Super Resolution From Image Sequences

Michal Irani and Shmuel Peleg*

Department of Computer Science
The Hebrew University of Jerusalem
91904 Jerusalem, Israel

Abstract

Image resolution can be improved when the relative displacements in image sequences is known accurately, and some knowledge of the imaging process is available. The proposed approach is similar to back-projection used in tomography. Examples of improved image resolution are given, with computation of the unknown image displacements.

1   Introduction

Image resolution depends on the physical characteristics of the sensor: the optics and the density and spatial response of the detector elements. Increasing the resolution by sensor modification may not be an available option. An increase in the sampling rate could, however, be achieved by obtaining more samples of the scene from a sequence of displaced pictures. An estimate of the sensor's spatial response helps obtain a sharper image.

An iterative algorithm to increase image resolution is described in this paper. Examples are shown for low resolution gray-level pictures, with an increase of resolution clearly observed after only a few iterations. The same method can also be used for deblurring a single blurred image.

Earlier research on super resolution was carried out by Tsai and Huang [6], who used frequency domain methods. Their work disregarded the blur in the imaging process, and only attempted to handle loss of data due to decimation by using translated images.

Gross [3] assumed that the imaging process is known, and that the relative shifts of the input pictures are known precisely. By merging the low resolution pictures over a finer grid using interpolation, he obtained a single blurred picture of higher spatial sampling rate. The merged picture was then deblurred by convolving it with a restoration filter, obtained by applying pseudo-inverse techniques to a matrix representing the blur operator. Similar to the work of Tsai and Huang [6], only translations are considered.

Peleg and Keren [11, 9] estimated an initial guess for

the higher resolution image, and simulated the imaging process (assumed to be known) to obtain a set of simulated low resolution images. They defined an error function between the actual and simulated low resolution images, which they minimized iteratively until no further improvement was obtained, or until the maximum number of allowed iterations was reached. This method gave good results for noise-free images, but was highly sensitive to noise and slow to converge.

The approach described in this paper is based on the resemblance of the presented problem to the reconstruction of a 2-D object from its 1-D projections in Computer-Aided Tomography (CAT) [4]. In tomography, images are reconstructed from their projections in many directions. In the super resolution case, each low resolution pixel is a "projection" of a region in the scene whose size is determined by the imaging blur. The high resolution image is constructed using an approach similar to the back-projection method used in CAT.

Accurate knowledge of the relative scene locations sensed by each pixel in the observed images is necessary for super resolution. This information is available in image regions where local deformation can be described by some parametric function. Such functions can describe, for example, perspective transformation. In this paper we assume that local motion can be described by translations and rotations only, but the approach is applicable also for other image motion models.

The imaging process, yielding the observed image sequence $\{g_k\}$, is modeled by:

$$g_k(m,n) = \sigma_k(h(f(x,y)) + \eta_k(x,y)),$$

where

- g_k is the k_{th} observed image frame,
- f is the original scene,
- h is a blurring operator,
- η_k is an additive noise term,
- σ_k is a non linear function which digitizes and decimates the image into pixels and quantizes the resulting pixels values from intensities into gray levels. σ_k also includes the displacement of the k_{th} frame,
- (x,y) is the center of the receptive field (in f) of the detector whose output is $g_k(m,n)$.

The receptive field (in f) of a detector whose output is $g_k(m,n)$ is defined uniquely by its center (x,y) and its

shape. The shape is determined by the region of support of the blurring operator h. Under the assumption that the displacement is a combination of translations and rotations, as done throughout this paper, the scene location (x, y) of the center of the receptive field for the observed location (m, n) is computed by:

$$x = x_k^0 + s_x \, m \cos \theta_k - s_y \, n \sin \theta_k \qquad (1)$$
$$y = y_k^0 + s_x \, m \sin \theta_k + s_y \, n \cos \theta_k,$$

where

- (x_k^0, y_k^0) is the translation of the k_{th} frame,
- θ_k is the rotation of the k_{th} frame about the origin,
- s_x and s_y are the sampling rates in the x and y directions respectively.

The algorithm presented in this paper attempts to reconstruct a higher resolution image, $\hat{f}$, which approximates f as accurately as possible. It is assumed that the acceleration of the camera while imaging a single frame is negligible.

