Motion Analysis for Image Enhancement: Resolution, Occlusion, and Transparency*

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. In scenes having multiple moving objects the motion computation is performed together with object segmentation by using a unique temporal integration approach.

After computing the motion for the different image regions, these regions can be enhanced by fusing several successive frames covering the same region. Enhancements treated here include improvement of image resolution, filling-in occluded regions, and reconstruction of transparent objects.

1 Introduction

We describe methods for enhancing image sequences using the motion information computed by a multiple motions analysis method. The multiple moving objects are first detected and tracked, using both a large spatial region and a large temporal region, and without assuming any temporal motion constancy. The motion models used to approximate the motions of the objects are 2-D parametric motions in the image plane, such as affine and projective transformations. The motion analysis is presented in a previous paper [16, 17], and will only be briefly described here.

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 their images. When the scene contains transparent moving objects, they can be reconstructed separately.

*This research was supported by the Israel Academy of Sciences. Email address of authors: {michalb, peleg}@cs.huji.ac.il.

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

Section 2 includes a brief description of a method used for segmenting the image plane into differently moving objects, computing their motions, and tracking them throughout the image sequence. Sections 3, 4, and 5 describe the algorithms for image enhancement using the computed motion information: Section 3 presents a method for improving the spatial resolution of tracked objects, Section 4 describes a method for reconstructing occluded segments of tracked objects, and Section 5 presents a method for reconstructing transparent moving patterns.

An initial version of this paper appeared in [15].

2 Detecting and Tracking Multiple Moving Objects

In this section we describe briefly a method for detecting and tracking multiple moving objects in image sequences, which is presented in detail in [17]. Any other good motion computation method can be used as well. In this approach for detecting differently moving objects, a single motion is first computed, and a single object which corresponds to this motion is identified and tracked. We call this motion the *dominant motion*, and the corresponding object the *dominant object*. Once a dominant object has been detected and tracked, 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.

When the image motion can be described by a 2-D parametric motion model, and this model is used for motion analysis, the results are very accurate at a fraction of a pixel. This accuracy results from two features:

1. The use of large regions when trying to compute the 2-D motion parameters.
2. Segmentation of the image into regions, each containing only a single 2-D motion.

2.1 2-D Motion Models

2-D parametric transformations are used to approximate the projected 3-D motions of objects on 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.

Given two grey level images of an object, $I(x, y, t)$ and $I(x, y, t+1)$, it is assumed that:

$$I(x + p(x, y, t), y + q(x, y, t), t + 1) = I(x, y, t), \tag{1}$$

where $(p(x, y, t), q(x, y, t))$ is the displacement induced on pixel (x, y) by the motion of the planar object between frames t and $t + 1$. It can be shown [10] that the desired motion (p, q) minimizes the following error function at Frame t in the region of analysis R:

$$Err^{(t)}(p, q) = \sum_{(x,y)\in R} (pI_x + qI_y + I_t)^2. \tag{2}$$

We perform the error minimization over the parameters of one of the following motion models:

1. **Translation:** 2 parameters, $p(x, y, t) = a, \quad q(x, y, t) = d$. In order to minimize $Err^{(t)}(p, q)$, its derivatives with respect to a and d are set to zero. This yields two linear equations in the two unknowns, a and d. Those are the two well-known optical flow equations [4, 20], where every small window is assumed to have a single translation. In this translation model, the entire *object* is assumed to have a single translation.
2. **Affine**: 6 parameters, $p(x, y, t) = a + bx + cy, \quad q(x, y, t) = d + ex + fy$. Deriving $Err^{(t)}(p, q)$ with respect to the motion parameters and setting to zero yields six linear equations in the six unknowns: a, b, c, d, e, f [4, 5].

3. **Moving planar surface** (a pseudo projective transformation): 8 parameters [1, 4], $p(x,y,t) = a + bx + cy + gx^2 + hxy, \quad q(x,y,t) = d + ex + fy + gxy + hy^2$. Deriving $Err^{(t)}(p,q)$ with respect to the motion parameters and setting to zero, yields eight linear equations in the eight unknowns: a, b, c, d, e, f, g, h.

2.2 Detecting the First Object

When the region of support of a single object in the image is known, its motion parameters can be computed using a multiresolution iterative framework [3, 4, 5, 6, 16, 17].

Motion estimation is more difficult in the common case when the scene includes several moving objects, and the region of support of each object in the image is not known. It was shown in [7, 16, 17] that in this case the motion parameters of a *single* object can be recovered accurately by applying the same motion computation framework (with some iterative extensions [16, 17]) to the *entire* region of analysis.

This procedure computes a single motion (the *dominant* motion) between two images. A segmentation procedure is then used (see Section 2.5) in order to detect the corresponding object (the *dominant* object) in the image. An example of a detected dominant object using an affine motion model between two frames is shown in Figure 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 (Section 2.3).

2.3 Tracking Detected Objects Using Temporal Integration

The algorithm for the detection of multiple moving objects discussed in Section 2.2 can be extended to track detected objects throughout long image sequences. This is done by temporal integration, where for each tracked object a dynamic internal representation image is constructed. This image is constructed by taking a weighted average of recent frames, registered with respect to the tracked motion. This image contains, after a few frames, a sharp image of the tracked object, and a blurred image of all the other objects. Each new frame in the sequence is compared to the internal representation image of the tracked object rather than to the previous frame [16, 17]. Following is a summary of the algorithm for detecting and tracking an object in an image sequence:

For each frame in the sequence (starting at $t = 0$) do:

1. Compute the dominant motion parameters between the internal representation image of the tracked object $Av(t)$ and the new frame $I(t+1)$, in the region $M(t)$ of the tracked object (see Section 2.2). Initially, $M(0)$ is the entire region of analysis.
2. Warp the current internal representation image $Av(t)$ and current 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 (see Section 2.5), using the registered mask $M(t)$ as an initial guess. This will be the segmented region $M(t+1)$ of the tracked object in frame $I(t+1)$.

