Image Sequence Enhancement Using Multiple Motions Analysis*

Michal Irani†   Shmuel Peleg

Institute of Computer Science
The Hebrew University of Jerusalem
91904 Jerusalem, ISRAEL

Abstract

Accurate computation of image motion enables the enhancement of image sequences. Motion computation in scenes having multiple moving objects is performed together with object segmentation by using a unique temporal integration approach.

Having accurate motion estimation for image regions, these regions can be enhanced by fusing all successive frames covering the same region. Enhancement includes improvement of image resolution, filling-in occluded regions, and reconstruction of components in scenes involving transparency.

1 Introduction

We describe 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 image motions such as affine and projective transformations. Objects are tracked using temporal integration of images registered according to the computed motions [9].

Once an object has been tracked and segmented, it can be enhanced using information from several frames. Tracked objects can be enhanced by filling-in occluded regions, and by improving the spatial resolution of the imaged objects. When the scene contains transparent motions, the transparent objects can be tracked and reconstructed separately.

Sect. 2 describes briefly a method for segmenting the image plane into differently moving objects and computing their motions using two frames. Sect. 3 describes briefly the tracking of detected objects using temporal integration. Sect. 4 describes the algorithms for image enhancement. More details on the motion analysis and segmentation methods can be found in [9].

*This research was supported by the Israel Academy of Sciences.

†M. Irani was partially supported by a fellowship from the Leibniz Center.

2 Multiple Motions 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 image regions to find other objects and their motions.

2.1 Processing the First Object

Motion Computation. It is assumed that the projected 3D motions of the objects can be approximated by some 2D parametric transformation 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 translation (2 parameters), affine transformation (6 parameters, [3]) and projective transformation (8 parameters [1]).

The motion parameters of a single object in the image plane can be recovered by applying the iterative detection method 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, 9].

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 canceled, and the

0-8186-2855-3/92 $3.00 ©1992 IEEE

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 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.

Noise Sensitivity. The motion analysis and segmentation computed from two frames, as described in this section, are sensitive to noise. It is difficult to distinguish between intensity variations due to noise and intensity variations due to motion. The problem of noise is overcome once the algorithm is extended to handle longer sequences using temporal integration.

3 Tracking by Temporal Integration

The algorithm for the detection of multiple moving objects described in Sect. 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. 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 mentioned in Sect. 2.1. Initially, $M(0)$ is the entire region of analysis. The temporally integrated image is denoted by $Av(t)$, and is constructed as follows:

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

where $register(P,Q)$ denotes the registration of images P and Q by warping P towards Q according to the motion of the tracked object computed between them, and $0 < w < 1$ (currently $w = 0.7$). An example of 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 $t = 0$, do:
1. Compute the dominant motion parameters between the temporal average $Av(t)$ and the new frame $I(t+1)$, in the region $M(t)$ of the tracked object (Sect. 2).

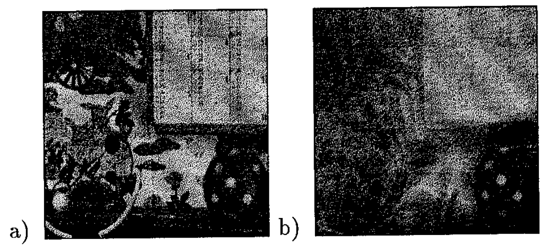

Figure 1: A temporally integrated image.
a) A single frame from a sequence. The scene contains four moving objects.
b) The temporally integrated image after 5 frames. The tracked motion is that of the ball. All other regions blur out.

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.
3. Identify the stationary regions in the registered images (Sect. 2.1), using the registered mask $M(t)$ as an initial guess. This will be the region $M(t+1)$ of the tracked object in frame $I(t+1)$.
4. Compute the next average $Av(t+1)$ using Eq. (1), and continue processing 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. Fig. 1 shows a temporally integrated image of a tracked rolling ball. Comparing each new frame to the temporally integrated image rather than to the previous frame gives the algorithm 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 Figs. 2.c and 2.d, temporal integration is used to detect and track the first object using an affine motion model. 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 in Fig. 2.c is the background, as all other regions in the image are blurred by their motion.