2   The Imaging Process

This section describes the two preliminary tasks of obtaining the parameters of the imaging process. The relative displacements of the input images at subpixel accuracy are computed, as well as the blur in the imaging process.

2.1   Image Registration

Keren and Peleg [9] used the following method, based on [10], which has been found to be the most accurate for our purposes. Horizontal shift a, vertical shift b and rotation angle θ between images g_1 and g_2 can be written as:

$$g_2(x, y) = g_1(x \cos \theta - y \sin \theta + a, \ y \cos \theta + x \sin \theta + b).$$

It has been shown [9] that solving the following equation for (a, b, θ) minimizes the difference between the image g_2 and the image g_1 warped by (a, b, θ):

$$\sum g_x{}^2 a + \sum g_x g_y b + \sum A g_x \theta = \sum g_x g_t,$$
$$\sum g_x g_y a + \sum g_y{}^2 b + \sum A g_y \theta = \sum g_y g_t, \qquad (2)$$
$$\sum A g_x a + \sum A g_y b + \sum A^2 \theta = \sum A g_t,$$

where $g_x = \frac{\partial g_1}{\partial x}$, $g_y = \frac{\partial g_1}{\partial y}$, $g_t = g_2 - g_1$, and $A = x g_y - y g_x$. The motion parameters a, b and θ will be computed by solving this set of linear equations. The above equations were obtained under assumptions which are valid only for small displacements.

2.2   Iterative Refinement

As images are recorded in discrete time intervals, the displacements between them may not be sufficiently small for the motion recovery method of Equations (2). We therefore iterate the following process for two given images g_1 and g_2 [9]:

1. Initially assume no motion between the frames.
2. Compute approximations to the motion parameters by solving Equations (2). Add the computed motion to the existing motion estimate.
3. Warp frame g_2 towards g_1 using the current motion estimates, and return to Step 2 with the warped image g_2.

g_2 gets closer to g_1 at every iteration, and as the residual corrections to (a, b, θ) computed in Step 2 get smaller, the motion parameters become more accurate. The process terminates when the corrections to (a, b, θ) approach zero.

Since frame g_1 remains unchanged, nine of the twelve coefficients in the set of equations are computed only once, and only three coefficients, depending on g_2, need to be recomputed every iteration. This saves time in the iterative process.

In order to speed up the process and improve accuracy, a Gaussian Pyramid data structure is used [12]. First, the motion parameters are computed for a reduced resolution image in the pyramid, where even large translations become small. The computed motion parameters are then interpolated into a larger image, the motion estimate is corrected through a few iterations, and again interpolated to the next resolution level. This process is continued until the original full-size image is reached.

2.3   Recovering the Blur

Some knowledge of the digitization process is necessary to simulate the imaging process. Images used in our experiments were imaged by a flatbed scanner, and its blurring function was evaluated by scanning a small white dot on a black background.

When the imaging process cannot be applied to control images, similar to the above mentioned white dot, the blur can be estimated from the degradation of features that are originally small points or sharp edges.

3   Super Resolution

In this section the super resolution algorithm is described in detail, with noise and stability analysis. Experimental results are also given.

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 "back-projecting" each value in the difference images onto its receptive field in $f^{(0)}$. This process is repeated iteratively to minimize the error function

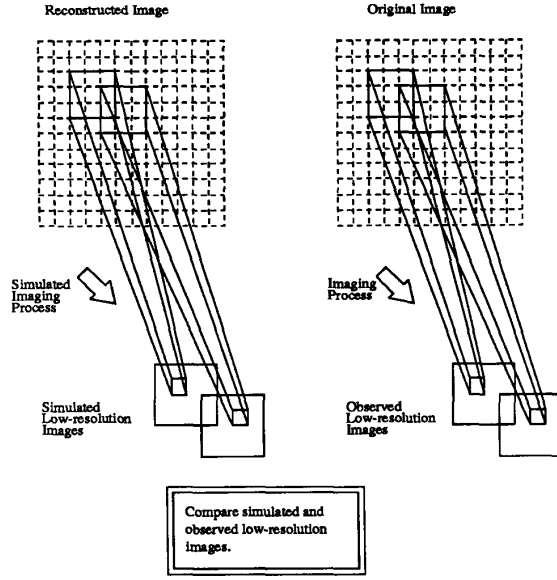$$e^{(n)} = \sqrt{\sum_k \sum_{(x,y)} (g_k(x, y) - g_k^{(n)}(x, y))^2}.$$

Figure 1: Schematic diagram of the super resolution algorithm.

