Detecting and Tracking Multiple Moving Objects Using Temporal Integration*

Michal Irani, Benny Rousso, Shmuel Peleg

Dept. of Computer Science
The Hebrew University of Jerusalem
91904 Jerusalem, ISRAEL

Abstract. Tracking multiple moving objects in image sequences involves a combination of motion detection and segmentation. This task can become complicated as image motion may change significantly between frames, like with camera vibrations. Such vibrations make tracking in longer sequences harder, as temporal motion constancy can not be assumed.

A method is presented for detecting and tracking objects, which uses temporal integration without assuming motion constancy. Each new frame in the sequence is compared to a dynamic internal representation image of the tracked object. This image is constructed by temporally integrating frames after registration based on the motion computation. The temporal integration serves to enhance the region whose motion is being tracked, while blurring regions having other motions. These effects help motion analysis in subsequent frames to continue tracking the same motion, and to segment the tracked region.

1 Introduction

Motion analysis, such as *optical flow* [7], is often performed on the smallest possible regions, both in the temporal domain and in the spatial domain. Small regions, however, carry little motion information, and such motion computation is therefore very inaccurate. Analysis of multiple moving objects based on optical flow [1] suffers from this inaccuracy.

The major difficulty in increasing the size of the spatial region of analysis is the possibility that larger regions will include more than a single motion. This problem has been treated for image-plane translations with the *dominant translation* approach [3, 4]. Methods with larger temporal regions have also been introduced, mainly using a combined spatio-temporal analysis [6, 10]. These methods assume motion constancy in the temporal regions, i.e., motion should be constant in the analyzed sequence.

In this paper we propose a method for detecting and tracking multiple moving objects using both a large spatial region and a large temporal region without assuming temporal motion constancy. When the large spatial region of analysis has multiple moving objects, the motion parameters and the locations of the objects are computed for one object after another. The method has been applied successfully to parametric motions such as affine and projective transformations. Objects are tracked using temporal integration of images registered according to the computed motions.

Sec. 2 describes a method for segmenting the image plane into differently moving objects and computing their motions using two frames. Sec. 4 describes a method for tracking the detected objects using temporal integration.

* This research has been supported by the Israel Academy of Sciences.

2 Detection of Multiple Moving Objects in Image Pairs

To detect differently moving objects in an image pair, a single motion is first computed, and a single object which corresponds to this motion is identified. We call this motion the *dominant motion*, and the corresponding object the *dominant object*. Once a dominant object has been detected, it is excluded from the region of analysis, and the process is repeated on the remaining region to find other objects and their motions.

2.1 Detection of a Single Object and its Motion

The motion parameters of a single translating object in the image plane can be recovered accurately, by applying the iterative translation detection method mentioned in Sec. 3 to the entire region of analysis. This can be done even in the presence of other differently moving objects in the region of analysis, and with no prior knowledge of their regions of support [5]. It is, however, rarely possible to compute the parameters of a higher order parametric motion of a single object (e.g. affine, projective, etc.) when differently moving objects are present in the region of analysis.

Following is a summary of the procedure to compute the motion parameters of an object among differently moving objects in an image pair:

1. Compute the dominant translation in the region by applying a translation computation technique (Sec. 3) to the entire region of analysis.
2. Segment the region which corresponds to the computed motion (Sec. 3). This confines the region of analysis to a region containing only a single motion.
3. Compute a higher order parametric transformation (affine, projective, etc.) for the segmented region to improve the motion estimation.
4. Iterate Steps 2-3-4 until convergence.

The above procedure segments an object (the *dominant* object), and computes its motion parameters (the *dominant* motion) using two frames. An example for the determination of the dominant object using an affine motion model between two frames is shown in Fig. 2.c. In this example, noise has affected strongly the segmentation and motion computation. The problem of noise is overcome once the algorithm is extended to handle longer sequences using temporal integration (Sec. 4).

3 Motion Analysis and Segmentation

This section describes briefly the methods used for motion computation and segmentation. A more detailed description can be found in [9].