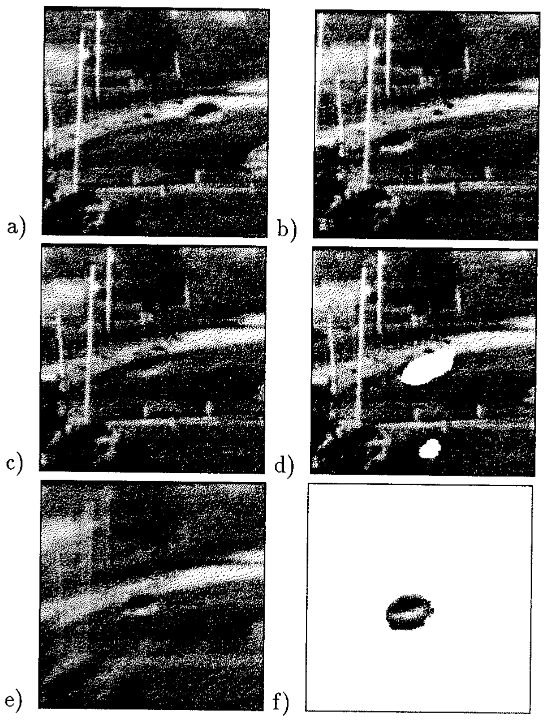

Figure 2: Detecting and tracking multiple moving objects using temporal integration (IR images).
a-b) The first and last frames. Both the background and the car are moving.
c) The temporally integrated image of the first tracked object (the background). The moving car blurs out.
d) Segmentation of the first tracked object (the background). White regions are those excluded from the tracked region.
e) The temporally integrated image of the second tracked object (the car). The background blurs out.
f) Segmentation of the second tracked object.

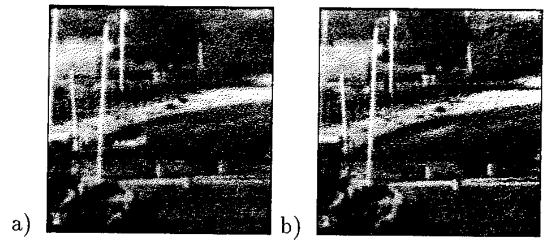

Figure 3: Reconstruction of occluded regions.
a) The car appears in all frames.
b) Full reconstruction of the background.

Tracking Other Objects. After segmentation of the first object, and the computation of its 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 from the region of analysis. The scheme is repeated recursively until no more objects can be detected [9]. An example of tracking the second object is shown in Figs. 2.e and 2.f.

4 Image Enhancement

Three methods are proposed for using the motion tracking for image enhancement: Reconstruction of occluded segments (Sect. 4.1), improvement of spatial resolution (Sect. 4.2), and reconstruction of transparent moving patters (Sect. 4.3).

4.1 Reconstruction of Occlusions

When parts of a tracked object are occluded in some frames, but appear in others, a more complete view of the object can be reconstructed. The image frames are registered using the computed motion parameters. The object is then reconstructed by temporally averaging gray levels of all pixels which were classified as object pixels. Object regions will be reconstructed even if they are occluded in some frames.

In the example shown in Fig. 3, the background of the infrared image sequence was completely reconstructed, eliminating the moving car from the scene.

4.2 Improvement of Spatial Resolution

The resolution of an image is determined by the physical characteristics of the sensor: the optics, the density of the detector elements, and their spatial response. Resolution improvement by modifying the sensor can be prohibitive. An increase in the sampling rate could, however, be achieved by obtaining more samples of the imaged object from a sequence of images in which the object appears moving. In this section, we present an algorithm for processing image sequences to obtain improved resolution of differently moving objects. This is an extension of our method presented in [8], which now handles more general motion models.