The algorithm is described schematically in Figure 1.

Definition 3.1 *A low resolution pixel $\vec{y}$ is* influenced *by a high resolution pixel $\vec{x}$, if $\vec{x}$ is in $\vec{y}$'s receptive field.*

Definition 3.2 *A low resolution image g is* influenced *by a high resolution pixel $\vec{x}$, if g contains a pixel $\vec{y}$ such that $\vec{y}$ is influenced by $\vec{x}$.*

The following notation is used:

- f – The target high resolution image to be constructed (unknown),
- $f^{(n)}$ – The approximation of f obtained after n iterations,
- g_k – The k_{th} observed low resolution image,
- $g_k^{(n)}$ – The low resolution image obtained by applying the simulated imaging process to $f^{(n)}$. If $f^{(n)}$ is the correct high resolution, we expect $g_k^{(n)} = g_k$,
- h^{PSF} – The point spread function of the imaging blur,
- h^{BP} – A back-projection kernel (the choice of h^{BP} is referred to later),
- $\vec{x}$ denotes a high resolution pixel,
- $\vec{y}$ denotes a low resolution pixel (influenced by $\vec{x}$).

Let $\vec{z}_{\vec{y}}$ denote the center of the receptive field of $\vec{y}$ in $f^{(n)}$, computed by (1). The imaging process is then expressed by:

$$g^{(n)}(\vec{y}) = \sum_{\vec{x}} f^{(n)}(\vec{x}) h^{PSF}(\vec{x} - \vec{z}_{\vec{y}}).$$

The iterative update scheme to estimate the high resolution image f is expressed by:

$$f^{(n+1)}(\vec{x}) = f^{(n)}(\vec{x})$$
$$+ \sum_{\vec{y} \in \cup_k Y_{k,\vec{x}}} (g_k(\vec{y}) - g_k^{(n)}(\vec{y})) \frac{(h_{\vec{x}\vec{y}}^{BP})^2}{c \sum_{\vec{y}' \in \cup_k Y_{k,\vec{x}}} h_{\vec{x}\vec{y}'}^{BP}} \quad (3)$$

where

- $Y_{k,\vec{x}}$ denotes the set $\{\vec{y} \in g_k \mid \vec{y} \text{ is influenced by } \vec{x}\}$,
- c is a (constant) normalizing factor,
- $h_{\vec{x}\vec{y}}^{BP} = h^{BP}(\vec{x} - \vec{z}_{\vec{y}})$.

Equation (3) computes the following: the value of $f^{(n)}$ at each high resolution pixel $\vec{x}$ is updated according to all low resolution pixels $\vec{y}$ which it influences. The contribution of the low resolution pixel $\vec{y}$ of an input image g_k is the error $(g_k(\vec{y}) - g_k^{(n)}(\vec{y}))$ multiplied by a factor of $\frac{h_{\vec{x}\vec{y}}^{BP}}{c}$. 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 generated by the various low resolution pixels are combined by taking their weighted average, using the coefficients of h^{BP} as weights.

It is important to bear in mind 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. Therefore, there are several possible solutions, and the algorithm may either converge to one of them, or oscillate among some of them. The choice of initial guess does not influence the performance of the algorithm (speed or stability). It may, however, influence which of the possible solutions is reached first. A good choice of initial guess is the average of the low resolution images. The average image is computed by registering all the low resolution images over a fixed finer grid. Each high-resolution pixel in the fine grid is taken to be the average of all the low resolution pixels stacked above it. Such an initial guess leads the algorithm to a smooth solution, which is usually a desired one.