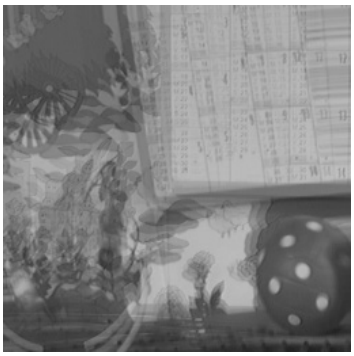
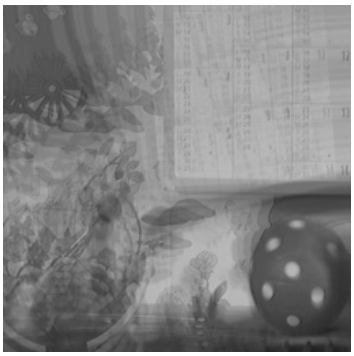
a)  b)  c) 

Figure 1: An example of the evolution of an internal representation image of a tracked object.
a) Initially, the internal representation image is the first frame in the sequence. The scene contains four moving objects. The tracked object is the ball.
b) The internal representation image after 3 frames.
c) The internal representation image after 5 frames. The tracked object (the ball) remains sharp, while all other regions blur out.

4. Compute the updated internal representation image $Av(t+1)$ by warping $Av(t)$ towards $I(t+1)$ using the computed dominant motion, and averaging it with $I(t+1)$.

When the motion model approximates of the temporal changes of the tracked object well enough, 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. Figure 1 shows an example of the evolution of an internal representation image of a tracked rolling ball. Comparing each new frame to the internal representation 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 Figure 2, temporal integration is used to detect and track the dominant object. Comparing the segmentation shown in Figure 2.c to the segmentation in Figure 2.d emphasizes the improvement in segmentation using temporal integration.

Another example of detecting and tracking objects using temporal integration is shown in Figure 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 all other regions in the image are blurred by their motion.

2.4 Tracking Other Objects

After detecting and tracking the first object, attention is directed at other objects. This is done by applying the tracking algorithm once more, this time to the rest of the image, after excluding the

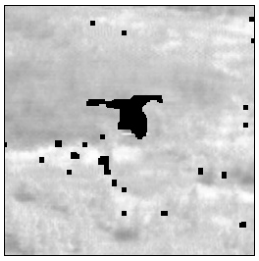
a)  b)  c)  d) 

Figure 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.

first detected object from the region of analysis. The scheme is repeated recursively, until no more objects can be detected.

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

a)  b)  c)  d) 

Figure 3: Detecting and tracking the dominant object in an infrared image sequence using temporal integration.
a-b) Two frames in the sequence. Both the background and the car are moving.
c) The internal representation 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.

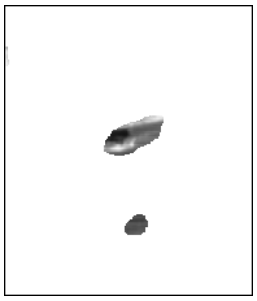
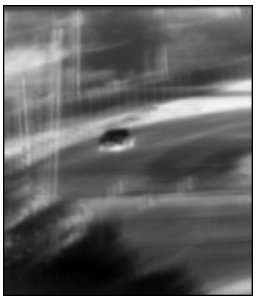
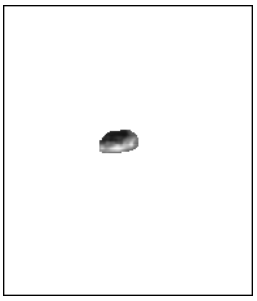
a)  b)  c) 

Figure 4: Detecting and tracking the second object using temporal integration.
a) The initial region of analysis after excluding the first dominant object (from Figure 3.d).
b) The internal representation image of the second tracked object (the car). The car remains sharp while the background blurs out.
c) Segmentation of the tracked car after 5 frames.

2.5 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 canceled after registration, and the tracked region is stationary in the registered images. The segmentation problem reduces therefore to identifying the stationary regions in the registered images.

Pixels are classified as moving or stationary using local analysis. The measure used for the classification is the average of the normal flow magnitudes over a small neighborhood of each pixel (typically a 3×3 neighborhood). In order to classify correctly large 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 when it conflicts the classification from the lower resolution level.

3 Improvement of Spatial Resolution

Once good motion estimation and segmentation of a tracked object are obtained, it becomes possible to enhance the images of this object.

Restoration of degraded images when a model of the degradation process is given is an ill-conditioned problem [2, 9, 11, 13, 19, 24]. 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 earlier method, which was presented in [14].

While earlier research on super-resolution [12, 14, 18, 25] treated only static scenes and pure translational motion in the image plane, we treat dynamic scenes and more complex 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 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 sensed 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. Finding f is the objective of the super-resolution algorithm.
- T_k is the 2-D geometric transformation from f to g_k, determined by the computed 2-D motion parameters of the tracked object in the image plane (not including the decrease in sampling rate between f and g_k). T_k is assumed to be invertible.
- h is a blurring operator, determined by the Point Spread Function of the sensor (PSF). 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 a downsampling 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 the pixel $g_k(m,n)$ 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 ${T_k}^{-1}((m,n))$.

