CVGIP: GRAPHICAL MODELS AND IMAGE PROCESSING
Vol. 53, No. 3, May, pp. 231–239, 1991

Improving Resolution by Image Registration

MICHAL IRANI AND SHMUEL PELEG*

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

Communicated by Rama Chellappa

Received June 16, 1989; accepted May 25, 1990

Image resolution can be improved when the relative displacements in image sequences are 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 for gray-level and color images, when the unknown image displacements are computed from the image sequence. © 1991 Academic Press, Inc.

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, together with a method for image registration with subpixel accuracy, is presented in this paper. Examples are shown for low-resolution gray-level and color images, with an increase in resolution clearly observed after only a few iterations. The same method can also be used for deblurring a single blurred image.

Earlier research on superresolution was carried out by Tsai and Huang [10], who used frequency domain methods. Their work disregarded the blur in the imaging process; they used translated images to handle loss of data due to decimation.

Gross [6] 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. As in the work of Tsai and Huang [10], only translations were considered.

Peleg and co-workers [16, 12] 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 the 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) [7]. In tomography, images are reconstructed from their projections in many directions. In the superresolution 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 superresolution. 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 an observed monochrome image sequence $\{g_k\}$, is modeled by

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

* Part of this work was done while this author was with David Sarnoff Research Center, Princeton, NJ 08543. This research was supported by a grant from the Israeli National Council for Research and Development.

1049-9652/91 $3.00

where

- g_k is the kth observed image frame,
- f is the original scene,
- h is a blurring operator,
- η_k is an additive noise term,
- σ_k is a nonlinear function which digitizes and decimates the image into pixels and quantizes the resulting pixel values from intensities into gray levels. σ_k also includes the displacement of the kth 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. Assuming 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

$$\begin{aligned} x &= x_k^0 + s_x m \cos\theta_k - s_y n \sin\theta_k \\ y &= y_k^0 + s_x m \sin\theta_k + s_y n \cos\theta_k, \end{aligned} \tag{1}$$

where

- (x_k^0, y_k^0) is the translation of the kth frame,
- θ_k is the rotation of the kth 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, $\bar{f}$, which approximates f as accurately as possible. It is assumed that the acceleration of the camera while imaging a single frame is negligible.

When enhancing the resolution of color images, it is suggested that the YIQ representation [15] be used. The monochrome superresolution algorithm may then be applied separately to each component, where the gray-level images are processed together with the Y component image sequence.

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, as well as the blur in the imaging process, are computed.

2.1. *Image Registration*

Keren *et al.* [12] used the following method, based on [13], which has been found to be the most accurate for our purposes. This image registration method, which we used to obtain the results reported in this paper, is reviewed here. Other image registration methods that assume different imaging models have also been developed [3, 8, 14].

Horizontal shift a, vertical shift b, and rotation angle θ between images g_1 and g_2 can be written as

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

Expanding $\sin(\theta)$ and $\cos(\theta)$ to the first two terms in their Taylor's series expansion gives

$$g_2(x, y) \approx g_1(x + a - y\theta - x\theta^2/2,\ y + b + x\theta - y\theta^2/2).$$

Expanding g_1 to the first term of its own Taylor's series expansion gives the first-order equation

$$\begin{aligned} g_2(x, y) \approx g_1(x, y) &+ (a - y\theta - x\theta^2/2)\frac{\partial g_1}{\partial x} \\ &+ (b + x\theta - y\theta^2/2)\frac{\partial g_1}{\partial y}. \end{aligned} \tag{2}$$

The error function between g_2 and g_1 after rotation by θ and translation by a and b can therefore be approximated by

$$\begin{aligned} E(a, b, \theta) = \sum \Big[& g_1(x, y) + (a - y\theta - x\theta^2/2)\frac{\partial g_1}{\partial x} \\ &+ (b + x\theta - y\theta^2/2)\frac{\partial g_1}{\partial y} - g_2(x, y)\Big]^2, \end{aligned}$$

where the summation is over the overlapping part of g_1 and g_2.

