Tracking Multiple Objects in Complex Scenes[1]

C. Beleznai[2], T. Schlögl[2], B. Wachmann[3], H. Bischof[4], W. Kropatsch[5]

[2]*Advanced Computer Vision GmbH - ACV, Donaucity-straße 1, A-1220 Vienna, Austria, {csaba.beleznai@acv.ac.at}*
[3]*Siemens AG Österreich, Programm- und Systementwicklung, Graz, Austria*
[4]*Institute for Computer Graphics and Vision, Graz University of Technology, Austria*
[3]*Pattern Recognition and Image Processing Group, Vienna University of Technology, Austria*

Abstract: *Automatic video surveillance-based security is one of the most dynamically developing segment of the security industry. There has been a large progress of related vision hardware in the recent years, however, methods used to detect and monitor moving objects still lack the robustness to handle specific events occurring in complex scenes. In this paper we propose an algorithm to resolve occlusion events based on the smoothness constraint of kinematic and color features. The methods are tested using two images sequences and qualitative results are presented in terms of the accuracy and stability of obtained trajectories.*

1 Introduction

Intelligent video surveillance systems gain more and more importance. Object tracking is an essential component of such a system, since interpreting the gathered spatio-temporal surveillance data opens up new possibilities, such as activity analysis and monitoring.

Tracking involves the estimation of interframe correspondences within the pool of detected image objects. Finding the correspondences for single, isolated objects is fairly straightforward, however, in realistic scenarios of practical interest often multiple, interacting objects appear generating more complex situations.

The presented work describes a tracking scheme, which is built upon a low-level motion-based segmentation method and handles events of appearance, disappearance, splitting and merging of objects. The latter occurrences represent the most difficult events of tracking. We propose a novel solution based on the kinematic smoothness constraint to resolve the ambiguity existing at merging and splitting events.

The paper is organized as follows: in Section 2 a brief description of the tracking task to be solved is given; Section 3 provides an overview on the applied tracking method; Section

[1] This work has been carried out within the K plus Competence Center ADVANCED COMPUTER VISION and was funded from the K plus program.

4 presents tracking results for a complex scenario and describes them in terms of the tracking stability.

2 The tracking task

The task of detecting and tracking moving objects from video deals with the problem of extracting moving objects (foreground-background separation) and generating corresponding persistent trajectories [1]. In the case of multiple objects in the scene, the tracking task is equivalent with the task of solving the correspondence problem. At each frame a set of models and a set of measured objects (blobs) are available. Each object is „identified“ by finding the matching model.

In our work moving objects are detected by motion-based segmentation of the image using the algorithm described in [2] resulting in a multiple-state (moving, non-moving, foreground and background) classification of pixels.

In general, high occurrences of objects that visually overlap cause difficulties for a tracking system. Since blob generation of moving objects is based on connected component analysis, touching objects generate a single merged object, where pixel classification, i.e. to which original blob individual pixels belong is hard to resolve. This leads to the problem, that in a merged state individual tracks can not be updated. In order to overcome this problem, we propose a solution using a technique, which generates plausible trajectories of the objects in a merged state by performing matching between objects entering and leaving the merged state. The matching is based on the kinematic smoothness constraint [3] and on the smoothness of the object color features. The method is presented in section 3.1.

We also describe a method handling splitting events in a robust manner. Often, the blob detector generates inconsistent blobs where the blob erroneously splits into two or more parts because the connected component analysis finds several smaller regions instead of the integral object. To avoid erroneous splitting events, we apply a postponed splitting scheme presented in section 3.2.

3 The tracking method

The main concept of our tracking scheme is based on hypotheses, similarly to the system described in [2]. During the initialization each detected blob creates a hypothesis with the following attributes: (1) time of creation; (2) position; (3) velocity vector; (4) confidence; (5) color; (6) status and (7) list of merged hypothesis indices.