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

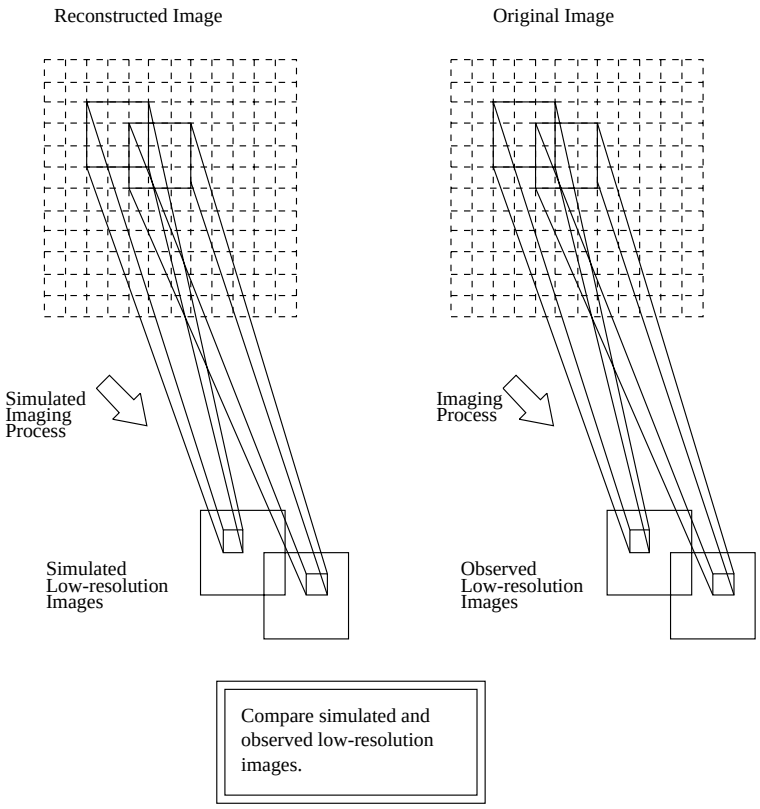


Figure 5: Schematic diagram of the super resolution algorithm. A reconstructed image is sought such that after simulating the imaging process, the simulated low-resolution images are closest to the observed low-resolution images. The simulation of the imaging process is expressed by Equation 3.

The Super-Resolution Algorithm. The presented algorithm for creating higher resolution images 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 $\{g_k^{(0)}\}_{k=1}^K$ corresponding to the observed input images $\{g_k\}_{k=1}^K$. If $f^{(0)}$ were the correct high resolution image, then the simulated images $\{g_k^{(0)}\}_{k=1}^K$ should be identical to the observed images $\{g_k\}_{k=1}^K$. The difference images $\{g_k - g_k^{(0)}\}_{k=1}^K$ are 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

$$e^{(n)} = \sqrt{\frac{1}{K}\sum_{k=1}^{K} \|g_k - g_k^{(n)}\|_2^2}$$

The algorithm is described schematically in Figure 5.

The imaging process of g_k at the n_{th} iteration is simulated by:

$$g_k^{(n)} = (T_k(f^{(n)}) * h) \downarrow s \tag{3}$$

where $\downarrow s$ denotes a downsampling operator by a factor s, and * is the convolution operator. The iterative update scheme of the high resolution image is expressed by:

$$f^{(n+1)} = f^{(n)} + \frac{1}{K} \sum_{k=1}^{K} {T_k}^{-1} \left(((g_k - g_k^{(n)}) \uparrow s) * p \right) \tag{4}$$

where K is the number of low resolution images, $\uparrow s$ is an upsampling operator by a factor s, and p is a "backprojection" kernel, determined by h and T_k as explained below. The average taking in Equation (4) reduces additive noise. The algorithm is numerically similar to common iterative methods for solving sets of linear equations [19], and therefore has similar properties, such as rapid convergence (see next paragraph).

In Figure 6, the resolution of a car's license plate was improved from 15 frames.

Analysis and Discussion. We introduce exact analysis of the superresolution algorithm in the case of deblurring: Restoring an image from K blurred images (taken from different viewing positions of the object), with 2-D *affine* transformations $\{T_k\}_{k=1}^{K}$ between them and the reconstruction viewing position, and *without* increasing the sampling rate. This is a special case of superresolution, which is simpler to analyze. In this case the imaging process is expressed by:

$$g_k^{(n)} = T_k(f^{(n)}) * h$$

and the restoration process in Equation (4) becomes:

$$f^{(n+1)} = f^{(n)} + \frac{1}{K} \sum_{k=1}^{K} {T_k}^{-1} \left((g_k - g_k^{(n)}) * p \right), \tag{5}$$

The following theorems show that the iterative super resolution scheme is an effective deblurring operator (proofs are given in the appendix).

Theorem 3.1 *The iterations of Equation (5) converge to the desired deblurred image f (i.e., an f that fulfills: $\forall k \;\; g_k = T_k(f) * h$), if the following condition holds:*

$$\|\delta - h * p\|_2 < \frac{1}{\frac{1}{K} \sum_{k=1}^{K} \|T_k\|_2} \tag{6}$$

where δ denotes the unity pulse function centered at $(0,0)$.
Remark: When the 2-D image motions of the tracked object consist of only 2-D translations and rotations, then Condition (6) reduces to $\|\delta - h * p\|_2 < 1$.