The minimum of $E(a, b, \theta)$ can be recovered by computing its derivatives with respect to a, b, and θ and setting them to zero. Solving the following equation for (a, b, θ) will thus minimize the difference between the image g_2 and the image g_1 warped by (a, b, θ):

$$\begin{aligned} \sum g_x^2 a + \sum g_x g_y b + \sum A g_x \theta &= g_x g_t, \\ \sum g_x g_y a + \sum g_y^2 b + \sum A g_y \theta &= g_y g_t, \\ \sum A g_x a + \sum A g_y b + \sum A^2 \theta &= A g_t, \end{aligned} \tag{3}$$

where $g_x = \partial g_1/\partial x$, $g_y = \partial g_1/\partial y$, $g_t = g_2 - g_1$, and $A = xg_y - yg_x$. 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 Eqs. (3). We therefore iterate the following process for two given images g_1 and g_2 [12]:

1. Initially assume no motion between the frames.
2. Compute approximations to the motion parameters by solving Eqs. (3). Add the computed motion to the existing motion estimate.
3. Warp frame g_2 toward 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, 9 of the 12 coefficients in the set of equations are computed only once, and only 3 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 [17]. 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 scanned by a flatbed scanner, and its blurring function was evaluated by scanning a small, white dot on a black background. We have approximated the point spread function by a 3×3 kernel.

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.

5. SUPERRESOLUTION

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

The presented algorithm for solving the superresolution 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 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

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

The algorithm is described schematically in Fig. 1.

DEFINITION 3.1. A low-resolution pixel **y** is *influenced* by a high-resolution pixel **x**, if **x** is in **y**'s receptive field.

DEFINITION 3.2. A low-resolution image g is *influenced* by a high-resolution pixel **x**, if g contains a pixel **y** such that **y** is influenced by **x**.

The following notation is used:

- f, the target high-resolution image to be constructed (unknown);

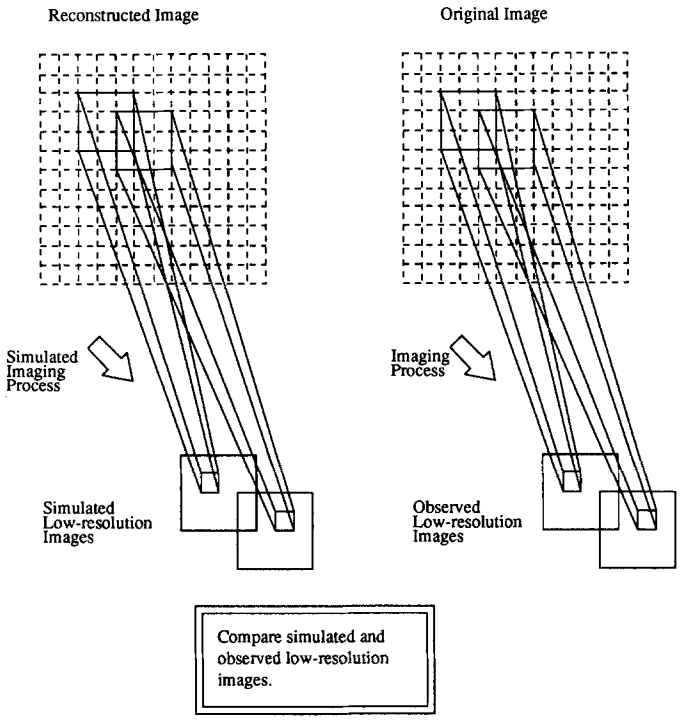

FIG. 1. Schematic diagram of the superresolution algorithm. A high-resolution reconstructed image (left) is sought, which gives simulated low-resolution images that are as close as possible to the observed low-resolution images.

- $f^{(n)}$, the approximation of f obtained after n iterations;
- g_k, the kth 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);
- $\mathbf{x}$ denotes a high-resolution pixel;
- $\mathbf{y}$ denotes a low-resolution pixel (influenced by $\mathbf{x}$).

