Isolating Multiple Image Motions for Enhancement and 3D Analysis*

Michal Irani Benny Rousso Shmuel Peleg

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

Abstract

Motion computation in scenes having multiple moving objects is performed together with object segmentation by using a temporal integration approach. Using an accurate 2D motion estimate for image regions, they can be enhanced by fusing all successive frames covering the same region. Enhancement includes improvement of image resolution and filling-in occluded regions. It is also shown how an accurate 2D motion estimate for a *single planar* surface in a general static scene can help to compute the 3D motion performed by the camera.

1 Introduction

A method for detecting and tracking multiple moving objects, using both a large spatial region and a large temporal region, is described. 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 2D affine and projective motions in the image plane. Once an object has been tracked and segmented, it can be enhanced using information from several frames [13, 14]. Enhancement includes filling-in occluded regions and improving spatial resolution.

The 2D detection and tracking algorithm can also be used for estimating the camera motion (*ego-motion*) in general static 3D scenes. Once a single planar surface in a general static scene is detected in the image, and its 2D motion parameters computed, we use this data for estimating the entire 3D scene structure and the 3D motion performed by the camera.

Sect. 2 describes briefly a method for detecting and tracking the differently moving objects in the sequence. Sect. 3 describes the algorithms for image enhancement. Sect. 4 describes the method for computing the 3D motion of the camera (the ego-motion) in a static scene. More details can be found in [14, 15, 16].

2 Multiple Motions in Image Sequences

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. Temporal integration is then used to track detected objects throughout the image sequence. More details can be found in [15].

It is assumed that the projected 3D motions of the objects can be approximated by some 2D parametric transformation in the image plane. This assumption is valid when the differences in depth caused by the motions are small relative to the distances of the objects from the camera. We have chosen to use an iterative, multi-resolution, gradient-based approach for motion computation [3, 5, 6]. The parametric motion models used in our current implementation are: pure 2D translation (2 parameters), 2D affine transformation (6 parameters, [5, 4]) and projective transformation (8 parameters [1, 4]).

Detecting the First Object. 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

*This research was supported by the Israel Science Foundation.
M. Irani and B. Rousso were partially supported by a fellowship from the Leibniz Center.

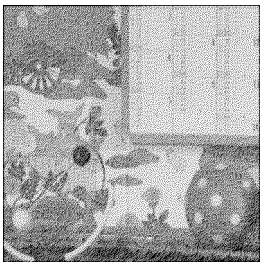
a)  b) 

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.

their regions of support [7, 16]. 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 problem becomes that of identifying the stationary regions. Detection of stationary regions is described in [15].

Tracking by Temporal Integration. Once an object has been detected, it can be tracked throughout the image sequence. 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 described in [15]. Initially, $M(0)$ is the entire region of analysis. The temporally integrated image is denoted by $Av(t)$, and is constructed as follows:

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

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.

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 Fig. 2, temporal integration is used to detect and track the first and second object. In this sequence, taken by an infrared camera, the background moves due to camera motion, while the car moves differently. It is evident that the tracked object in Fig. 2.c is the background, as the background maintains its sharpness, while all other regions in the image are blurred by their motion, and that the tracked object in Fig. 2.e is the car.

3 Image Enhancement

Once an object has been tracked and segmented, it can be enhanced using information from several frames. The methods presented for image enhancement are reconstruction of occluded segments and improvement of spatial resolution. More details can be found in [13, 14].

3.1 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 was completely reconstructed, eliminating the walking girl from the scene.