Proof: see appendix.

Theorem 3.2 *Given Condition (6), the algorithm converges at an exponential rate (the norm of the error converges to zero faster than q^n for some $0 < q < 1$),* regardless *of the choice of initial guess $f^{(0)}$.*

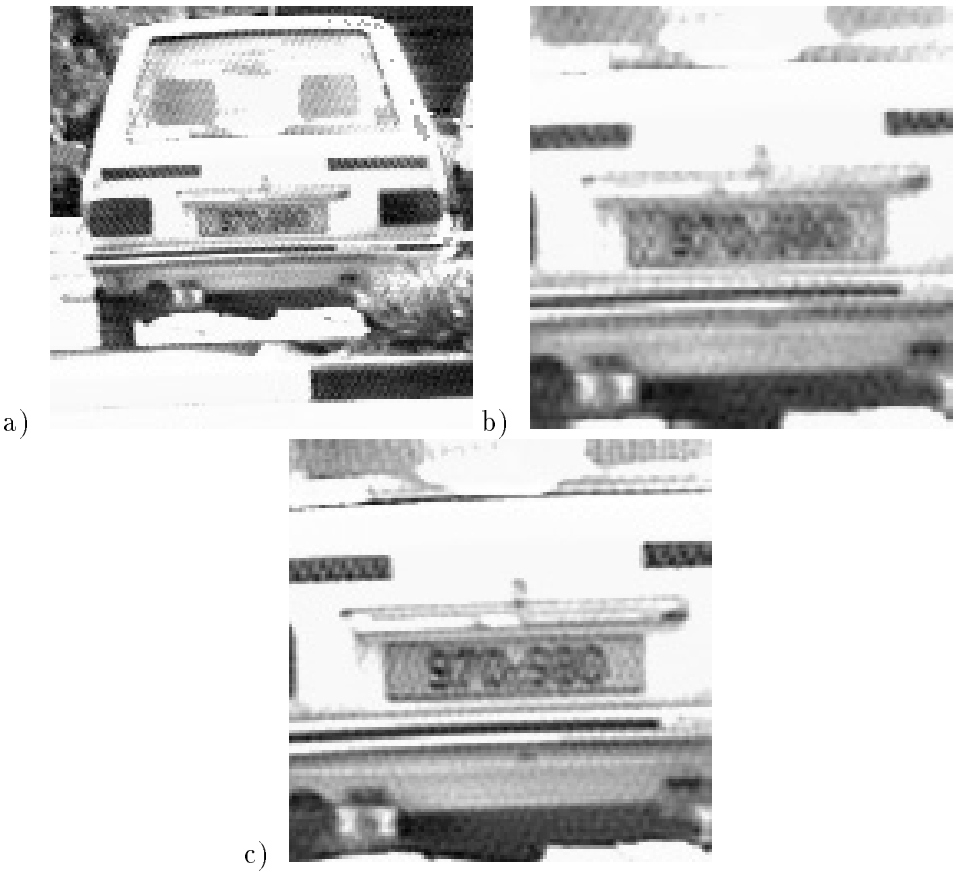


Figure 6: Improvement of spatial resolution using 15 frames. The sampling rate was increased by 2 in both directions.
a) The best frame from the image sequence.
b) The license plate magnified by 2 using bilinear interpolation.
c) The improved resolution image.

Proof: see appendix.

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. According to Theorem 3.2 the algorithm converges regardless of the choice of the initial guess. However, since there may be more than one correct solution to the problem, the choice of the initial guess does determine which of the solutions is reached. A good choice of the 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 feature.

Another issue is the choice of the backprojection kernel p. Unlike h, which represents properties of the sensor (the PSF), there is some freedom in the choice of p. p is chosen so that Condition (6)

holds. The smaller $\|\delta - h * p\|_2$ is, the faster the algorithm converges (see proof of Theorem 3.1). Ideally, if $\|\delta - h * p\|_2 = 0$, then the algorithm converges in a single iteration. This, however, means that p is the inverse kernel of h, which may not exist (as h is a low pass filter), or which may numerically be unstable to compute. Permitting $\|\delta - h * p\|_2 > 0$ (but still within the bounds of Condition (6)), allows p to be other than the exact inverse of h, and therefore increases the numerical stability, but slows down the speed of convergence. In other words, there is a tradeoff between the stability of the algorithm and its speed of convergence, determined by the choice of p.

The algorithm converges rapidly (usually within less than 5 iterations), and can be implemented on parallel machines. The complexity of the algorithm is low: $O(KN \log N)$ operations per iteration, where N is the number of pixels in the high resolution image f, and K is the number of low resolution images. Since the number of iterations is very small, this is also a good estimate of the complexity of the entire algorithm. The algorithm can be implemented in real-time, as only simple arithmetic operations are involved in the computation.

4 Reconstruction of Occlusions

When parts of a tracked object are occluded in some frames by another moving object, but these parts appear in other frames, a more complete view of the occluded object can be reconstructed [15, 26]. The image frames are registered using the computed motion parameters of the tracked object, and the occluded parts of that object are then reconstructed by temporally averaging gray levels of all pixels which were classified as object pixels in the corresponding segmentation masks. This process ignores the pixels when they are occluded by another moving object in the foreground, and the missing regions will be reconstructed even if they are occluded in most frames.