Another issue is the choice of h^{BP}. Unlike h^{PSF}, which represents properties of the sensor, h^{BP} can be chosen arbitrarily. A possible choice of h^{BP} is $h^{BP} = h^{PSF}$, but a closer observation of the mathematical analysis shows that more than one choice of h^{BP} may lead to convergence. The choice of h^{BP} does, however, affect the characteristics of the solution reached when there are several possible solutions. h^{BP} may therefore be utilized as an additional constraint, so that the solution reached is smooth, or has other desired properties. Additional considerations for the choice of h^{BP} appear in Section 4.

This super resolution algorithm performs well both on real and computer simulated images. Improvement in resolution is clearly observed even when a very small number of low resolution images are available. The algorithm converges rapidly (usually within less than 5 iterations), and is very stable. The complexity of the algorithm is low: $O(KN \min\{M, \log N\})$ operations per iteration, where N is the size of the high resolution image f, M is the size of the blurring kernel and K is the number of low resolution pictures. Since the number of iterations is very small, this is also a good estimate of the complexity of the complete algorithm. Proof of convergence at an exponential rate is given in [8] for the special (and simpler) case of deblurring. The algorithm has parallel characteristics: the contributions (to be averaged) of the low resolution pixels to the high resolution pixels, within a single iteration, may all be computed independently. Synchronization is needed only at the end of each iteration, when the values have to be averaged to obtain the new value.

Figure 2 shows the result of applying the algorithm to three low resolution images recorded by a scanner and translated relative to each other. The sampling rate was doubled in both directions. The relative displacements were computed as described in Section 2.1, and the blur was estimated as described in Section 2.3 .

4 Deblurring

Restoration of degraded images when a model of the degradation process is given, is considered as an ill-conditioned problem [1, 2, 5, 7, 13]. In this section deblurring of a single image is shown to be a special case of super resolution, and convergence conditions of the algorithm with stability analysis are given for this case. Deblurring a single image is achieved by applying the algorithm to a single input image, without increasing the sampling rate. Equation (3) then reduces to

$$f^{(n+1)}(\vec{x}) = f^{(n)}(\vec{x}) + \sum_{g}(g(\vec{y}) - g^{(n)}(\vec{y}))\frac{(h^{BP}(\vec{x} - \vec{y}))^2}{c}.$$

Using convolutions, this can be rewritten as:

$$f^{(n+1)} = f^{(n)} + (g - g^{(n)}) * \frac{h^{AUX}}{c}, \qquad (4)$$

where

- $*$ is the convolution operator.
- $(h^{BP})^2 = h^{AUX}$.

When the image blur is expressed by:

$$g = f * h^{PSF}, \qquad (5)$$

the following theorems show that the iterative super resolution scheme is an effective deblurring operator. Full proofs can be found in [8].
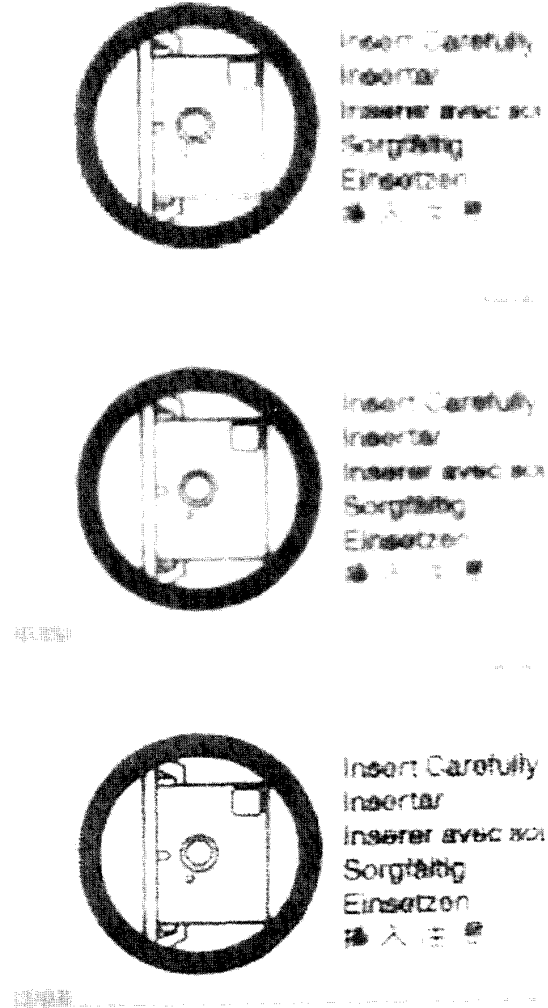

Figure 2: Super resolution from three input real images.

a) One of three original images.
b) Initial guess - average of the three input images after registration.
c) Improved resolution image.

