects when partial or fully occlusion occurred. The interactions of objects will cause the close objects to generate a single merged object, such that the visual features of the occluded objects can not be distinguished and tracked separately.

To address this problem, we need a data association algorithm that can deal with merging and splitting objects. Joint Probabilistic Data Association Filter (JPDAF) [2] proposed by Bar-Shalom et al., can address situations where two targets are merged. However, the JPDAF algorithm performs data association for a fixed number of objects tracked over two frames. Serious errors will take place if there is a change in the number of objects. Another classical data association approach is the multiple hypothesis tracking (MHT) proposed by Reid [3]. For solving the problems of merging and splitting objects, the MHT has been extended [4]. The common problem of the MHT method is that it is computationally exponential both in memory and time. On the other hand, for occlusion handling, Yang et al. [5] uses the relationship between detected blobs and tracks to recognize the occlusion event first, and then associate the detected blobs to the tracks when objects split based on the feature information. The advantage is that it can know when the occlusion event occurs. This prior information is useful for object occlusion handling. However, it does not discriminate the moving objects during occlusion, and increases the chance of failures. On the contrary, Senior [6] proposed an appearance-based tracking algorithm that can track objects during occlusions by a probabilistic pixel reclassification algorithm. However, the accuracy of the prediction step will influence the appearance model updates and the tracking accuracy, especially when objects merge and split.

In this paper, we propose a tracking method that combines the concepts proposed by Yang et al. [5] and Senior [6]. We first recognize the motion situations (objects merging and splitting), and then apply different template finding approach on each motion. For resisting the non-rigidity of the object's motion, we model the observed appearance of the moving object by the appearance model [6]. By this way, the accuracy of the data association can be improved in spite of objects merging or splitting. For convenience, we do not consider the effect of noisy foreground extractions, such as those due to the object fragmentation. Moreover, the proposed framework can also be implemented for real time purpose by several real-time techniques (e.g., a faster algorithm, implementation optimization, and an embedded system).

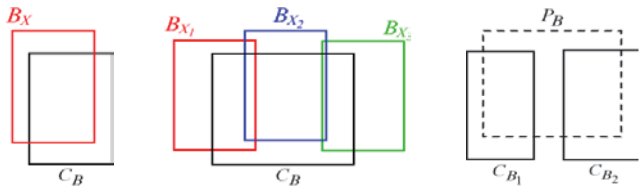
The rest of this paper is organized as follows. In Section II, we describe the proposed method. In Section III, experimental results and discussions are presented. Finally, some conclusions are given in Section V.

2. TRACKING WITH OCCLUSION HANDLING

The proposed method consists of two steps. For each video frame, the motion situations of the moving objects is determined first, and then different template finding

978-1-4577-1302-6/11/$26.00 ©2011 IEEE

approaches are executed according to the motion situation. The details are given below.



(a) (b) (c)

Fig.1. Concepts of three motion situations. (a) Single object situation: One current blob C_B covers one tracking box B_X. (b) Occlusion situation: One current blob C_B covers multiple tracking boxes B_{X_1}, B_{X_2} ,and B_{X_3}. (c) Object-splitting situation: One previous blob P_B covers multiple current blobs C_{B_1} and C_{B_2}.

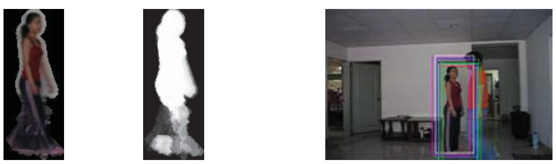

(a) (b) (c)

Fig. 2. (a) The color model of a woman. (b) The probability mask of the woman. (c) Some particles are spread on the woman using the appearance model (color model (a) and probability mask (b)) as the observation model.

A. Motion-situation recognition

We use tracking boxes and the detected blobs (represented by bounding boxes) extracted from the previous frame and the current frame to recognize three motion situations: a single object (single object situation), multiple objects merging together (occlusion situation), and splitting objects (object-splitting situation). Fig. 1 shows concepts of the three tracking cases. This additional process is based on blobs and does not increase the processing time much. This technique can detect occlusion situations easily and update the appearance model at different speeds to prevent erroneous updating. In [6], the three motion situations are not distinguished and the updating processing is the same in all situations. Besides, unlike the method in [5], by our approach the splitting event is detected by the blobs extracted from the previous frame and the current frame. This is simpler and can reduce the error produced by unstable tracking results.