In the example shown in Figure 7, the background of the image sequence (the room scene) was completely reconstructed, eliminating the walking girl from the scene. The background was reconstructed in all frames, generating a new sequence with no trace of the moving girl.

a) b) c)

Figure 7: Reconstruction of occluded regions.
a) Five frames from the sequence. The camera is panning, and a girl walks from right to left. The girl appears in all frames and occludes parts of the background in each frame in the sequence.
b) Segmentation: black regions are those excluded from the tracked background.
c) Full reconstructions of the background in all frames. The girl is eliminated.

5 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, an object viewed through another transparent object, etc. In this section, we present a method for isolating and reconstructing tracked objects in transparent motion.

The presented scheme assumes additive transparency (such as in reflections). However, this scheme could be applied also to cases of multiplicative transparency (as in moving shadows and viewing through a semi-transparent media) by using the logarithm operation. Taking the logarithm of the input images changes the multiplicative effects into additive effects, and once the tracking is done, the exponent is taken to return to the original scale.

Previous analysis of transparency [6, 8, 21, 22, 23] assumed constant motion over several successive frames, which excludes most sequences taken from an unstabilized moving camera. Some methods [6, 21, 23] elegantly avoid the segmentation problem. They require, however, high order derivatives (the order increases with the number of objects), which make them sensitive to noisy data.

In our work we do not assume any motion constancy. We temporally integrate the image frames rather than use temporal derivatives. This provides robustness and numerical stability to the tracking algorithm. This approach not only tracks the moving transparent objects, but also reconstructs them.

Transparent motions yield several motion components at each point, and segmentation cannot be used to isolate one of the transparent objects. In practice, however, due to varying image contrast, in many image regions one object is more prominent than other objects, and 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 Section 2.3 to track the dominant transparent object. The temporal averaging restores the dominant transparent object in its internal representation image, while blurring out the other transparent objects, making them less noticeable. Comparing each new frame to the internal representation image of the tracked object rather than to the previous frame gives the algorithm a strong bias to keep tracking the same transparent object, as it is the only object in the internal image that is still similar to its image in the new frame (Figure 8).

For recovering the second transparent object, the temporal integration tracking technique is applied once more to the sequence, after some delay. Let $Av_1(t)$ denote the internal representation image of the first transparent object. Starting at frame $I(t)$, the algorithm is applied 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 out in the internal representation 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 Section 2.3. The tracking algorithm is applied once again to the *original* image sequence, and not to frame differences as in [6]. Now that the algorithm tracks the second dominant object, the new internal representation image $Av_2(t)$ restores the second dominant transparent object, and blurs out the other transparent objects, including the first dominant object.

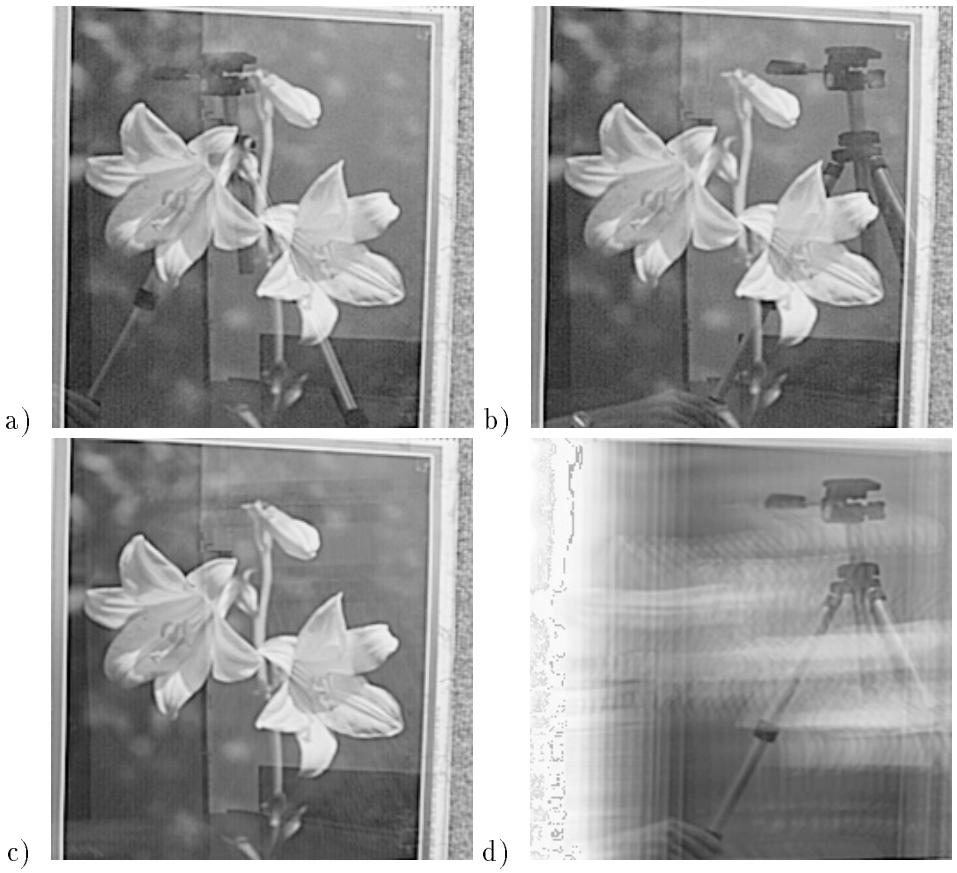


Figure 8: Reconstruction of "transparent" objects.
a-b) The first and last frames in a sequence. A moving tripod is reflected in the glass of a picture of flowers.
c) The internal representation image of the first tracked object (the picture of flowers) after 14 frames. The picture of flowers was reconstructed. The reflection of the tripod faded out.
d) The internal representation image of the second tracked object (the reflection of the tripod) after 14 frames. The reflection of the tripod was reconstructed. The picture of flowers faded out.