The confidence attribute provides a probability measure that the hypothesis belongs to a consistently appearing object. Each time a hypothesis is matched to an existing object, the confidence is increased, whereas upon no match the confidence is lowered. Below a certain confidence level the given hypothesis is removed. Due to this scheme, suddenly appearing and disappearing objects will have a linked hypothesis with low confidence. Hypotheses belonging to objects exhibiting short-term disappearance - due to occlusions or failure of the blob detection – will still survive and provide intact trajectories. Details on the color feature matching are given in section 3.3 describing the cost functions used for matching. The status attribute yields the information whether the hypothesis is in a merged state. The member list of a hypothesis is usually empty, upon merging, however, a new hypothesis is generated containing the merged hypothesis indices in the member list.

The scheme of the tracking algorithm is outlined as follows:

1. Initialization step (see above);
2. In each frame F_i observations are compared to the existing hypotheses, similarity measures based on the Euclidean distance, overlap and color similarities are computed and correspondences are established using a greedy matching scheme.
3. If a hypothesis matches exactly one blob, the hypothesis properties (position, velocity, color) are updated and the hypothesis confidence is increased. The velocity and color update follow a temporal update scheme analogously as described in [5].
3. In case of detecting a non-matched hypothesis, its confidence is lowered. If the hypothesis confidence has dropped below a specific threshold, the hypothesis is deleted and the corresponding trajectory is considered to be completed.
4. When a non-matched blob is found, a new hypothesis with an initial low confidence is generated and a new trajectory is initiated.
5. If a blob matches two or more hypotheses, merging is assumed. The state of the merging hypotheses is set to “merged” and they become temporarily disabled by restricting them from any further matching process (Step 2). Moreover a new group-hypothesis (GH) is generated possessing the index list of merging hypotheses.

   It can happen, that some hypotheses involved in the merging event are already group-hypotheses. In this case, a new GH is generated inheriting the members of the previous GH and the previous GH is deleted.

   If a GH exists for a longer time (typically >50 frames was used), the GH is converted into an ordinary hypothesis by inheriting the trajectory of the member hypotheses, deleting the member hypotheses and emptying the member list.

6. If a hypothesis matches several blobs, we assume a potential splitting event. Splitting is performed only if the splitting condition (see section 3.2) is fulfilled. When a hypothesis to be split is not a GH (such as a group of people formed before entering the camera field of view), new hypotheses are generated inheriting the trajectory information of the original hypothesis. If the splitting hypothesis is a GH, the matching scheme described in section 3.1 is used to reactivate and compare the member hypotheses of the GH to the observed blobs.
7. Using the hypothesis velocity vectors, hypothesis positions are propagated using a simple first-order motion model.

3.1 Resolving object overlaps

As it is described in the above section, in the case of blob overlaps the observation of a single merged blob does not allow to reconstruct the trajectories of the original entering blobs. In human tracking scenarios temporary overlaps might last for several seconds depending on the camera viewpoint and the walking speed of humans. Assuming stable kinematic and color features for the participating objects, one can expect that the objects entering the merged state at frame F_i and objects leaving the merged state at frame F_k ($k > i$) will have similar features.

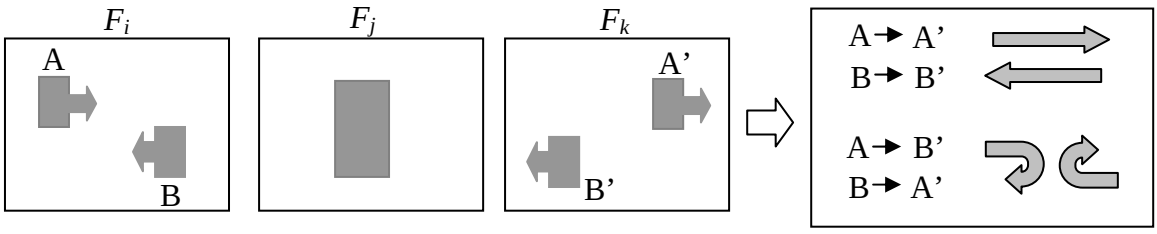

Figure 1: An overlapping event of blobs and possible matches between the blobs entering and leaving the merged state. Note that the first possible match (top right) corresponds to a smooth trajectory, whereas the second possible match (bottom right) provides blob movements with a sudden change in direction.