B. Template finding in different motion situations

For different motion situation, we apply different template finding methods to reduce the chance of getting a broken trajectory. On the other hand, we model the template of the observed appearance of the moving object by the appearance model [6]. The appearance model contains an RGB color model (e.g. Fig.2 (a)) and a probability mask (e.g. Fig.2 (b)). The corresponding methods of treating these three motion situations are:

1) Single object situation

This case is normal and simple. Only one moving object is tracked. To update the appearance model more accurately, we propose using the three-step search algorithm by Koga et al. [7] to predict the maximum likelihood location. Moreover, the update rates (α and λ in [6]) of the appearance model are small in this situation to obtain a newer appearance model and the initial probability of the new foreground pixel will be also higher. In our system, we set the update rates and the initial probability of the new foreground pixel as ($\alpha = 0.7$, $\lambda = 0.7$) and 0.4, respectively.

2) Occlusion situation

To predict the location of each moving object well, we involve the particle filter and the three-step search techniques. We use the appearance model created in the previous frame as the prior of the particle filter. The likelihood of the appearance model between the created appearance model and the current extracted pattern will be regarded as the observation probability for each particle to estimate their similarity. For each tracker on the current frame, a few number of particle filters will be adopted to predict the locations of the tracking results roughly (e.g. Fig. 2 (c)). The three-step search algorithm will be used to refine the locations obtained from the particle filters. On the other hand, the update rates (α and λ in [6]) of the appearance model are chosen larger to prevent erroneous updating from the wrong tracking results. The initial probability of the new foreground pixel will be also smaller in this situation. In our system, we set the update rates and the initial probability of the new coming foreground pixel as ($\alpha = 0.98$, $\lambda = 0.98$) and 0.1, respectively. Compared with [6], which searches for the maximum likelihood over a small search regions around the centroid in the occlusion situation (first column, Fig. 3), our method can find the maximum likelihood of the created appearance model more accurately (second and third columns, Fig. 3).

Fig. 3. Examples of finding the maximum likelihood of the created appearance model in the occlusion situation. The found maximum likelihood is colored with green. Top row: finding the maximum likelihood of the created appearance model of the man in the left side. Second row: finding the maximum likelihood of the created appearance model of the man in the right side. First column: found by searching over a small search regions around the centroid. Second column: found by the particle filters. Third column: found by the particle filters and refined by the three-step search

algorithm.

3) Object-splitting situation

In this case, the object re-identification is required. Here, we use particle filters of each tracker on each blob to find out the blob a tracker being associated with. Additionally, the weighting of the motion direction and the velocity of each tracker will be also considered and added with the likelihood weighting of each particle. Finally, the three-step search algorithm is applied to refine the predicted location determined by the particle filter. The parameters of the update rates and the initial probability of the new foreground pixel are the same as the occlusion situation to prevent erroneous updating.

3. RESULTS AND DISCUSSIONS

In the following, we show some results of our proposed tracking approach, compared with two other popular tracking algorithms (particle filtering tracking [8], and appearance model tracking [6]). The mean sift tracking algorithm [9] is not considered due to its weakness in dealing with object occlusion.

We have applied the proposed method to three video clips captured by stationary cameras. The frame rate is about 25-30 frames/second and the frame size is 320x240 pixels. For background training, each video clip is with some background frames in the beginning. The first and the second testing video clips are captured from an outdoor square close to a building and have 1600 frames and 1100 frames, respectively. The third testing video clip is captured from a road and has 1100 frames. On the other hand, to show the ability of occlusion handing of the tracking algorithm, there are some short-time complete occlusions occurred in each video. The numbers of short-time complete occlusions of these testing video clips are 7 in the first, and 5 in the second and third.