In Figure 8, the reconstruction of two transparent moving objects in a real image sequence is shown.

6 Concluding Remarks

Temporal integration of registered images proves to be a powerful approach to motion analysis, enabling human-like tracking of moving objects. Once good motion estimation and segmentation of a tracked object are obtained, it becomes possible to enhance the object images. Fusing information on tracked objects from several registered frames enables reconstruction of occluded regions, improvement of image resolution, and reconstruction of transparent moving objects.

APPENDIX

The appendix contains proofs of Theorems 3.1 and 3.2. The following notations will be used:

- $\mathbf{T}$ denotes the transformation from the deblurred image f to a blurred image g.
- $\tilde{\mathbf{T}}$ denotes the respective 2-D affine transformation describing the geometric transformation of *pixel* coordinates from f to g, i.e.,

  $$(\mathbf{T}(f))(x,y) = f\left(\tilde{\mathbf{T}}^{-1}(x,y)\right).$$

  Remarks:

  1. $\tilde{T}$ is assumed to be invertible.
  2. It is easy to show from this definition that T is a linear transformation of f.
- Since $\tilde{\mathbf{T}}$ is a 2-D affine transformation, it can be expressed in matrix notation by:

  $$\tilde{\mathbf{T}}(x,y) = \vec{\mathbf{d}} + \mathbf{M} \cdot \begin{pmatrix} x \\ y \end{pmatrix},$$

  where $\vec{\mathbf{d}}$ is a 2×1 vector, and $\mathbf{M}$ is a 2×2 matrix.
- $\tilde{\mathbf{M}}_{\mathbf{T}}$ denotes the linear transformation part of $\tilde{T}$ (on pixel coordinates), which uses the matrix M only (without the displacement $\vec{d}$), i.e.,

  $$\tilde{\mathbf{M}}_{\mathbf{T}}(x,y) = \mathbf{M} \cdot \begin{pmatrix} x \\ y \end{pmatrix},$$

  and respectively, $\mathbf{M_T}$ is the linear transformation on images defined by:

  $$(\mathbf{M_T}(f))(x,y) = f\left(\tilde{\mathbf{M}}_{\mathbf{T}}^{-1}(x,y)\right).$$
- $\mathbf{det}(\mathbf{M})$ denotes the determinant of the matrix M.

In order to prove Theorems 3.1 and 3.2, we introduce the following two lemmas:

Lemma 1

(1.a) $\|T\|_2 = |\det(M)|^{\frac{1}{2}}$

(1.b) $\|T^{-1}\|_2 = \frac{1}{|\det(M)|^{\frac{1}{2}}}$

(1.c) $\|M_T\|_2 = |\det(M)|^{\frac{1}{2}}$

(1.d) $\|M_{T^{-1}}\|_2 = \frac{1}{|\det(M)|^{\frac{1}{2}}}$

Remark: It follows from this lemma that whenever the affine transformation $\tilde{T}$ is limited to 2-D translations and rotations, then $\|T\|_2 = 1$. When the transformation contains a scaling by a factor s, then $\|T\|_2 = s$.

Proof:

(1.a)

$$\begin{aligned}
\|T(f)\|_2 &= \left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |\left(T(f)\right)(x,y)|^2\, dx\, dy\right)^{\frac{1}{2}} \\
&= \left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |f\left(\tilde{T}^{-1}(x,y)\right)|^2\, dx\, dy\right)^{\frac{1}{2}} \\
&= \left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |f(u,v)|^2 \cdot |\det(M)|\, du\, dv\right)^{\frac{1}{2}} \\
&\qquad\qquad \textbf{(change of integral variables by } (u,v) = \tilde{T}^{-1}(x,y) \textbf{)} \\
&= |\det(M)|^{\frac{1}{2}} \cdot \|f\|_2 \quad .
\end{aligned}$$

Therefore:

$$\begin{aligned}
\|T\|_2 &\stackrel{\text{def}}{=} Sup_{\|f\|_2=1}\left(\|T(f)\|_2\right) \\
&= Sup_{\|f\|_2=1}\left(|\det(M)|^{\frac{1}{2}} \cdot \|f\|_2\right) \\
&= |\det(M)|^{\frac{1}{2}} \cdot Sup_{\|f\|_2=1}\left(\|f\|_2\right) \\
&= |\det(M)|^{\frac{1}{2}} \quad . \qquad \blacksquare
\end{aligned}$$

(1.b)

It is easy to show that $\tilde{T^{-1}}(x,y) = \tilde{T}^{-1}(x,y) = -M^{-1}\cdot\vec{d} + M^{-1}\cdot\begin{pmatrix} x \\ y \end{pmatrix}$. Therefore, according to Lemma 1.a, $\|T^{-1}\|_2 = |\det(M^{-1})|^{\frac{1}{2}} = \frac{1}{|\det(M)|^{\frac{1}{2}}}$. $\blacksquare$

(1.c)

Since $\tilde{M}_T(x,y) = \mathbf{M}\cdot\begin{pmatrix} x \\ y \end{pmatrix}$, then according to Lemma 1.a $\|M_T\|_2 = |\det(M)|^{\frac{1}{2}}$. $\blacksquare$

(1.d)

It is easy to show that $M_{T^{-1}} = {M_T}^{-1}$. Therefore, according to Lemmas 1.b and 1.c, $\|M_{T^{-1}}\|_2 = \frac{1}{|\det(M)|^{\frac{1}{2}}}$. $\blacksquare$

Lemma 2

$$T(f_1 * f_2) = \frac{1}{{\|T\|_2}^2}\cdot\left(T(f_1) * M_T(f_2)\right)$$

where $*$ denotes the convolution operator.