Figure 1 illustrates an example scenario of blob overlaps. Blobs entering (A, B) and leaving (A', B') the merged state are matched. Due to the imposed trajectory smoothness constraint the first match describing anti-parallel motion will be accepted.

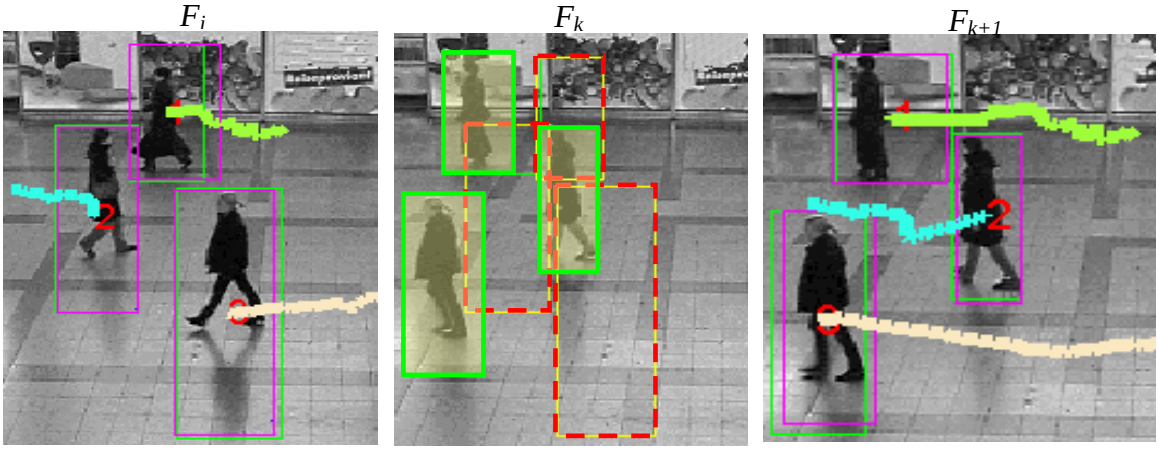

Figure 2: Three images illustrating the merging event and its resolution. The left image displays the blobs entering the merged state. In the middle, blobs are shown leaving the merged state (semi-transparent boxes) along with the possible input correspondences (dashed outlines). In the right image the resulting matches are shown with the updated trajectories.

A second example is shown in Figure 2. Upon splitting (middle image), the group hypothesis is used to retrieve information about the previously entering hypotheses. If the number of the exiting objects and entering hypotheses is equal (N = M), an exhaustive search procedure is initiated to find the minimum cost match based on the cost function described in Section 3.3.

If the number of entering hypotheses is larger than that of the exiting objects (N > M), a greedy algorithm is used to match M-1 entry hypotheses and exiting objects and the non-matched exiting object is assumed to contain a group. This latter assumption may be incorrect when more than three objects participate at the merging event and split off gradually from the group, however, such cases occur less frequently, usually in scenarios of high people density.

When the number of entering hypotheses is smaller than that of the exiting objects (N<M), N-1 exiting objects are matched with entering hypotheses using a greedy matching scheme and for the rest of exiting objects new hypotheses are generated.

3.2 Splitting verification

Splitting might occur due to the failure of blob detection or due to actual splitting of two or more objects. If the splitting is temporary, the split parts remain in each others close vicinity, whereas for an actually splitting group of objects, the objects become entirely separated after a short period of time. Thus the spatial extent of the hypothesis (W_H, H_H) and the spatial extent of the split ensemble (W_E, H_E) are investigated. If the size of the ensemble becomes significantly larger than that of the hypothesis, the splitting event is accepted. This scheme is illustrated in Figure 3.