The comparison is based on three measures describe as the following:

1) Tracker missing rate

This is used to measure the number of frames that one or more tracking failures occur (loss of the tracker or that the tracker is taken away by other objects or background noise). We define two kinds of tracker missing rate. The first is the average tracker missing rate calculated from all video frames a tracking algorithm processed. It is defined as $T_{miss} = O_{f_{miss}} / T_f$, where T_f is the total number of frames that a tracking algorithm processed, $O_{f_{miss}}$ is the number of frames (counted in the range of T_f) that some tracker missings occur. It should be noted that once a tracker missing occurs in one frame, it may continue missing in subsequent frames and the errors will be also counted in $O_{f_{miss}}$. The second kind of tracker missing rate is defined as $P_{miss} = O^N_{f_{miss}} / T^N_f$, to measure the average tracker missing rate of a tracking algorithm for frames with a specified number (N) of moving objects in one frame. T^N_f is the total number of frames (each frame contains exactly N moving objects) that a tracking algorithm executed , and $O^N_{f_{miss}}$ is the number of frames (counted in the range of T^N_f) that some tracker missing occur. In a surveillance video, the number of the moving objects varies in each frame. By measuring the tracker missing error for different number of moving objects, we can find the tracking capacity of the tracking algorithm. In our testing cases, the range of the number of the moving objects in one frame (N) is from one to four. For example, the numbers of the moving objects in the top row of Fig.4 are 2, 3, 3, and 4.

2) Tracker error rate

This is used to measure the number of frames that a tracker changes to track other moving objects. These error frames will be accumulated if the trackers continue to track the wrong moving object in the next frame. Due to that the trajectory of each moving object is an important content in our application, error tracking will generate error trajectory of the moving object. We also define two kinds of tracker error rate: $T_{error} = O_{f_{error}} / T_f$ and $P_{error} = O^N_{f_{error}} / T^N_f$, where $O_{f_{error}}$ is the number of frames (counted in the range of T_f) that some tracker errors occur, and $O^N_{f_{error}}$ is the number of frames (counted in the range of T^N_f) that some tracker errors occur. The measures T_{error} and P_{error} are the average tracker error rate of each tracking algorithm with the number of moving objects unspecified and specified in one frame, respectively.

3) Correctness of Tracker numbers

This is used to observe the total number of trackers a tracking algorithm generates and to check if the number of generated trackers of a tracking algorithm matches the number of moving objects in the testing video.

In the experiments, we applied these measures and tested the tracking algorithms P (particle filtering tracking [8]), A (appearance model tracking [6]), and O (our tracking algorithm) on the selected video clips V1 (the first video clip), V2 (the second video clip), and V3 (the third video clip).

Table I shows the results of T_{miss} and T_{error} of each tracking algorithms. Table II shows the results of P_{miss} and P_{error} on different number of moving objects occurring in each frame (*N-Object*). The second video clip (V2) does not have frames containing four objects, thus the values of P_{miss} and P_{error} are not shown. Table III shows the number of moving objects appeared in the video clip and detected by each tracking algorithm. As we can see in Table I and Table II, our proposed method has the lowest error rates of the tracker missing and the tracker error. Table III shows that by our algorithm the detected number of moving objects matches the number of moving objects in each testing video exactly. Through these results, we can say that our proposed method is more stable, at least for our test cases.

Figures 4, 5, and 6 show the partial tracking results of these examples and indicate the occlusion situation of the moving objects.

TABLE I

T_{miss} AND T_{error} BY THREE TRACKING ALGORITHMS

	P		A		O	
	T_{miss}	T_{error}	T_{miss}	T_{error}	T_{miss}	T_{error}
V1	0.0056	0	0.0145	0.5250	0.0081	0
V2	0.0813	0.3145	0.0086	0.5140	0	0.0225
V3	0.0029	0.2747	0.0050	0.2623	0	0

TABLE II

THE RESULTS OF P_{miss} AND P_{error} ON DIFFERENT OBJECT NUMBER N.