While earlier research on super-resolution [7, 8, 10] has dealt only with static scenes and with pure translational motion of the entire scene in the image plane, we deal with dynamic scenes and with more complex motions within the image plane. The segmentation of the image plane into the differently moving objects and their tracking, using the algorithm mentioned in Sections 2 and 3, enables processing of each object separately.

The Imaging Model. The imaging process, yielding the observed image sequence $\{g_k\}$, is modeled by: $g_k(m,n) = \sigma_k(h(T_k(f(x,y))) + \eta_k(x,y))$, where

- g_k is the image of the tracked object in the k_{th} frame.
- f is a high resolution image of the tracked object in a desired reconstruction view (the objective of the super-resolution algorithm).
- T_k is the 2D geometric transformation from f to g_k, determined by the computed motion parameters of the tracked object and by the degree of decrease in resolution.
- h is a blurring operator, determined by the point spread function of the sensor. When lacking knowledge of the sensor's properties, it is assumed to be a Gaussian.
- η_k is an additive noise term.
- σ_k is an operator which digitizes and decimates the image into pixels and quantizes the resulting pixels values.

The *receptive field* (in f) of a detector whose output is $g_k(m,n)$ is uniquely defined by its center (x,y) and its shape. The shape is determined by the region of support of the blurring operator h, and by the inverse geometric transformation ${T_k}^{-1}$. Similarly, the center (x,y) is obtained by ${T_k}^{-1}((m,n))$.

The construction of a higher resolution image $\hat{f}$, which approximates f as accurately as possible, and surpasses the visual quality of the observed images in $\{g_k\}$, is attempted. It is assumed that the acceleration of the camera while imaging a single image frame is negligible.

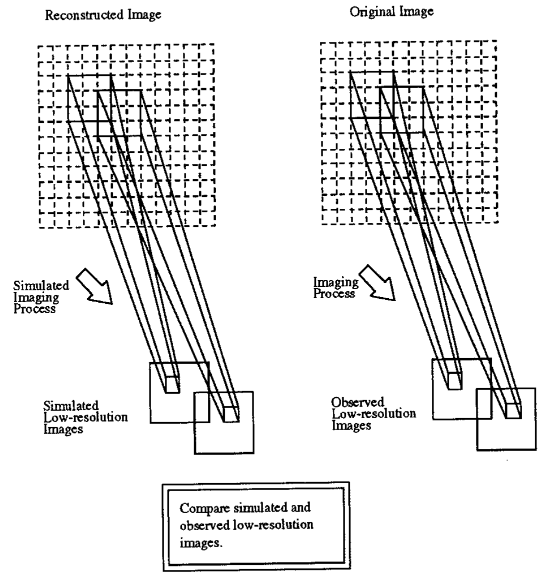

Figure 4: Schematic diagram of the super resolution algorithm

The Super-Resolution Algorithm. The presented algorithm for solving the super resolution problem is iterative. Starting with an initial guess $f^{(0)}$ for the high resolution image, the imaging process is simulated to obtain a set of low resolution images $\{g_k^{(0)}\}$ corresponding to the observed input images $\{g_k\}$. If $f^{(0)}$ were the correct high resolution image, then the simulated images $\{g_k^{(0)}\}$ should be identical to the observed images $\{g_k\}$. The difference images $\{g_k - g_k^{(0)}\}$ are then computed, and used to improve the initial guess by "backprojecting" each value in the difference images onto its receptive field in $f^{(0)}$, yielding an improved high resolution image $f^{(1)}$. This process is repeated iteratively to minimize the error function

$$e^{(n)} = \sqrt{\sum_k \sum_{(x,y)} (g_k(x,y) - g_k^{(n)}(x,y))^2}$$

The algorithm is schematically described in Fig. 4.