Proof:

$$
\begin{aligned}
(T(f_1) * M_T(f_2))(x,y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (T(f_1))(\alpha,\beta)\cdot(M_T(f_2))(x-\alpha,y-\beta)\; d\alpha\; d\beta \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_1\left(\tilde{T}^{-1}(\alpha,\beta)\right)\cdot f_2\left(\tilde{M}_T^{-1}(x-\alpha,y-\beta)\right)\; d\alpha\; d\beta \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_1\left(\tilde{T}^{-1}(\alpha,\beta)\right)\cdot f_2\left(\tilde{M}_T^{-1}(x,y)-\tilde{M}_T^{-1}(\alpha,\beta)\right)\; d\alpha\; d\beta \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_1\left(\tilde{T}^{-1}(\alpha,\beta)\right)\cdot f_2\left(\tilde{T}^{-1}(x,y)-\tilde{T}^{-1}(\alpha,\beta)\right)\; d\alpha\; d\beta \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_1(\gamma,\delta)\cdot f_2\left(\tilde{T}^{-1}(x,y)-(\gamma,\delta)\right)\cdot|\det(M)|\; d\gamma\; d\delta \\
&\qquad\qquad \textbf{(change of integral variables by } (\gamma,\delta)=\tilde{T}^{-1}(\alpha,\beta)\textbf{)} \\
&= |\det(M)|\cdot(f_1 * f_2)(\tilde{T}^{-1}(x,y)) \\
&= |\det(M)|\cdot(T(f_1 * f_2))(x,y) \\
&= \|T\|_2{}^2\cdot(T(f_1 * f_2))(x,y) \qquad \textbf{(using Lemma 1.a)} \qquad \blacksquare
\end{aligned}
$$

Theorem 3.1
Let T_k denote the transformation from the deblurred image f to the blurred image g_k The iterations of Equation (5) converge to the desired deblurred image f (i.e., an f that fulfills: $\forall k \quad g_k = T_k(f) * h$), if the following condition holds:

$$\|\delta - h * p\|_2 < \frac{1}{\frac{1}{K}\sum_{k=1}^{K}\|T_k\|_2} \tag{6}$$

where δ denotes the unity pulse function centered at $(0,0)$.

Proof: Mathematical manipulations on the left hand side of Equation (5) yield:

$$
\begin{aligned}
f^{(n+1)} &= f^{(n)} + \frac{1}{K}\sum_{k=1}^{K} T_k{}^{-1}\left((g_k - g_k^{(n)}) * p\right) \\
&= \frac{1}{K}\sum_{k=1}^{K}\left(f^{(n)} + T_k{}^{-1}\left((g_k - g_k^{(n)}) * p\right)\right) \\
&= \frac{1}{K}\sum_{k=1}^{K} T_k{}^{-1}\left(T_k(f^{(n)}) + (g_k - g_k^{(n)}) * p\right) \\
&= \frac{1}{K}\sum_{k=1}^{K} T_k{}^{-1}\left(T_k(f^{(n)}) + g_k * p - g_k^{(n)} * p\right) \\
&= \frac{1}{K}\sum_{k=1}^{K} T_k{}^{-1}\left(T_k(f^{(n)}) + g_k * p - T_k(f^{(n)}) * h * p\right)
\end{aligned}
$$

$$= \frac{1}{K}\sum_{k=1}^{K} {T_k}^{-1}\left(T_k(f^{(n)}) * (\delta - h * p) + g_k * p\right)$$

Therefore,

$$f^{(n+1)} = \frac{1}{K}\sum_{k=1}^{K} {T_k}^{-1}\left(T_k(f^{(n)}) * (\delta - h * p) + g_k * p\right) \tag{7}$$

is another way of expressing the iterative scheme defined by Equation (5).

It is easy to show that the desired f is a fixed point of Equation (7), by replacing $f^{(n+1)}$ and $f^{(n)}$ with f, and g_k with $T_k(f) * h$. Therefore,

$$f = \frac{1}{K}\sum_{k=1}^{K} {T_k}^{-1}\left(T_k(f) * (\delta - h * p) + g_k * p\right). \tag{8}$$