		1-Object		2-Object		3-Object		4-Object	
		P_{miss}	P_{error}	P_{miss}	P_{error}	P_{miss}	P_{error}	P_{miss}	P_{error}
V1	P	0.0511	0	0.0253	0	0.1157	0	0	0
	A	0	0	0.0073	0.4095	0.0319	0.8962	0	1
	O	0	0	0	0	0.0229	0	0	0
V2	P	0	0.4887	0.0018	0.2476	0.5826	0.2283	-	-
	A	0	0.0279	0.0134	0.5230	0.0084	1	-	-
	O	0	0	0	0.0164	0	0.1111	-	-
V3	P	0	0	0.0032	0.6524	0.0162	0.5853	0	1
	A	0	0	0.0107	0.5857	0.0087	0.7017	0.0588	1
	O	0	0	0	0	0	0	0	0

TABLE III

NUMBER OF OBJECTS DETECTED BY EACH TRACKING ALGORITHM

	# of moving objects in video	P	A	O
V1	8	9	15	8
V2	7	7	15	7
V3	7	9	11	7

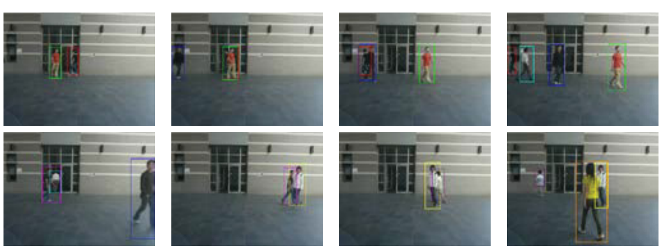


Fig.4. Some tracking results of the first testing video segment. The occurrence order of each frame is presented from left to right and top to down. The bounding box of each moving object has been drawn with a color different from others. There are eight objects (eight people) moving in the scene.

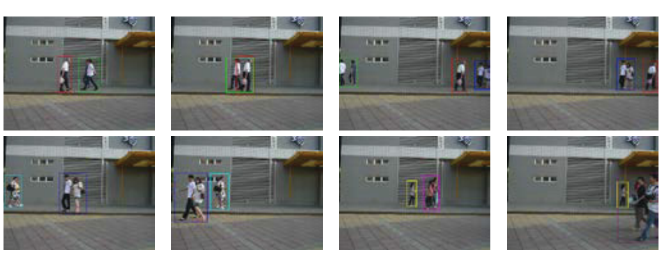


Fig.5. Some tracking results of the second testing video segment. The occurrence order of each frame is presented from left to right and top to down. There are seven moving objects will be detected.

Fig.6. Some tracking results of the third testing video segment. The occurrence order of each frame is presented from left to right. There are seven moving objects moving in the scene.

4. Conclusions

We have proposed a method that is efficient to handle object occlusion. The method is based on recognizing the motion situations of the moving object first, and then applies different template finding approaches to obtain good association between the detected blobs and targets. For resisting the non-rigidity of the object's motion, the observed appearance of the moving object is modeled by the appearance model. Experiment results show that this method is robust for short-time complete occlusion and partial occlusion.

In the future, we will try to formulate the problem of the object fragmentation in this framework. Furthermore, the multiple hypotheses tracking (MHT) approach can also be incorporated to handle long occlusion situation.

5. REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM J. Comput. Surv.*, vol. 38, no. 4, Dec. 2006.

[2] K. C. Chang and Y. Bar-Shalom, "Joint Probabilistic Data Association for Multitarget Tracking with Possibly Unresolved Measurements and Maneuvers," *IEEE Trans. Autom. Control*, vol. 29, no. 7, pp. 585–594, Jul. 1984.

[3] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[4] Y. Maa, Q. Yub ,and I. Cohena, "Target tracking with incomplete detection," *J Comput. Vis. Image Understanding*,vol.113, no. 4, pp. 580-587, Apr. 2009.

[5] T.Yang, S.Z.Li,Q.Pan, and J. Li, "Real-time Multiple Objects Tracking with Occlusion Handling in Dynamic Scenes," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2005, pp. 970-975.

[6] A. Senior, "Tracking people with probabilistic appearance models," in *Proc. Int. Workshop Perf. Eval. of Tracking and Surveillance*, 2002, pp. 48–55.

[7] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. Nat. Telecommun. Conf.*, 1981, pp.G5.3.1–G5.3.5.

[8] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "An adaptive color-based particle filter," *J Image Vis. Comput.*, vol. 21, no.1, pp. 99-110, Jan. 2003.

[9] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2000, pp. 142-149.