Motion Computation. It is assumed that the motion of the objects can be approximated by 2D parametric transformations in the image plane. We have chosen to use an iterative, multi-resolution, gradient-based approach for motion computation [2, 3, 4]. The parametric motion models used in our current implementation are: pure translations (two parameters), affine transformations (six parameters [3]), and projective transformations (eight parameters [1]).

Segmentation. Once a motion has been determined, we would like to identify the region having this motion. To simplify the problem, the two images are registered using the detected motion. The motion of the corresponding region is therefore cancelled, and the problem becomes that of identifying the stationary regions.

In order to classify correctly regions having uniform intensity, a multi-resolution scheme is used, as in low resolution pyramid levels the uniform regions are small. The lower resolution classification is projected on the higher resolution level, and is updated according to higher resolution information (gradient or motion) when it conflicts the classification from the lower resolution level.

Moving pixels are detected in each resolution level using only local analysis. A simple grey level difference is not sufficient for determining the moving pixels. However, the grey level difference normalized by the gradient gives better results, and was sufficient for our experiments. Let $I(x,y,t)$ be the gray level of pixel (x,y) at time t, and let $\nabla I(x,y,t)$ be it spatial intensity gradient. The *motion measure* $D(x,y,t)$ used is the weighted average of the intensity differences normalized by the gradients over a small neighborhood $N(x,y)$ of (x,y).

$$D(x,y,t) \stackrel{\text{def}}{=} \frac{\sum_{(x_i,y_i)\in N(x,y)} |I(x_i,y_i,t+1) - I(x_i,y_i,t)|\,|\nabla I(x_i,y_i,t)|}{\sum_{(x_i,y_i)\in N(x,y)} |\nabla I(x_i,y_i,t)|^2 + C} \tag{1}$$

where the constant C is used to avoid numerical instabilities. The motion measure (1) is propagated in the pyramid according to its certainty at each pixel. At the highest resolution level a threshold is taken to segment the image into moving and stationary regions. The stationary region $M(t)$ represents the tracked object.

4 Tracking Objects Using Temporal Integration

The algorithm for the detection of multiple moving objects described in Sec. 2 is extended to track objects in long image sequences. This is done by using temporal integration of images registered with respect to the tracked motion, without assuming temporal motion constancy. The temporally integrated image serves as a dynamic internal representation image of the tracked object.

Let $\{I(t)\}$ denote the image sequence, and let $M(t)$ denote the segmentation mask of the tracked object computed for frame $I(t)$, using the segmentation method described in Sec. 3. Initially, $M(0)$ is the entire region of analysis. The temporally integrated image is denoted by $Av(t)$, and is constructed as follows:

$$\begin{cases} Av(0) & \stackrel{\text{def}}{=} I(0) \\ Av(t+1) & \stackrel{\text{def}}{=} w \cdot I(t+1) + (1-w) \cdot register(Av(t), I(t+1)) \end{cases} \tag{2}$$

where currently $w = 0.3$, and $register(P,Q)$ denotes the registration of images P and Q by warping P towards Q according to the motion of the tracked object computed between them. A temporally integrated image is shown in Fig. 1.

Following is a summary of the algorithm for detecting and tracking the dominant object in an image sequence, starting at $t = 0$:

1. Compute the dominant motion parameters between the integrated image $Av(t)$ and the new frame $I(t+1)$, in the region $M(t)$ of the tracked object (Sec. 2).
2. Warp the temporally integrated image $Av(t)$ and the segmentation mask $M(t)$ towards the new frame $I(t+1)$ according to the computed motion parameters.

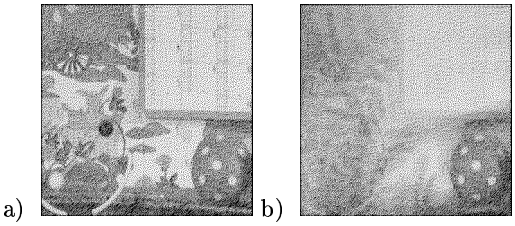

Fig. 1. An example of a temporally integrated image.
a) A single frame from a sequence. The scene contains four moving objects.
b) The temporally integrated image after 5 frames. The tracked motion is that of the ball which remains sharp, while all other regions blur out.