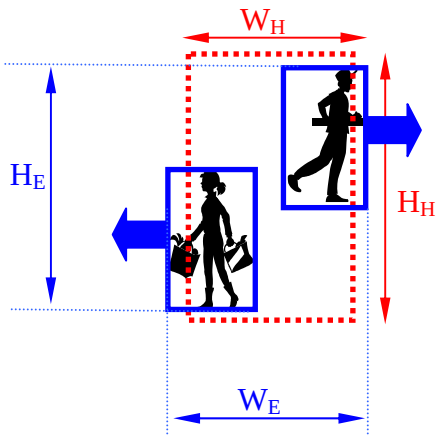

Figure 3: Illustration showing a possible splitting event. The size of the hypothesis (W_H, H_H) and the size of the ensemble (W_E, H_E) are determined and the size ratios are computed

The verification is performed by computing the width and height ratios (W_H / W_E and H_H / H_E). If any of the computed ratios drops below a threshold (typically 0.75 was used), the splitting event is allowed.

3.3 Cost function

The kinematic smoothness constraint uses the cost function described in [4]. The kinematic cost function S_k consists of two terms:

$$S_k(\mathbf{v_i}, \mathbf{v_j}) = w_1 \left(1 - \frac{\mathbf{v_i} \cdot \mathbf{v_j}}{\|\mathbf{v_i}\| \cdot \|\mathbf{v_j}\|}\right) + w_2 \left(1 - \frac{2\sqrt{\|\mathbf{v_i}\| \cdot \|\mathbf{v_j}\|}}{\|\mathbf{v_i}\| + \|\mathbf{v_j}\|}\right) \quad (1)$$

The two input vectors $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$ are compared and changes in the direction (first term) and changes in the magnitude are penalized. The first term contains the direction cosine, thus co-parallel velocity orientations thus yield a zero contribution. Changes in the velocity magnitude generate higher costs within the second term. The weights w_1 and w_2 were used to combine the two cost terms. The setting $w_1 = 0.7$ and $w_2 = 0.3$ was used, since due to the perspective view of our test scenes the velocity vector orientation was a more stable feature than its magnitude.

A second cost function derived for 2D-color histograms was used as well. Detailed description of this method is given in [5].

$$S_c = \sqrt{\sum_{rg=0}^{m-1}\sum_{by=0}^{m-1}\left(\frac{T_{rg,by}^{H}}{S_T^{H}} - \frac{T_{rg,by}^{B}}{S_T^{B}}\right)^2}$$

where $T_{rg,by}^{H}$, $T_{rg,by}^{B}$ are color templates (2D-histograms) of a hypothesis and a blob and S_T is a normalization constant ($S_T = \sum_{rg=0}^{m-1}\sum_{by=0}^{m-1} T_{rg,by}$) such that $0 \le S_C \le 1$. The scaling by the sum of the template values ensures the scale invariance of the color distance measures.

The two cost functions are combined by using two weight parameters W_1 and W_2 ($W_1 + W_2 = 1$). The overall cost function S is thus obtained by:

$$S = W_1 S_K + W_2 S_C$$

A lower weight for the color cost function was used ($W_1 = 0.8$ $W_2 = 0.2$) because of the high occurrence of uniform colored objects in our scenes.

4 Results and discussion

The tracking method has been applied to two image sequences containing multiple objects. One of the test sequences consists of the first 1300 frames of evaluation data set used by the Second IEEE Workshop on Performance Evaluation of Tracking and Surveillance. The image sequence contains eight moving objects, people and cars. Three object occlusions (merging and a subsequent splitting) occur and all of them is correctly resolved by the tracker. The hypothesis-based tracking approach successfully tracks blobs which are not detected for several frames. Several spurious splitting event is filtered out based on the splitting constraint defined in Section 3.2. An example frame for the tracking results is shown in Figure 4.