Figure 3: Deblurring a synthetically blurred image.

a) Original image.
b) Blurred image.
c) Restored image.

Figure 4: Deblurring a real input image, with an arbitrary image as an initial guess.

a) Input image.
b) An arbitrary image used as the initial guess.
c) Deblurred image.

Theorem 4.1 *Let H^{PSF} and H^{AUX} denote the Fourier transforms of h^{PSF} and h^{AUX}, respectively. If:*

$$\forall \bar{\omega} \quad 0 < \frac{H^{PSF}(\bar{\omega})H^{AUX}(\bar{\omega})}{c} < 1 \qquad (6)$$

then the iterations converge to the original image f .

It is clear that condition (6) does not hold for frequencies $\bar{\omega}$ for which $H^{PSF}(\bar{\omega}) = 0$. Those frequencies are completely lost and cannot be restored by any method. For any other $\bar{\omega}$, however, h^{AUX} and c may be chosen so that condition (6) is fulfilled (recall that we are dealing with a finite space), and the original frequency is fully restored. The behavior of frequencies $\bar{\omega}$ for which $H^{PSF}(\bar{\omega}) = 0$ will be examined later.

Theorem 4.2 *Given condition (6), the algorithm converges at an exponential rate[†], regardless of the choice of initial guess $f^{(0)}$.*

One of the main benefits of the algorithm is in the freedom of choice of the auxiliary filter h^{AUX} and the normalizing constant c, so that condition (6) holds for as many frequencies as possible, ensuring optimal conditions for convergence. A sensible choice of h^{AUX} and c increases numerical stability. The further the term $\frac{H^{PSF}(\bar{\omega})H^{AUX}(\bar{\omega})}{c}$ is from its boundary limits (0 and 1), the more accurate the restored frequency is, and the less sensitive it is to noise and errors. Since the speed of convergence is accelerated as the term approaches 1, there is a tradeoff between stability and speed of convergence.

It should be emphasized that even if condition (6) in Theorem 4.1 does not hold for all frequencies $\bar{\omega}$, the proof still holds for the frequencies that fulfill this condition.

Following is an examination of the behavior of the algorithm when condition (6) does not hold [8]. This happens only when $H^{PSF}(\bar{\omega}) = 0$, as in any other case $H^{AUX}(\bar{\omega})$ and c can be chosen so that condition (6) does hold. An arbitrary choice of $H^{AUX}(\bar{\omega})$ causes $F^{(n)}(\bar{\omega})$ to diverge as $n \rightarrow \infty$. However, if $H^{AUX}(\bar{\omega}) = 0$, then $\forall n \; F^{(n)}(\bar{\omega}) = F^{(0)}(\bar{\omega})$, that is, the algorithm preserves the component of $\bar{\omega}$ in the initial guess $f^{(0)}$. This is one of the reasons for using the average of the input images as the initial guess.

[†]*i.e., the norm of the error converges to zero faster than q^n for some $0 < q < 1$.*

A good choice of h^{AUX} is usually not unique. In the common case of a real and symmetric h^{PSF}, a possible choice of h^{AUX} is $h^{AUX} = h^{PSF}$. This is a good choice, because for a real and symmetric h^{PSF}, H^{PSF} is also a real function, therefore $\forall \bar{\omega} \; H^{PSF}(\bar{\omega})H^{AUX}(\bar{\omega}) = (H^{PSF}(\bar{\omega}))^2 \geq 0$. This means that for all non-zero frequency components, condition (6) holds if c is sufficiently large. Stability is achieved in this case because as $H^{PSF}(\bar{\omega})$ tends to zero, so does $H^{AUX}(\bar{\omega})$. This prevents such frequency components from varying much in a few iterations, hence remaining similar to their initial value in $f^{(0)}$. For the same reason, noise is not amplified by such frequency components.