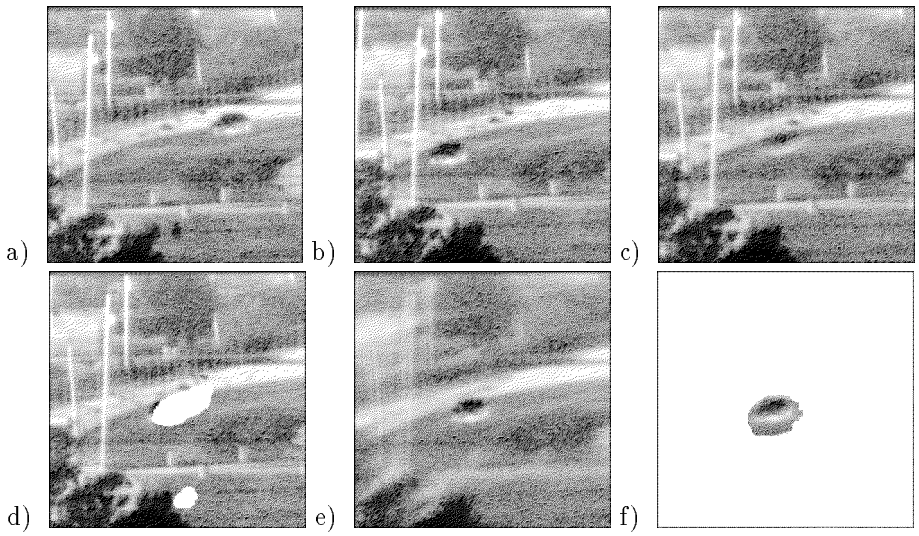


Figure 2: Detecting and tracking multiple moving objects using temporal integration (IR images).
a-b) The first and last frames. Both the background and the car are moving.
c) The temporally integrated image of the first tracked object (background). The car blurs out.
d) Segmentation of the first tracked object (background). White regions are those not belonging to the tracked region.
e) The temporally integrated image of the second tracked object (car). The background blurs out.
f) Segmentation of the second tracked object.

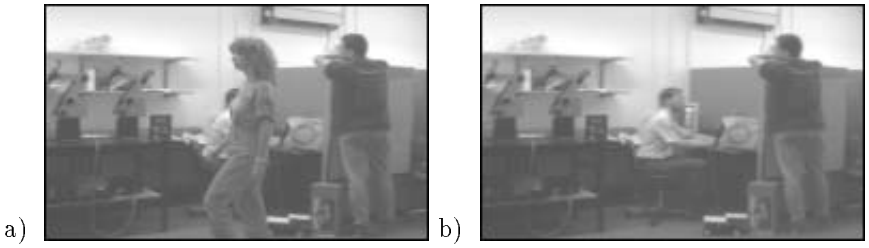


Figure 3: Reconstruction of occluded regions.

a) The girl appears in all frames and occludes a part of the background.
b) Full reconstruction of the background without the girl.

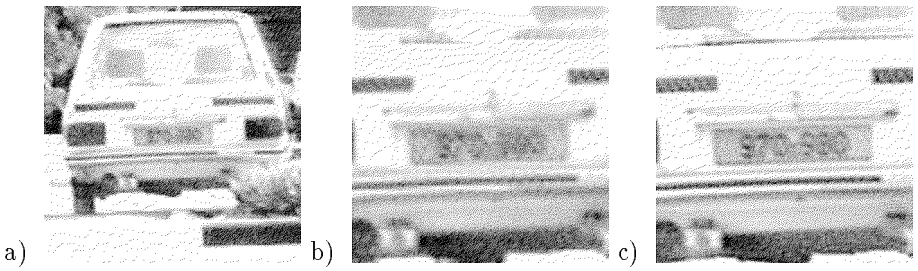


Figure 4: Improvement of spatial resolution using 15 frames. The sampling rate was increased by 2 in both directions. a) The best frame from the image sequence. b) The license plate magnified by 2 using bilinear interpolation. c) The improved resolution image.

3.2 Improvement of Spatial Resolution

The resolution of an image is determined by the physical characteristics of the sensor. 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 [13, 14] an algorithm was presented for processing image sequences to obtain improved resolution of differently moving objects.

While earlier research on super-resolution [12, 13, 18] 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 2D motions. The segmentation of the image plane into the differently moving objects and their tracking, using the algorithm mentioned in Section 2 enables processing of each object separately. The algorithm presented in [13, 14] for increasing the image resolution is similar to common iterative methods for solving sets of linear equations [19], and has similar properties, such as a rapid convergence (at exponential rate).

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 are used to improve the resolution of the initial guess by reducing the total energy in those difference images. For details see [13, 14].

In Fig. 4, the resolution of a car's license plate was improved using 15 frames and 5 iterations.

4 Ego-Motion in Static Scenes

Direct estimation of 3D motion is a difficult and ill-conditioned problem, due to the very large number of unknowns – the 3D motion parameters of the camera plus the depth at each point. 2D motion estimation, on the other hand, is a numerically stable problem, because the 2D problem is highly overdetermined (only six unknowns in the affine model, eight unknowns in the projective model).