The imaging process of g_k at the n_{th} iteration is simulated by: $g^{(n)} = T_k(f^{(n)}) * h$. Let $\vec{x}$ denote a high resolution pixel, and $\vec{y}$ denote a low resolution pixel. Then the iterative update scheme of the high resolution image is expressed by:

$$f^{(n+1)}(\vec{x}) = f^{(n)}(\vec{x}) + \frac{\sum\limits_{\vec{y} \in \cup_k Y_{k,\vec{x}}} (g_k(\vec{y}) - g_k^{(n)}(\vec{y})) h^k_{\vec{x}\vec{y}}}{\sum\limits_{\vec{y} \in \cup_k Y_{k,\vec{x}}} h^k_{\vec{x}\vec{y}}} \quad (2)$$

where $Y_{k,\vec{x}}$ denotes the set of all pixels in g_k that include $\vec{x}$ in their receptive field, and $h^k_{\vec{x}\vec{y}}$ is the relative contribution of $f^{(n)}(\vec{x})$ in the imaging process of the pixel $g_k^{(n)}(\vec{y})$.

Eq. (2) computes the following: The value of $f^{(n)}$ at each high resolution pixel $\vec{x}$ is updated according to all the low resolution pixels $\vec{y}$ which it influences. The contribution to $\vec{x}$ to a low resolution pixel $\vec{y}$ belonging to an input image g_k is the error $(g_k(\vec{y}) - g_k^{(n)}(\vec{y}))$ multiplied by a factor of $h^k_{\vec{x}\vec{y}}$. Therefore, strongly influenced low resolution pixels also strongly influence $f^{(n+1)}(\vec{x})$, while weakly influenced low resolution pixels hardly influence $f^{(n+1)}(\vec{x})$. Since receptive fields of different low resolution pixels overlap, $f^{(n+1)}(\vec{x})$'s new value is influenced by several low resolution pixels. All corrections suggested by the various low resolution pixels are then averaged. Taking an average also reduces additive noise.

It is important to note that the original high resolution frequencies may not always be fully restored. For example, if the blurring function is an ideal low pass filter, and its Fourier transform has zero values at high frequencies, it is obvious that the frequency components which have been filtered out cannot be restored. In such cases, there is more than one high resolution image which gives the same low resolution images after the imaging process. A good choice of initial guess is the average of the registered low resolution images of the tracked object in the desired reconstruction view: $f^{(0)} = \frac{1}{K}\sum_{k=1}^{K} {T_k}^{-1}(g_k)$. Such an initial guess leads the algorithm to a smooth solution, which is usually a desired one. The algorithm converges rapidly (usually within less than 5 iterations), and has parallel characteristics.

In Fig. 5, the resolution of a car's license plate was improved from 15 frames.

4.3 Reconstruction of Objects in Transparent Motion

A region contains transparent motions if it contains several differently moving image patterns that appear superimposed. For example: moving shadows, spotlights, reflections in water, transparent surfaces moving past one another, etc. In this section, we present a method for isolating and reconstructing

a)

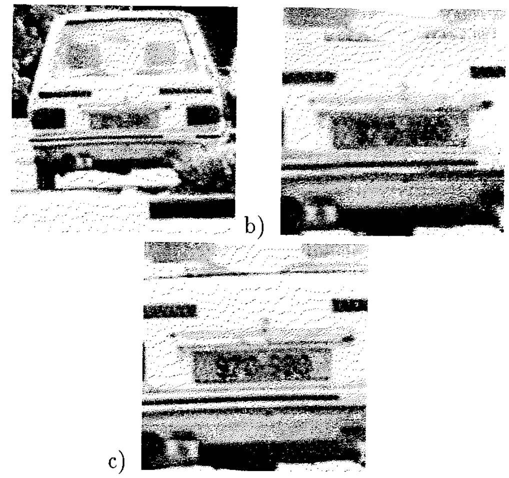

Figure 5: Improvement of spatial resolution using 15 frames. The sampling rate was increased by 2 in both directions.
a) The best frame from the image sequence.
b) The license plate magnified by 2 using bilinear interpolation.
c) The improved resolution image.

tracked objects in transparent motion. Previous analysis of transparency [4, 6, 11] assumed some motion constancy, which excludes most sequences taken from an unstabilized moving camera. Their work detected the motions of two superimposed transparent motions, but did not reconstruct the transparent objects.