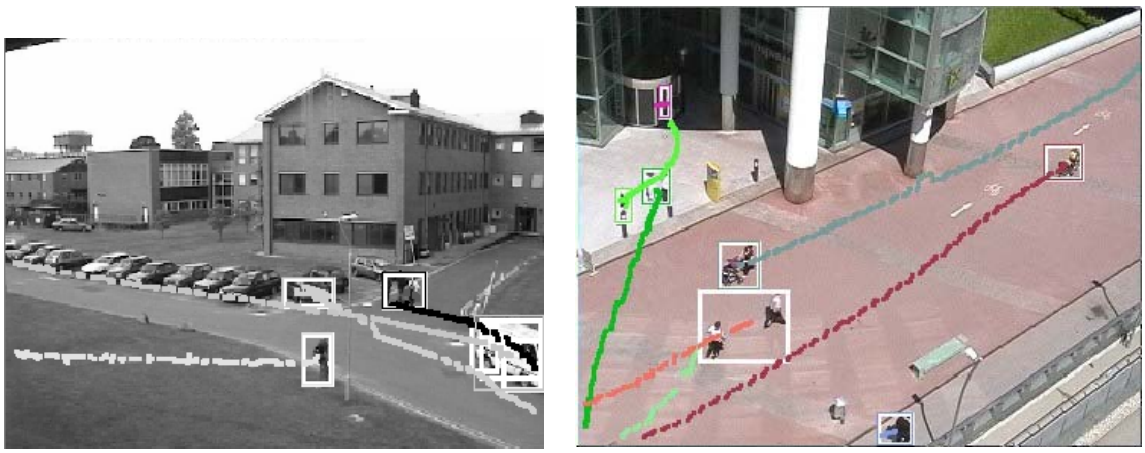

Figure 4: Tracking results for two scenarios containing multiple interacting objects.

A second sequence consisting of 3040 frames was also used to test the tacking method. The sequence contains 25 individuals generating 10 overlapping events.

A qualitative summary of the observed events is summarized in Table 1.

Sequence	No. of detected objects	Merging events	Spurious splitting events	Broken trajectories
1	8 out of 8	3 (3 correctly resolved)	3 (3 filtered out correctly)	0
2	25 out of 25	10 (8 correctly resolved)	5 (5 filtered out correctly)	3

Table 1: Table summarizing the critical events processed by the tracking method.

As it can be seen from Table 1, in the test sequences the system detects objects in a reliable manner. Two merging events of the Sequence 2 are not correctly resolved due to very long duration of occlusion (> 100 frames). Filtering out of spurious splitting events works well in all of the cases yielding no additional hypothesis. Broken trajectories were generated in sequence 2 because of short-term occlusions of people by the column in the center of the image. An optional post-processing of trajectories could be used to re-establish trajectories.

5. Conclusion

We have presented a tracking system using a novel approach based on the smoothness constraint to resolve blob overlaps and reconstruct object trajectories. A new method to evaluate splitting events is also presented and evaluated using two image sequences containing multiple objects. The obtained results show that the smoothness constraint applies well for the observed human motion and merging events can be resolved well by using this technique. Due to the applied splitting constraint many spurious splitting events are filtered out yielding an improved detection of individual blobs.

A further improvement of the tracker is anticipated to handle low frame rate (<10 fps) image sequences, where the hypothesis-blob matching process often fails due to the larger interframe object displacements.

6. References

[1] I. Haritaoglu, D. Harwood and L. S. Davis, "*W⁴: Real-Time Surveillance of People and Their Activities*", IEEE Trans. On Pattern Analysis and Matching Int., Vol. 22, No. 8, August 2000, pp. 809

[2] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa, "*A System for Video Surveillance and Monitoring: VSAM Final Report*", Technical Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, May 2000

[3] D. Chetverikov and J. Verestóy, „*Motion Tracking of Dense feature Point Sets*“, Proc. 21th Workshop of the Austrian Pattern recognition Group, Halstatt, Oldenbourg Verlag, 1997, pp. 233-242

[4] I. K. Sethi and R. Jain, „Finding trajectories of feature points in a monocular image sequence“, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 12, 1990, pp. 87-91

[5] T. Ellis and M. Xu, "*Object Detection and Tracking in an Open Dynamic World*", Proceedings 2nd IEEE Int. Workshop on PETS, Kauai, Hawaii, USA, December 9, 2001