TRACKING DEFORMABLE OBJECTS AND DEALING WITH SAME CLASS OBJECT OCCLUSION

René Alquézar
Institut de Robòtica i Informàtica Industrial (IRI), Universitat Politècnica de Catalunya-CSIC
c/ Llorens i Artigas 4-6, 08028, Barcelona, Spain

Nicolás Amézquita, Francesc Serratosa
DEIM, Universitat Rovira i Virgili, Av. dels Països Catalans 26, 43007, Tarragona, Spain

Keywords: Object tracking in video sequences, Motion and tracking, Object crossing and occlusion.

Abstract: This paper presents an extension of a previously reported method for object tracking in video sequences to handle the problems of object crossing and occlusion by other objects in the same class that the one followed. The proposed solution is embedded in a system that integrates recognition and tracking in a probabilistic framework. In a recent work, a method to approach the object occlusion problem was proposed that failed when the object crossed or was occluded by another object of the same class. Here we present an attempt to overcome this limitation and show some promising results. The method is based on the assumption that when two objects cross each other there is not a brusque change of the trajectories. Our system uses object recognition results provided by a neural net that are computed from colour features of image regions for each frame. The location of tracked objects is represented through probability images that are updated dynamically using both recognition and tracking results. From these probabilities and a prediction of the motion of the object in the image, a binary decision is made for each pixel and object.

1 INTRODUCTION

Over recent years, much research has been developed to solve the problem of object tracking under occlusions, because, in real-world tracking, a target being partly or entirely covered by other objects for an uncertain period of time is common. Occlusions pose several challenges to object tracking systems such as determining the beginning and the end of an occlusion and predicting the location of the target during and at the end of the occlusion.

Determining occlusion status is very hard for the trackers, nevertheless, various approaches that analyze occlusion situations have been proposed. The most common one is based on background subtraction (Senior *et al*., 2006). Although this method is reliable, yet it only works with a fixed camera and a known background. Other approaches are based on examining the measurement error for each pixel (Nguyen, 2004; Zhou, 2004). The pixels that their measurement error exceeds a certain value are considered to be occluded. A mixture of distributions is used in (Jepson *et al*., 2003) to model the observed value of each pixel, where the occluded pixels are characterized by having an abrupt difference with respect to a uniform distribution. On the other hand, contextual information is exploited in (Ito and Sakane, 2001; Hariharakrishnan, 2005).

Determining the re-emergence of the target and recapture its position after it is completely occluded for some time is the other main challenge. Setting a similarity threshold is one method, yet the optimal threshold value is difficult to determine. This problem is circumvented in (Nguyen, 2004), where the image region that matches the best with the template over a prefixed duration is assumed to be the reappearing target. Recently, new object tracking methods that are robust to occlusion have been reported with very promising results (Pan and Hu, 2007; Zhu *et al*., 2008).

When the occluder has similar features than the target, both are mistaken in the re-emergence process, even more if we are dealing with

Alquezar R., Amezquita N. and Serratosa F. (2009).
TRACKING DEFORMABLE OBJECTS AND DEALING WITH SAME CLASS OBJECT OCCLUSION.
In *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*, pages 590-594
DOI: 10.5220/0001791605900594

deformable objects of changing shape. This paper presents an extension of a previously reported method for object tracking in video sequences (Amézquita, 2007; Amézquita, 2008) to handle the problems of object tracking in the re-emergence process when the target is deformable and it is occluded by other objects in the same class that the one followed. The extended tracking method is embedded in a system that integrates recognition and tracking in a probabilistic framework. Our system uses object recognition results provided by a neural net that are computed from color features of image regions for each frame. The location of tracked objects is represented through probability images that are updated dynamically using both recognition and tracking results. The prediction of the object's apparent motion and size takes into account the cases of occlusion entering, full occlusion and occlusion exiting, which are detected automatically.

The rest of the paper is organized as follows. A formal description of our probabilistic approach for object recognition and tracking is given in Section 2. The new part of the tracking method that deals with same-class object crossing is described in Section 3 and the experimental results obtained are shown in Section 4. Conclusions are presented in Section 5.