This method, therefore, has the advantage of being stable even in neighborhoods of zero-valued frequency components of the blur. This compares well with other deblurring methods, such as inverse filtering, which tend to amplify noise.

Figure 3 shows the result of deblurring an image which was blurred synthetically by convolving it with a 7×7 blurring kernel.

Figure 4 shows the result of deblurring an image scanned by a flatbed scanner. The blurring function in this case was the measured point spread function of the scanner. The initial guess was some arbitrary image, to show the small effect of the initial guess on the result.

5   Heuristic Improvements

5.1   Fixing Stable Pixels

To increase speed and stability of the algorithm, a high resolution pixel is fixed when it is assigned the same value (or nearly the same value) two successive iterations. This pixel will not be considered again in future iterations.

This fixing process increases the speed of convergence, since at later iterations fewer pixels are examined. It also prevents harmful salt-and-pepper type of noise from contaminating large surrounding areas. It is unlikely that a noisy pixel will be fixed since its gray level usually differs from those of its neighboring pixels.

5.2   Noise Reduction

Referring back to the updating scheme of Equation (3), the new value of a high resolution pixel in each iteration is

119

Figure 5: Super resolution from ten noisy images.

a) One of the ten low-resolution images
b) Initial guess - average of the low-resolution images after registration
c) Improved resolution image

computed by taking the average of all contributions of the various low resolution pixels. Taking an average in itself already handles additive noise. In order to handle multiplicative noise as well, contributions having extreme (high and low) values are eliminated before taking the average. Only contributions whose values are neither maximal nor minimal are averaged, eliminating both additive and multiplicative noise. For such noise cleaning a reasonable number of low resolution images is needed.

Figure 5 shows the result of applying the algorithm to ten low resolution images, contaminated by zero-mean gaussian noise of standard deviation 10. The sampling rate was doubled in each direction.

6   Concluding Remarks

Super resolution is shown to be feasible for image sequences, when the relative displacements can be computed accurately, and with approximate knowledge of the imaging process.

An iterative algorithm for computing super resolution has been presented. It was shown that when the algorithm is applied to a single image without increasing the sampling rate, super resolution reduces to deblurring.

The suggested algorithm performed well for both computer-simulated and real images, and has been shown, theoretically and practically, to converge quickly. The algorithm can be executed in parallel for faster hardware implementation.

Accurate knowledge of the relative displacements of scene regions is essential for using image sequences. In this paper we have assumed a simple uniform motion of translation and rotation for the entire image, and implemented an accurate method for computing this displacement. This method, however, can also be applied to other types of motion, such as perspective transformation, multiple motions in the image, etc. As long as the image can be divided into regions such that each region undergoes some uniform motion, resolution can be improved in the regions.

All images used in this paper were digitized using uniform sampling. This is, however, not necessary; the process can be applied to images sampled in any arbitrary nonuniform manner. Samples not on a uniform sampling grid can be accommodated, as well as blur which varies between sample locations.

References

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

[2] 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.

[3] D. Gross. Super resolution from sub-pixel shifted pictures. Master's thesis, Tel-Aviv University, October 1986.

[4] D.A. Hayner and W.K. Jenkins. The missing cone problem in computer tomography. In T.S. Huang, editor, *Advances in Computer Vision and Image Processing*, volume 1, pages 83–114. JAI Press Inc., 1984.

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

[6] 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.

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

[8] M. Irani and S. Peleg. Super resolution from image sequences. Technical Report 89-7, Dept. of Computer Science, The Hebrew University of Jerusalem, June 1989.

[9] D. Keren, S. Peleg, and R. Brada. Image sequence enhancement using sub-pixel displacements. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 742–746, Ann Arbor, Michigan, June 1988.

[10] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, August 1981.

[11] S. Peleg, D. Keren, and L. Schweitzer. Improving image resolution using subpixel motion. *Pattern Recognition Letters*, pages 223–226, 1987.

[12] A. Rosenfeld, editor. *Multiresolution Image Processing and Analysis*. Springer-Verlag, 1984.

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