3. Identify the stationary regions in the registered images above (Sec. 3), using the registered mask $M(t)$ as an initial guess. This will be the tracked region in $I(t+1)$.
4. Compute the integrated image $Av(t+1)$ using (2), and process the next frame.

When the motion model approximates well enough the temporal changes of the tracked object, shape changes relatively slowly over time in registered images. Therefore, temporal integration of registered frames produces a sharp and clean image of the tracked object, while blurring regions having other motions. An example of a temporally integrated image of a tracked rolling ball is shown in Fig. 1. Comparing each new frame to the temporally integrated image rather than to the previous frame gives the a strong bias to keep tracking the same object. Since additive noise is reduced in the the average image of the tracked object, and since image gradients outside the tracked object decrease substantially, both segmentation and motion computation improve significantly.

In the example shown in Fig. 2, temporal integration is used to detect and track a single object. Comparing the segmentation shown in Fig. 2.c to the segmentation in Fig. 2.d emphasizes the improvement in segmentation using temporal integration.

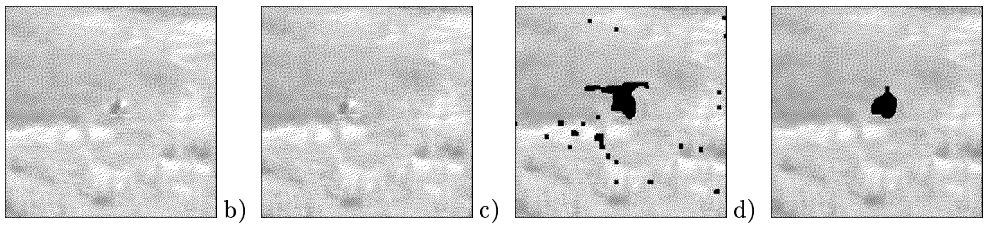

Fig. 2. Detecting and tracking the dominant object using temporal integration.
a-b) Two frames in the sequence. Both the background and the helicopter are moving.
c) The segmented dominant object (the background) using the dominant affine motion computed between the first two frames. Black regions are those excluded from the dominant object.
d) The segmented tracked object after a few frames using temporal integration.

Another example for detecting and tracking the dominant object using temporal integration is shown in Fig. 3. In this sequence, taken by an infrared camera, the background moves due to camera motion, while the car has another motion. It is evident that the tracked object is the background, as other regions were blurred by their motion.

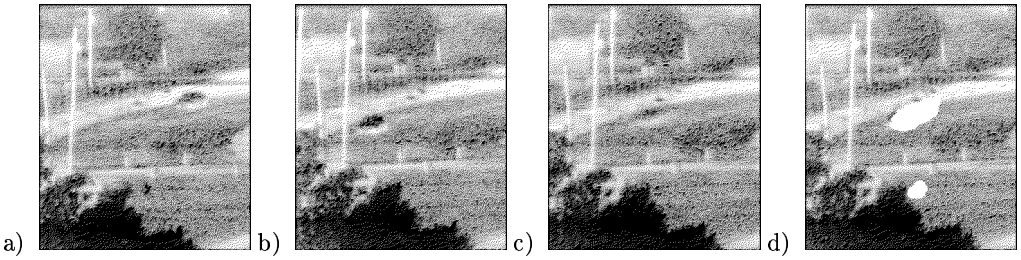

Fig. 3. Detecting and tracking the dominant object in an image sequence using temporal integration.
a-b) Two frames in an infrared sequence. Both the background and the car are moving.
c) The temporally integrated image of the tracked object (the background). The background remains sharp with less noise, while the moving car blurs out.
d) The segmented tracked object (the background) using an affine motion model. White regions are those excluded from the tracked region.

This temporal integration approach has characteristics similar to human motion detection. For example, when a short sequence is available, processing the sequence back and forth improves the results of the segmentation and motion computation, in a similar way that repeated viewing helps human observers to understand a short sequence.