2 A PROBABILISTIC APPROACH FOR OBJECT RECOGNITION AND TRACKING

Let us assume that we have a sequence of 2D color images $I^t(x,y)$ for $t=1,\ldots,L$, and we want to track in the sequence N objects of interest of different types (associated with classes $c=1,\ldots,N$, where $N\geq 1$ and a special class $c=N+1$ is reserved for the background). Furthermore, let us assume that the initial position of each object is known and represented by N binary images, $p_c^0(x,y)$, for $c=1,\ldots,N$. We want to obtain N sequences of binary images $T_c^t(x,y)$ that mark the pixels belonging to each object in each image. We can initialize these tracking images from the given initial positions of each object, i.e. $T_c^0(x,y)= p_c^0(x,y)$.

In our approach (Amézquita, 2008), we divide the system in three main modules. The first one performs object recognition in the current frame (static recognition) and stores the results in the form of probability images. This can be achieved by using a classifier (e.g. a neural network) that has been trained previously to classify image regions of the same objects using a different but similar sequence of images, where the objects have been segmented and labeled. Hence, we assume that the classifier is now able to produce a sequence of class probability images $Q_c^t(x,y)$ for $t=1,\ldots,L$ and $c=1,\ldots,N+1$, where the value $Q_c^t(x,y)$ represents the estimated probability that the pixel (x,y) of the image $I^t(x,y)$ belongs to the class c.

In the second module (dynamic recognition), the results of the first module are used to update a second set of probability images, P_c, with a meaning similar to that of Q_c but now taking into account as well both the recognition and tracking results in the previous frames through a dynamic iterative rule (Amézquita, 2007). We store and update $N+1$ probability images $P_c^t(x,y)$, where the value $P_c^t(x,y)$ represents the probability that the pixel (x,y) in time t belongs to the tracked object of class c (for $c=1,\ldots,N$) or to the background (for $c=N+1$).

Finally, in the third module (tracking decision), tracking binary images are determined for each object from the current dynamic recognition probabilities, the previous tracking image of the same object and some other data that contribute to provide a prediction of the object's apparent motion in terms of translation and scale changes as well as to handle the problems of object occlusion and crossing. The tracking images $T_c^t(x,y)$ for the objects ($1\leq c\leq N$) can be calculated dynamically using the pixels probabilities $p^t(x,y)$ according to a decision function d that involves additional arguments and results:

$$\left\langle T_c^t(x,y),\hat{T}_c^t(x,y),\vec{m}_c^t,O_c^t,C_c^t,A_c^t,\varepsilon_c^t,\delta_c^t\right\rangle = d\begin{pmatrix} p^t(x,y),T_c^{t-1}(x,y),\hat{T}_c^{t-1}(x,y), \\ \vec{m}_c^{t-1},O_c^{t-1},O_c^{t-2},C_c^{t-1},C_c^{t-2}, \\ A_c^{t-1},A_c^{t-2},\varepsilon_c^{t-1},\delta_c^{t-1} \end{pmatrix} \qquad (1)$$

where we distinguish between the *a posteriori* tracking image $T_c^t(x,y)$ and an *a priori* prediction $\hat{T}_c^t(x,y)$, which is robust (to some extent) to occlusion; an occlusion flag O_c^t is determined at each step and the two previous flags O_c^{t-1} and O_c^{t-2} help to know whether the object is entering or exiting an occlusion; the object mass center C_c^t and area A_c^t needed for estimating the apparent motion are measured either from $T_c^t(x,y)$ or $\hat{T}_c^t(x,y)$ depending on whether the object is visible or occluded; ε_c^t and δ_c^t are two adaptive parameters that control the level of uncertainty in the *a priori* prediction $\hat{T}_c^t(x,y)$, since the uncertainty grows with the duration of an occlusion; finally, $\vec{m}_c^t$ is a movement weighted average vector that represents the past trajectory direction of the tracked object.

A specific tracking decision function *d* was proposed in (Amézquita, 2008), which is able to cope with the object crossing and occlusion problems if the occluding objects are from different class that the tracked objects. Experimental results showed the effectiveness of the method except when the target object was occluded by an object with similar appearance (same class from the static-recognition module). In the next section, we describe a new method to deal with this problem.