Previous works on 3D motion estimation use the optical or normal flow field derived between two frames [1, 2, 8, 17, 20, 21], or the correspondence of previously extracted distinguished features (points, lines, contours) [10, 22]. Methods for computing the ego-motion *directly* from image intensities were also suggested [9, 11, 23], but each method has its limitations.

In this section we propose the following scheme in order to use the robustness of the 2D motion computation for computing 3D motion:

1. The 2D image motion of a single planar surface is computed (Sect. 2).

2. The two frames are registered according to the computed 2D motion parameters of the detected plane. This cancels the rotational component of the 3D camera motion, and the 3D translation of the camera can be computed from the two registered frames.

3. The 3D rotation of the camera is computed from the previously computed 3D translation of the camera and from the 2D motion parameters of the detected plane.

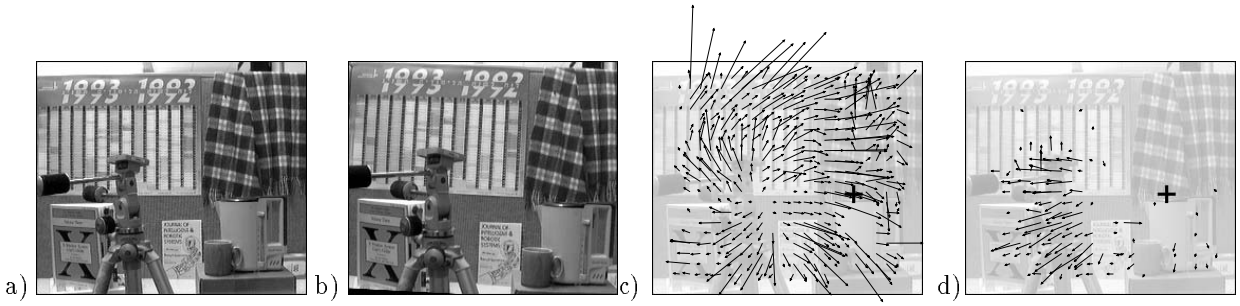


Figure 5: The optical flow before and after registration of the plane defined by the wall.
a) The first frame.
b) The second frame, taken after translating the camera by $(T_X, T_Y, T_Z) = (1.7cm, 0.4cm, 12cm)$ and rotating it by $(\Omega_X, \Omega_Y, \Omega_Z) = (0^\circ, -1.8^\circ, -3^\circ)$.
c) Optical flow between Figs. 5.a and 5.b (before registration), overlayed on Fig. 5.a .
d) Optical flow after registration of the wall. It is induced by pure translation, and points to the correct FOE (marked by +).

4. The 3D scene structure can then be reconstructed from the computed 3D motion parameters of the camera (using a scheme similar to that suggested in [9]).

4.1 Projected 2D Motion

When the field of view is not very large and the rotation is relatively small [1], a 3D motion of the camera between two image frames creates a 2D displacement (u, v) of an image point (x, y) in the image plane, which can be expressed by [4]:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f_c(\frac{T_X}{Z} + \Omega_Y) & + & x\frac{T_Z}{Z} & + & y\Omega_Z & - & x^2\frac{\Omega_Y}{f_c} & + & xy\frac{\Omega_X}{f_c} \\ -f_c(\frac{T_Y}{Z} - \Omega_X) & - & x\Omega_Z & + & y\frac{T_Z}{Z} & - & xy\frac{\Omega_Y}{f_c} & + & y^2\frac{\Omega_X}{f_c} \end{bmatrix} \quad (1)$$

where, (X, Y, Z) denote the Cartesian coordinates of the scene point projected onto (x, y), (T_X, T_Y, T_Z) and $(\Omega_X, \Omega_Y, \Omega_Z)$ are the 3D motion parameters of the camera (translation and rotation, respectively), and f_c is the focal length of the camera.