Theoretically, transparent motion yields several motion components at each point, and segmentation cannot be used to isolate one of the transparent objects. In practice, however, pixels do not equally support different motions, so segmentation can be used to extract pixels which support better a single motion in the region of analysis. We use the temporal integration scheme described in Sect. 3 to track the dominant transparent object. The temporal averaging restores the dominant transparent object, while blurring out the other transparent objects, making them less noticeable. Comparing each new frame to the temporally integrated image rather than to the previous frame gives the algorithm a strong bias to keep tracking the same transparent object.

For recovering the second transparent object, the temporal integration method is applied once more to the sequence, after some delay. Let $Av_1(t)$ denote the temporally integrated image of the first transparent object. Starting at frame $I(t)$, the algorithm is applied

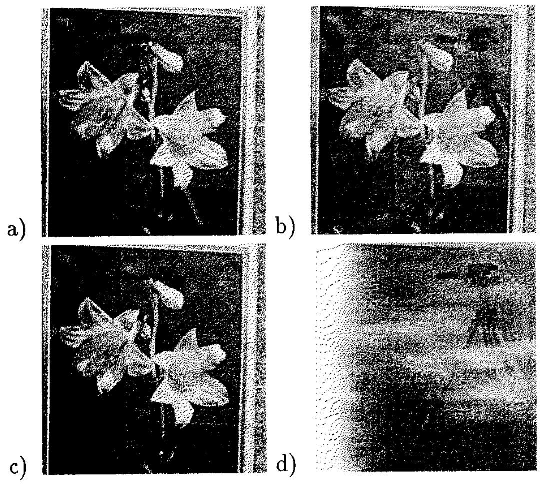

Figure 6: Reconstruction of "transparent" objects.
a-b) The first and last frames in a sequence. A moving tripod is reflected in the glass of a picture of flowers.
c) The first reconstructed object was the picture. The tripod faded away.
d) The second reconstructed object was the tripod. The flowers faded away.

only to pixels for which the value of $|I(t) - Av_1(t)|$ is high. This difference image has high values in regions which contain prominent features of transparent objects in $I(t)$ that faded away in the temporally integrated image $Av_1(t)$, and low values in regions which correspond to the first dominant transparent object. Therefore, we use the values of the absolute difference image as an initial mask for the search of the next dominant object in the temporal integration algorithm from Sect. 3. Now that the algorithm tracks the second dominant object, the new temporally integrated image $Av_2(t)$ restores the second dominant transparent object, and blurs out the other transparent objects, including the first dominant object.

In Fig. 6, the reconstruction of transparent moving objects is shown.

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 enables accurate segmentation and motion computation. Tracking can then proceed on other objects.

Information from several registered frames enables enhancement of tracked objects like reconstruction of occluded regions, improvement of image resolution, and reconstruction of transparent objects.

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, 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*, 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] T. Darrell and A. Pentland. Robust estimation of a multi-layered motion representation. In *IEEE Workshop on Visual Motion*, pages 173–178, Princeton, New Jersey, October 1991.

[7] T.S. Huang and R.Y. Tsai. Multi-frame image restoration and registration. In T.S. Huang, editor, *Advances in Computer Vision and Image Processing*, volume 1, pages 317–339. JAI Press Inc., 1984.

[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 and tracking multiple moving objects using temporal integration. In *European Conference on Computer Vision*, May 1992.

[10] S.P. Kim, N.K. Bose, and H.M. valenzuela. Recursive reconstruction of high resolution image from noisy undersampled multiframes. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(6):1013–1027, June 1990.

[11] 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.