3 DEALING WITH SAME-CLASS OBJECT CROSSING

In order to circumvent the same-class crossing problem, we carry out a post-processing step that removes from both $T_c^t(x,y)$ and $\hat{T}_c^t(x,y)$ some possible artifacts or distracters (setting some initially 1-valued pixels to zero). In fact, we only do that in the case that $T_c^t(x,y)$ contains non-connected components. This typically occurs when the same-class crossing or occlusion has just finished and the tracking method is misled to follow both the object and the distracter. Then, we need to choose one component and discard the other(s). The movement vector $\vec{m}_c^t$ is used for that purpose in the following way. Let C_{ci}^t be the mass center of the *i*-th connected component and define an associated movement vector $\vec{z}_{ci}^t = C_{ci}^t - C_c^{t-1}$ for each component. We select the component *i* whose vector has the maximal projection (normalized by its norm) to the previous movement vector and set the pixels of the rest of components to zero in the tracking images $T_c^t(x,y)$ and $\hat{T}_c^t(x,y)$. This is, we select *i* such that

$$i = \arg\max \left\{ \frac{\langle \vec{z}_{ci}^t, \vec{m}_c^{t-1} \rangle}{\left\| \vec{z}_{ci}^t \right\|} \right\} \quad (2)$$

which is the one for which $\vec{z}_{ci}^t$ is the most collinear vector with respect to $\vec{m}_c^{t-1}$. The movement weighted average vector $\vec{m}_c^t$ is updated afterwards as follows:

$$\vec{m}_c^t = \begin{cases} \left(\vec{v}_c^t + (t-1)\vec{m}_c^{t-1}\right)/t & \text{if } t < 1/\beta \\ \beta\,\vec{v}_c^t + (1-\beta)\vec{m}_c^{t-1} & \text{if } t \geq 1/\beta \end{cases} \quad (3)$$

where β is a positive parameter between 0 and 1, and $\vec{v}_c^t$ is the current movement defined by $\vec{v}_c^t = \vec{z}_{ci}^t$ (being *i* the selected component if many or the unique one). Note that the second row in (3) is a typical moving average computation, while the first row denotes a simple average for the starting steps, and both give the same result for $t=1/\beta$.

4 EXPERIMENTAL RESULTS

Several video sequences have been employed for an experimental validation of the proposed approach. They all show an office scene where two blue balls are moving on a table and one occludes temporally the other one. Hence, we defined $N=1$ objects of interest: the blue ball to track. A test sequence is at http://deim.urv.cat/~francesc.serratosa/test.avi. A similar but different sequence was used for training a neural network to discriminate between blue balls and typical sample regions in the background (at http://deim.urv.cat/~francesc.serratosa/bluetraining.avi).

All images in the sequences were segmented independently using the EDISON implementation of the mean-shift segmentation algorithm (Comaniciu and Meer, 1999), code available at http://www.caip.rutgers.edu/riul/research/code.html. The local features extracted for each spot were the RGB colour averages and variances. For object learning, spots selected through ROI (region-of-interest) windows in the training sequence were collected to train a two-layer perceptron using backpropagation. The trained network was applied to estimate the class probabilities for all the spots in the test sequences. The spot class probabilities were replicated for all the pixels in the same spot.

For object tracking in the test sequences, ROI windows for the blue ball to track were only marked in the first image to initialise the tracking process and the dynamic class probabilities.

The results for the test sequences were stored in videos where each frame has a layout of 2 x 3 images with the following contents: the top left is the image segmented by EDISON; the top middle is the image of static probabilities given by the neural net for the current frame; the top right is the *a priori* binary tracking image; the bottom left is the image of dynamic probabilities; the bottom right is the *a posteriori* binary tracking image; and the bottom middle is a labelled image where yellow pixels correspond to pixels labelled as "certainly belonging to the object", light blue pixels correspond to pixels initially labelled as "uncertain" but with a high dynamic probability, dark blue pixels correspond to pixels labelled as “uncertain” and with a low probability, dark grey pixels correspond to pixels labelled as “certainly not belonging to the object” but with a high probability (false detections) and the rest are black pixels with both a low probability and a “certainly not belonging to the object” label.

Two experiments have been performed on the test sequence deim.urv.cat/~francesc.serratosa/test.avi depending on the initialisation of the tracking. In this sequence, two blue balls are moving and they overlap during some frames. In test 1, the tracking was initialised at the right ball and in test 2, the tracking was initialised at the left ball. The static recognition module (neural net) considers that both balls belong to the same class. In both sequences, the temporal overlapping is correctly managed by the method since the ball is well relocated after exiting the occlusion. Our results are attainable at deim.urv.cat/~francesc.serratosa/test_results_1.avi and deim.urv.cat/~francesc.serratosa/test_results_2.avi.