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

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

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

$$f^{(n+1)}(\mathbf{x}) = f^{(n)}(\mathbf{x}) + \sum_{\mathbf{y} \in \cup_k Y_{k,\mathbf{x}}} (g_k(\mathbf{y}) - g_k^{(n)}(\mathbf{y})) \frac{(h_{\mathbf{xy}}^{\mathrm{BP}})^2}{c \sum_{\mathbf{y}' \in \cup_k Y_{k,\mathbf{x}}} h_{\mathbf{xy}'}^{\mathrm{BP}}}, \quad (4)$$

where

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

In Eq. (4) the value of $f^{(n)}$ at each high-resolution pixel $\mathbf{x}$ is updated according to all low-resolution pixels $\mathbf{y}$ which it influences. The contribution of the low-resolution pixel $\mathbf{y}$ of an input image g_k is the error $(g_k(\mathbf{y}) - g_k^{(n)}(\mathbf{y}))$ multiplied by a factor of $h_{\mathbf{xy}}^{\mathrm{BP}}/c$. Therefore, strongly influenced low-resolution pixels also strongly influence $f^{(n+1)}(\mathbf{x})$, while weakly influenced low-resolution pixels hardly influence $f^{(n+1)}(\mathbf{x})$. Since receptive fields of different low-resolution pixels overlap, $f^{(n+1)}(\mathbf{x})$'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 value 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. 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 another desired property. Additional considerations for choosing h^{BP} appear in Section 4.

This superresolution algorithm performs well on both 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 five iterations) and is very stable. The complexity of the algorithm is low; there are $O(KN \min\{M, \log N\})$ 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. 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.

3.1. *Color Superresolution*

The section describes the application of the superresolution algorithm to color image sequences.

The color images are first transformed into YIQ representation [15, 2], since in this representation most of the energy is concentrated in the luminance (Y) component,

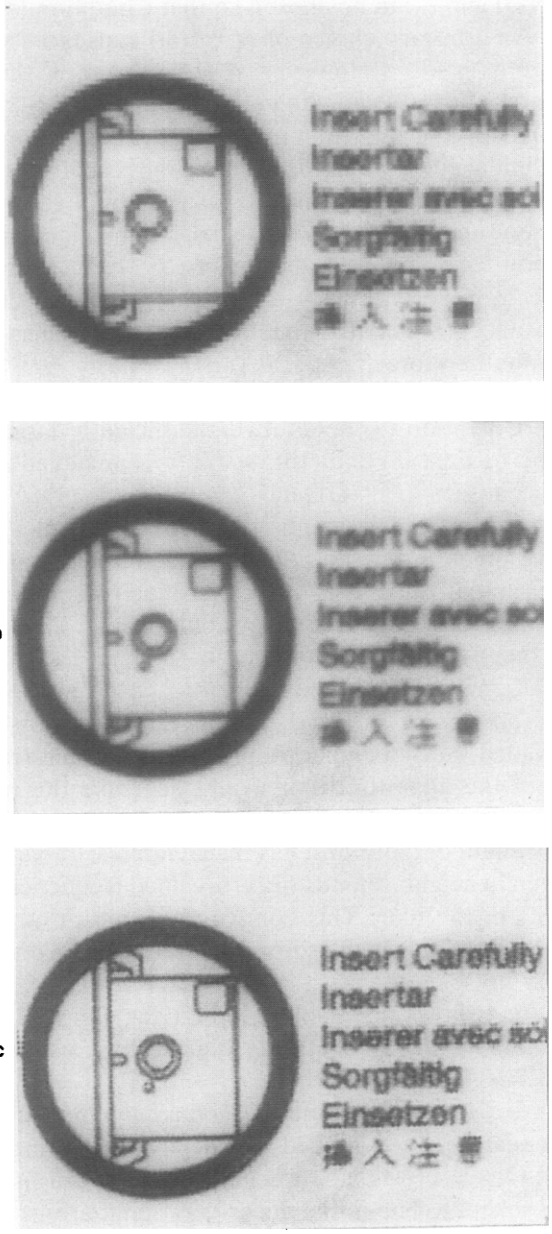

FIG. 2. Superresolution from three 100 × 70 monochrome input images. (a) One of the three original images. (b) Initial guess: average of the three input images after registration. (c) Improved resolution image.

with little energy in the chrominance (I,Q) components [4]. It is therefore sufficient to apply the iterative monochrome superresolution algorithm only to the Y component, and use simpler processing of the I and Q components. In our experiments it was sufficient to average each of the chrominance component sequences after registration. Applying superresolution to the Y component enables the mixing of color and gray-level images, as the gray-level images could be mixed with the Y component images.

The superresolution algorithm for color images is therefore the following:

Step 1. Transform the color images into YIQ representation.

Step 2. Apply the monochrome superresolution algorithm to the Y component images.

Step 3. Register the images at the two chrominance image sequences using the same registration parameters obtained in Step 2 for the Y component. Create an average image for each of the I and Q components.