4.2 Reducing General Motion to Translation

Let (u, v) denote the 2D displacement field between f_1 and f_2, and let (u_s, v_s) denote the 2D motion parameters of a single 3D plane in the scene. Let f_1^{Reg} denote the frame obtained by warping frame f_1 towards f_2 according to (u_s, v_s). f_1^{Reg} and f_2 will be registered over regions of the projected 3D plane within the image, and unregistered over other image regions. The 2D motion between the registered frames (f_1^{Reg} and f_2) is therefore: $(u^{Reg}, v^{Reg}) = (u - u_s, v - v_s)$. Using Eq. (1) we get:

$$\begin{bmatrix} u^{Reg}(x,y) \\ v^{Reg}(x,y) \end{bmatrix} = \begin{bmatrix} u(x,y) - u_s(x,y) \\ v(x,y) - v_s(x,y) \end{bmatrix} = \begin{bmatrix} -f_c T_X(\frac{1}{Z} - \frac{1}{Z_s}) & + & xT_Z(\frac{1}{Z} - \frac{1}{Z_s}) \\ -f_c T_Y(\frac{1}{Z} - \frac{1}{Z_s}) & + & yT_Z(\frac{1}{Z} - \frac{1}{Z_s}) \end{bmatrix} \quad (2)$$

where, $Z_s = Z_s(x, y)$ is the depth function of the 3D plane at pixel (x, y), and $Z = Z(x, y)$ is the real depth at that pixel.

This registration cancels the rotation parameters $(\Omega_X, \Omega_Y, \Omega_Z)$ in Equation (2), leaving the *original* translation parameters (T_X, T_Y, T_Z) between the *registered* images, with *new* scene depths $Z^{Reg}(x, y)$ defined by: $\frac{1}{Z^{Reg}(x,y)} = \frac{1}{Z(x,y)} - \frac{1}{Z_s(x,y)}$. Note that $Z^{Reg}(x, y)$ may also be negative, as opposed to the original scene.

In Fig. 5, the optical flow is displayed before and after registration of two frames according to the computed 2D motion parameters of the wall at the back of the scene. After registration the optical flow points towards the FOE.

Once the rotation is cancelled by the registration of the plane, the ambiguity between image motion caused by rotation and that caused by translation no longer exists. When only 3D translation exists, the induced image motion is directed towards the FOE (Focus of Expansion). The computation of the 3D translation therefore becomes a highly overdetermined and a numerically stable problem (as there are only two unknowns to the problem – the location of the FOE in the image plane).

a)  b) 

Figure 6: The inverse depth map.

a) First frame.
b) The obtained inverse depth map. Bright regions correspond to close objects. Dark regions correspond to distant objects. The depth was not computed near the the image boundaries.

4.3 Computing 3D Rotation

Assuming that the detected parametric surface is planar, (i.e., (X,Y,Z) lies on a planar surface in the 3D scene), it can be described by $Z = A+B\cdot X+C\cdot Y$. By perspective projection, this yields: $\frac{1}{Z} = \alpha+\beta\cdot x+\gamma\cdot y$ where: (x,y) are image coordinates, and $\alpha = \frac{1}{A}$, $\beta = -\frac{B}{f_c A}$, $\gamma = -\frac{C}{f_c A}$. Therefore, Eq. (1) can be rewritten as [1, 4]:

$$\begin{bmatrix} u_s \\ v_s \end{bmatrix} = \begin{bmatrix} a & + & b\cdot x & + & c\cdot y & + & g\cdot x^2 & + & h\cdot xy \\ d & + & e\cdot x & + & f\cdot y & + & g\cdot xy & + & h\cdot y^2 \end{bmatrix} \tag{3}$$

where:

$$\begin{array}{rclcrcl} a &=& -f_c\alpha T_X &-& f_c\Omega_Y & & \\ b &=& \alpha T_Z &-& f_c\beta T_X & & \\ c &=& \Omega_Z &-& f_c\gamma T_X & & \\ d &=& -f_c\alpha T_Y &+& f_c\Omega_X & & \end{array} \qquad \begin{array}{rclcrcl} e &=& -\Omega_Z &-& f_c\beta T_Y & & \\ f &=& \alpha T_Z &-& f_c\gamma T_Y & & \\ g &=& -\frac{\Omega_Y}{f_c} &+& \beta T_Z & & \\ h &=& \frac{\Omega_X}{f_c} &+& \gamma T_Z & & \end{array} \tag{4}$$

The parameters (a,b,c,d,e,f,g,h) are the 2D motion parameters of the detected 3D plane, computed as described in Sect. 2. Given these 2D motion parameters and the 3D translation parameters of the camera (T_X, T_Y, T_Z), then the 3D rotation parameters of the camera $(\Omega_X, \Omega_Y, \Omega_Z)$ (as well as the surface parameters (α,β,γ)) can be obtained by solving the set (4) of eight linear equations in six unknowns.