Figure 1 shows one of the first frames of test 1. Although the neural net recognises two objects (up middle image), the tracking module discards the non-target object (up and down right images).

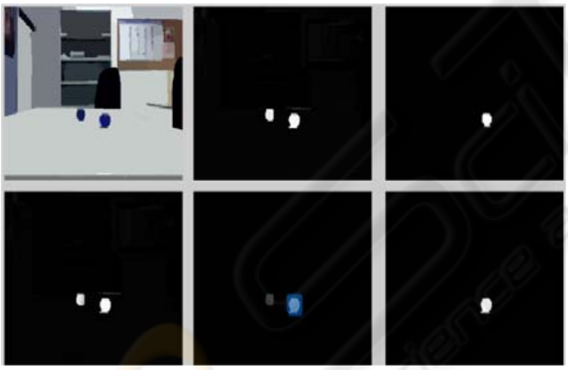

Figure 1: Results for one of the frames in test 1.

In order to compare the performance of our method (referred to as PIORT, for *Probabilistic Integrated Object Recognition and Tracking*) in front of other previously reported tracking methods, the two experiments were also carried out applying the following methods (their code was downloaded from the VIVID tracking evaluation web site www.vividevaluation.ri.cmu.edu):

1. Template Match by Correlation (TMC), which refers to normalized correlation template matching (Comaniciu *et al*., 2003).
2. Basic Meanshift (BM) (Comaniciu *et al*., 2000; Comaniciu and Meer, 2002).
3. Histogram Ratio Shift (HRS) (Collins et al., 2005).
4. Variance Ratio Feature Shift (VRFS) (Collins and Liu, 2005).
5. Peak Difference Feature Shift (PDFS) (Collins and Liu, 2005).
6. Graph-Cut Based Tracker (GCBT) (Bugeau and Pérez, 2008).

From the tracking results of all the tested methods, two evaluation metrics were computed for each frame: the spatial overlap and the centroid distance (Yin *et al*., 2007). The spatial overlap is defined as the overlapping level $A(GT_k,ST_k)$ between the ground truth GT_k and the system track ST_k in a specific frame k:

$$A(GT_k, ST_k) = \frac{\text{Area}\left(GT_k \cap ST_k\right)}{\text{Area}\left(GT_k \cup ST_k\right)} \tag{4}$$

and $Dist(GTC_k, STC_k)$ refers to the Euclidean distance between the centroids of the ground truth (GTC_k) and the system track (STC_k) in frame k. Naturally, the larger the overlap and the smaller the distance, the better the accuracy of the system track.

Since the centroid distance can only be computed if both GT_k and ST_k are non-null, a failure ratio FR was defined as the number of frames in which either GT_k or ST_k was null (but not both) divided by the total number of frames. This third evaluation measure is especially sensitive to occlusion cases.

Tables 1 and 2 present respectively the results (mean ± std. deviation) of the two former evaluation measures for test 1 (tracking right ball) and test 2 (tracking left ball). As can be seen, our tracking method PIORT outperformed the rest in both tests (except in the case of the centroid distance in test 1, where it was slightly under the performance of the VRFS tracker).

Table 1: Tracking evaluation results for test 1.

Test 1 (tracking right ball in test sequence)		
Tracking Method	Spatial Overlap	Centroid Distance
TMC	0.56±0.10	5.07±2.07
BM	0.60±0.06	3.19±1.21
HRS	0.46±0.11	6.03±2.05
VRFS	0.66±0.07	1.15±0.47
PDFS	0.63±0.10	2.01±0.94
GCBT	0.64±0.18	13.20±52.52
PIORT	0.84±0.09	1.38±1.39

Table 2: Tracking evaluation results for test 2.

Test 2 (tracking left ball in test sequence)		
Tracking Method	Spatial Overlap	Centroid Distance
TMC	0.22±0.27	44.34±52.24
BM	0.23±0.29	42.51±50.42
HRS	0.25±0.31	44.93±51.96
VRFS	0.28±0.35	42.82±52.62
PDFS	0.50±0.30	36.27±86.95
GCBT	0.20±0.27	70.69±68.80
PIORT	0.60±0.23	3.94±4.98

Regarding the failure ratio, a value of zero was obtained for all methods except *FR*=0.09 for GCBT in test 1 and *FR*=0.28 for PDFS tracker in test 2.