We shall now show that $\lim_{n\to\infty} f^{(n)} = f$:

$$\begin{aligned}
\|Err^{(n+1)}\|_2 &= \|f^{(n+1)} - f\|_2 \\
&= \|\frac{1}{K}\sum_{k=1}^{K} {T_k}^{-1}\left(T_k(f^{(n)}) * (\delta - h * p) + g_k * p\right) - \\
&\qquad \frac{1}{K}\sum_{k=1}^{K} {T_k}^{-1}\left(T_k(f) * (\delta - h * p) + g_k * p\right)\|_2 \qquad \textbf{(using (7) and (8))} \\
&= \|\frac{1}{K}\sum_{k=1}^{K} {T_k}^{-1}\left(T_k(f^{(n)} - f) * (\delta - h * p)\right)\|_2 \\
&= \|\frac{1}{K}\sum_{k=1}^{K}\left(|det(M_k)| \cdot (f^{(n)} - f) * M_{{T_k}^{-1}}(\delta - h * p)\right)\|_2 \\
&\qquad \textbf{(using Lemma 2 and Lemma 1.b)} \\
&= \|(f^{(n)} - f) * \frac{1}{K}\sum_{k=1}^{K}\left(|det(M_k)| \cdot M_{{T_k}^{-1}}(\delta - h * p)\right)\|_2 \\
&\le \|f^{(n)} - f\|_2 \cdot \frac{1}{K}\sum_{k=1}^{K}\left(|det(M_k)| \cdot \|M_{{T_k}^{-1}}(\delta - h * p)\|_2\right) \\
&\le \|f^{(n)} - f\|_2 \cdot \frac{1}{K}\sum_{k=1}^{K}\left(|det(M_k)| \cdot \|M_{{T_k}^{-1}}\|_2 \|\delta - h * p\|_2\right) \\
&= \|Err^{(n)}\|_2 \cdot \|\delta - h * p\|_2 \cdot \frac{1}{K}\sum_{k=1}^{K}\left(|det(M_k)| \cdot \|M_{{T_k}^{-1}}\|_2\right) \\
&= \|Err^{(n)}\|_2 \cdot \|\delta - h * p\|_2 \cdot \frac{1}{K}\sum_{k=1}^{K}\left(|det(M_k)| \cdot \frac{1}{|det(M_k)|^{\frac{1}{2}}}\right) \qquad \textbf{(using Lemma 1.d)} \\
&= \|Err^{(n)}\|_2 \cdot \|\delta - h * p\|_2 \cdot \frac{1}{K}\sum_{k=1}^{K} |det(M_k)|^{\frac{1}{2}}
\end{aligned}$$

$$\vdots \qquad \textbf{(unfolding the recursion)}$$

$$\leq \quad \|Err^{(0)}\|_2 \cdot \left(\|\delta - h * p\|_2 \cdot \frac{1}{K} \sum_{k=1}^{K} |det(M_k)|^{\frac{1}{2}} \right)^{n+1}$$

According to Condition (6)) and Lemma 1.a

$$\|\delta - h * p\|_2 \cdot \frac{1}{K} \sum_{k=1}^{K} |det(M_k)|^{\frac{1}{2}} < 1 \quad ,$$

therefore

$$\lim_{n \to \infty} \left(\|\delta - h * p\|_2 \cdot \frac{1}{K} \sum_{k=1}^{K} |det(M_k)|^{\frac{1}{2}} \right)^{n+1} = 0 \quad , \tag{9}$$

and therefore

$$\lim_{n \to \infty} \|Err^{(n)}\|_2 = 0 \quad .$$

This proves that $\lim_{n \to \infty} f^{(n)} = f$.

Remark: When the 2-D image motions of the tracked object consist of only 2-D translations and rotations, then Condition (6) reduces to $\|\delta - h * p\|_2 < 1$. The reason for this is that $|\det(M_k)| = 1$ for such affine transformations $\tilde{T}_k$, and therefore, according to Lemma 1.a : $\|T_k\|_2 = 1$. ∎

Theorem 3.2
Given Condition (6), the algorithm converges at an exponential rate (the norm of the error converges to zero faster than q^n for some $0 < q < 1$), regardless *of the choice of initial guess $f^{(0)}$.*

Proof: Equation (9) confirms the exponential speed of convergence. The proof of Theorem 3.1 shows that $\lim_{n \to \infty} \|Err^{(n)}\|_2 = 0$ *regardless* of the magnitude $\|Err^{(0)}\|_2$, and therefore the choice of the initial guess $f^{(0)}$ does not affect the convergence. ∎

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] H.C. Andrews and B.R. Hunt, editors. *Digital Image Restoration.* Prentice Hall, 1977.

[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] 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.

[9] R.C. Gonzalez. Image enhancement and restoration. In T.Y. Young and K.S. Fu, editors, *Handbook of Pattern Recognition and Image Processing*, pages 191–213. Academic Press, 1986.

[10] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[11] T.S. Huang, editor. *Image Enhancement and Restoration.* JAI Press, 1986.

[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] R.A. Hummel, B. Kimia, and S.W. Zucker. Deblurring gaussian blur. *Computer Vision, Graphics, and Image Processing*, 38:66–80, 1986.

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

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

[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] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *To appear in International Journal of Computer Vision*, 1993.

[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] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130, 1981.

[21] M. Shizawa. On visual ambiguities due to transparency in motion and stereo. In *European Conference on Computer Vision*, pages 411–419, Santa Margarita Ligure, May 1992.

[22] 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.

[23] M. Shizawa and K. Mase. Principle of superposition: A common computational framework for analysis of multiple motion. In *IEEE Workshop on Visual Motion*, pages 164–172, Princeton, New Jersey, October 1991.

[24] H. Shvayster and S. Peleg. Inversion of picture operators. *Pattern Recognition Letters*, 5:49–61, 1985.

[25] H. Ur and Gross D. Improved resolution from subpixel shifted pictures. *CVGIP: Graphical Models and Image Processing*, 54:181–186, 1992.

[26] J. Wang and E. Adelson. Layered representation for motion analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–366, New York, June 1993.