Experimental Results. The camera motion between Figure 5.a and Figure 5.b was: $(T_X, T_Y, T_Z) = (1.7cm, 0.4cm, 12cm)$ and $(\Omega_X, \Omega_Y, \Omega_Z) = (0°, -1.8°, -3°)$. The computation of the 3D motion parameters of the camera yielded: $(T_X, T_Y, T_Z) = (1.68cm, 0.16cm, 12cm)$ and $(\Omega_X, \Omega_Y, \Omega_Z) = (-0.05°, -1.7°, -3.25°)$. (The translation magnitude cannot be determined, only its direction. T_Z was therefore set to the correct size $12cm$, and the other parameters were then scaled accordingly).

Once the 3D motion parameters of the camera were computed, the 3D scene structure was reconstructed using a scheme similar to that suggested in [9]. In Fig. 6, the computed inverse depth map of the scene $(\frac{1}{Z(x,y)})$ is displayed.

5 Concluding Remarks

A method was presented for detecting and tracking several moving objects in dynamic scenes, using 2D parameterization. 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.

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

Detection of a single planar surface with its 2D motion parameters can also be used for computing the 3D motion parameters of a camera (the ego-motion) in a static scene in a robust way.

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] Y. Aloimonos and Z. Duric. Active egomotion estimation: A qualitative approach. In *European Conference on Computer Vision*, pages 497–510, Santa Margarita Ligure, May 1992.

[3] J.R. Bergen and E.H. Adelson. Hierarchical, computationally efficient motion estimation algorithm. *J. Opt. Soc. Am. A.*, 4:35, 1987.

[4] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, pages 237–252, Santa Margarita Ligure, May 1992.

[5] 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.

[6] 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.

[7] 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.

[8] R. Guissin and S. Ullman. Direct computation of the focus of expansion from velocity field measurements. In *IEEE Workshop on Visual Motion*, pages 146–155, Princeton, NJ, October 1991.

[9] K. Hanna. Direct multi-resolution estimation of ego-motion and structure from motion. In *IEEE Workshop on Visual Motion*, pages 156–162, Princeton, NJ, October 1991.

[10] B.K.P. Horn. Relative orientation. *International Journal of Computer Vision*, 4(1):58–78, June 1990.

[11] B.K.P. Horn and E.J. Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, 2(1):51–76, June 1988.

[12] 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.

[13] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical Models and Image Processing*, 53:231–239, May 1991.

[14] M. Irani and S. Peleg. Image sequence enhancement using multiple motions analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, Champaign, June 1992.

[15] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *To appear in International Journal of Computer Vision.*

[16] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *European Conference on Computer Vision*, pages 282–287, Santa Margarita Ligure, May 1992.

[17] A.D. Jepson and D.J. Heeger. A fast subspace algorithm for recovering rigid motion. In *IEEE Workshop on Visual Motion*, pages 124–131, Princeton, NJ, October 1991.

[18] 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.

[19] R.L. Lagendijk and J. Biemond. *Iterative Identification and Restoration of Images.* Kluwer Academic Publishers, Boston/Dordrecht/London, 1991.

[20] D.T. Lawton and J.H. Rieger. The use of difference fields in processing sensor motion. In *DARPA IUWorkshop*, pages 78–83, June 1983.

[21] S. Negahdaripour and S. Lee. Motion recovery from image sequences using first-order optical flow information. In *IEEE Workshop on Visual Motion*, pages 132–139, Princeton, NJ, October 1991.

[22] F. Lustman O.D. Faugeras and G. Toscani. Motion and structure from motion from point and line matching. In *Proc. 1st International Conference on Computer Vision*, pages 25–34, London, 1987.

[23] M.A. Taalebinezhaad. Direct recovery of motion and shape in the general case by fixation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14:847–853, August 1992.