5 CONCLUSIONS

A previously proposed method for object tracking, which was integrated in a probabilistic framework for object recognition and tracking in video sequences (Amézquita, 2007; Amézquita, 2008), has been extended in this work to deal with same-class object crossing and occlusion. The new method is able to select and track only the target object after it crosses or is occluded by another object which is recognised as belonging to the same class. However, this has been achieved under the assumption that the trajectory of the target object is relatively stable in the preceding part of the sequence. The method may fail in a large changing motion of this object (either caused by its own motion or by a moving camera). In that case, a more complex criterion would be needed to select the target object after crossing or occlusion. This is left for future work.

ACKNOWLEDGEMENTS

This research was partially supported by Consolider Ingenio 2010, project CSD2007-00018, by the CICYT project DPI 2007-61452 and by a grant from the *Universitat Rovira i Virgili* (URV).

REFERENCES

Amézquita Gómez N., Alquézar R. and Serratosa F., 2007. A new method for object tracking based on regions instead of contours. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR), Minneapolis, Minnesota*.

Amézquita Gómez N., Alquézar R. and Serratosa F., 2008. Dealing with occlusion in a probabilistic object tracking method. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR), Anchorage, Alaska*.

Bugeau A. and Pérez P., 2008. Track and cut: simultaneous tracking and segmentation of multiple objects with graph cuts. In *Proc. Third Int. Conf. on Computer Vision Theory and Applications*, Funchal, Madeira, Portugal.

Collins R. and Liu Y., 2005. On-line selection of discriminative tracking features. In *IEEE Trans. Pattern Anal. Machine Intell.*, 27 (10), 1631-1643.

Collins R., Zhou X. and Teh S.K., 2005. An open source tracking testbed and evaluation web site. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS'2005)*.

Comaniciu D. and Meer P., 1999. Mean shift analysis and applications. In *Proceedings ICCV'99*, 1197-1203.

Comaniciu D. and Meer P., 2002. Mean shift: a robust approach toward feature space analysis. In *IEEE Trans. Pattern Anal. Machine Intell.*, 24 (5), 603-619.

Comaniciu D., Ramesh V. and Meer P., 2000. Real-time tracking of non-rigid objects using mean shift. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Hilton Head, SC, vol. II, 142–149.

Comaniciu D., Ramesh V. and Meer P., 2003. Kernel-based object tracking. In *IEEE Trans. Pattern Anal. Machine Intell.*, 25 (4), 564-577.

Hariharakrishnan K. and Schonfeld D., 2005. Fast object tracking using adaptive block matching: In *IEEE Trans. Multimedia*, 7 (5), 853-859.

Ito K. and Sakane S., 2001. Robust view-based visual tracking with detection of occlusions: In *Proc. Int. Conf. Robotics Automation*, vol. 2, 1207-1213.

Jepson A.D., Fleet D.J., and EI-Maraghi T.F., 2003. Robust online appearance models for visual tracking. In *IEEE Trans. PAMI*, 25 (10), 1296-1311.

Nguyen H.T. and Smeulders A.W.M., 2004. Fast occluded object tracking by a robust appearance filter. In *IEEE Trans. Pattern Anal. Mach. Intell.*, 26 (8), 1099-1104.

Pan J., and Hu B., 2007. Robust occlusion handling in object tracking. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR), Minneapolis, Minnesota*.

Senior A. *et al.*, 2006. Appearance models for occlusion handling. In *J. Image Vis. Comput.*, 24 (11), 1233-1243.

Yin F., Makris D. and Velastin S.A., 2007. Performance evaluation of object tracking algorithms. In *Proc. 10th IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance (PETS'2007)*.

Zhou S.K., Chellappa R. and Moghaddam B., 2004. Visual tracking and recognition using appearance-adaptive models in particle filters. In *IEEE Trans. Image Process.*, 13 (11), 1491-1506.

Zhu L., Zhou J. and Song J., 2008. Tracking multiple objects through occlusion with online sampling and position. In *Pattern Recognition*, 41 (8), 2447-2460.