4.1 Tracking Other Objects

After segmentation of the first object, and the computation of its affine motion between every two successive frames, attention is given to other objects. This is done by applying once more the tracking algorithm to the "rest" of the image, after excluding the first detected object. To increase stability, the displacement between the centers of mass of the regions of analysis in successive frames is given as the initial guess for the computation of the dominant translation. This increases the chance to detect fast small objects.

After computing the segmentation of the second object, it is compared with the segmentation of the first object. In case of overlap between the two segmentation masks, pixels which appear in the masks of both the first and the second objects are examined. They are reclassified by finding which of the two motions fits them better.

Following the analysis of the second object, the scheme is repeated recursively for additional objects, until no more objects can be detected. In cases when the region of analysis consists of many disconnected regions and motion analysis does not converge, the largest connected component in the region is analyzed.

In the example shown in Fig. 4, the second object is detected and tracked. The detection and tracking of several moving objects can be performed in parallel, by keeping a delay of one or more frame between the computations for different objects.

5 Concluding Remarks

Temporal integration of registered images proves to be a powerful approach to motion analysis, enabling human-like tracking of moving objects. The tracked object remains sharp while other objects blur out, which improves the accuracy of the segmentation and the motion computation. Tracking can then proceed on other objects.

Enhancement of the tracked objects now becomes possible, like reconstruction of occluded regions, and improvement of image resolution [8].

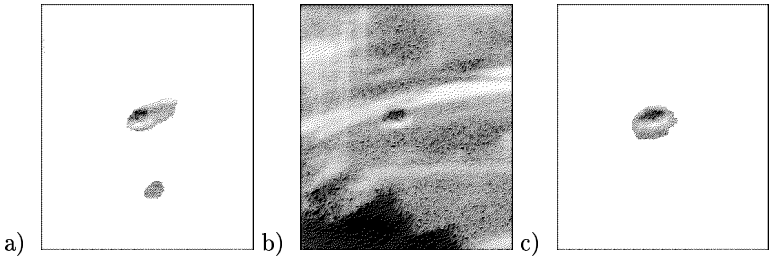

Fig. 4. Detecting and tracking the second object using temporal integration.
a) The initial segmentation is the complement of the first dominant region (from Fig. 3.d).
b) The temporally integrated image of the second tracked object (the car). The car remains sharp while the background blurs out.
c) Segmentation of the tracked object after 5 frames.

References

1. G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(4):384–401, July 1985.
2. J.R. Bergen and E.H. Adelson. Hierarchical, computationally efficient motion estimation algorithm. *J. Opt. Soc. Am. A.*, 4:35, 1987.
3. J.R. Bergen, P.J. Burt, K. Hanna, R. Hingorani, P. Jeanne, and S. Peleg. Dynamic multiple-motion computation. In Y.A. Feldman and A. Bruckstein, editors, *Artificial Intelligence and Computer Vision: Proceedings of the Israeli Conference*, pages 147–156. Elsevier (North Holland), 1991.
4. J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg. Computing two motions from three frames. In *International Conference on Computer Vision*, pages 27–32, Osaka, Japan, December 1990.
5. P.J. Burt, R. Hingorani, and R.J. Kolczynski. Mechanisms for isolating component patterns in the sequential analysis of multiple motion. In *IEEE Workshop on Visual Motion*, pages 187–193, Princeton, New Jersey, October 1991.
6. D.J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, 1:279–302, 1988.
7. B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
8. M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical Models and Image Processing*, 53:231–239, May 1991.
9. M. Irani, B. Rousso, and S. Peleg. Detecting multiple moving objects using temporal integration. Technical Report 91-14, The Hebrew University, December 1991.
10. M. Shizawa and K. Mase. Simultaneous multiple optical flow estimation. In *International Conference on Pattern Recognition*, pages 274–278, Atlantic City, New Jersey, June 1990.

This article was processed using the LaTeX macro package with ECCV92 style