Step 4. Use the high-resolution Y component and the low-resolution I and Q components to generate a high-resolution RGB image.

Figure 3 shows the result of applying this algorithm to three low-resolution RGB images scanned by a flat-bed scanner. The sampling rate was doubled in both directions. Since it is hard to reproduce color images, only the green image is displayed.

4. DEBLURRING

Restoration of degraded images, when a model of the degradation process is given, is considered as an ill-conditioned problem [1, 5, 9, 11, 18]. In this section deblurring of a single image is shown to be a special case of superresolution, 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 (4) then reduces to

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

Using convolutions, this can be written as

$$f^{(n+1)} = f^{(n)} + (g - g^{(n)}) * \frac{h^{\mathrm{AUX}}}{c}, \tag{5}$$

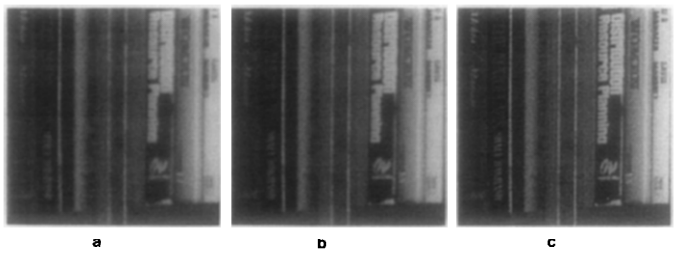

FIG. 3. Superresolution from three 100 × 100 color images. To enable printing, only the green band is shown. (a) The green band of an original low-resolution color image. (b) The initial guess: average of the three input images after registration. The average was performed in the YIQ domain. (c) The green band of the improved resolution image. Resolution has been improved only for the Y component.

where

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

When the image blur is expressed by

$$g = f * h^{PSF}, \tag{6}$$

the following theorems show that the iterative superresolution scheme is an effective deblurring operator.

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

$$\forall \omega \quad 0 < \left| 1 - \frac{H^{PSF}(\omega)H^{AUX}(\omega)}{c} \right| < 1, \tag{7}$$

then the iterations of Eq. (5) converge to the original image f.

Proof. See Appendix.

It is clear that Condition (7) does not hold for frequencies ω for which $H^{PSF}(\omega) = 0$. Those frequencies are completely lost and cannot be restored by any method. For any other ω, however, h^{AUX} and c may be chosen so that Condition (7) is fulfilled (recall that we are dealing with a finite space), and the original frequency is fully restored. The behavior of frequencies ω for which $H^{PSF}(\omega) = 0$ will be examined later.

THEOREM 4.2. *Given Condition (7), 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. See Appendix.

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 (7) holds for as many frequencies as possible, ensuring optimal conditions for convergence. A sensible choice of h^{AUX} and c increases numerical stability. The farther the term $|1 - H^{PSF}(\omega)H^{AUX}(\omega)/c|$ is from its lower limit (0), the less sensitive the algorithm is to noise and errors. However, since the speed of convergence is acclerated as the term approaches 0, there is a trade-off between stability and speed of convergence.

It should be emphasized that even if Condition (7) in Theorem 4.1 does not hold for all frequencies ω, the proof still holds for the frequencies that fulfill this condition.

The following is an examination of the behavior of the algorithm in case Condition (7) does not hold. This happens only when $H^{PSF}(\omega) = 0$, as in any other case $H^{AUX}(\omega)$ and c can be chosen so that Condition (7) does hold. An arbitrary choice of $H^{AUX}(\omega)$ causes $F^{(n)}(\omega)$ to diverge as $n \to \infty$. However, if $H^{AUX}(\omega) = 0$, then according to Eqs. (8) and (10) in the Appendix $\forall n\ F^{(n)}(\omega) = F^{(0)}(\omega)$; i.e., the algorithm preserves the component of ω in the initial guess $f^{(0)}$. This is one of the reasons for using the average of the input images as the initial guess.

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, $\forall \omega \quad H^{PSF}(\omega)H^{AUX}(\omega) = (H^{PSF}(\omega))^2 \geq 0$. This means that for all nonzero frequency components, Condition (7) holds if c is sufficiently large. According to Eqs. (8) and (10), stability is achieved in this case because as $H^{PSF}(\omega)$ tends to zero, so does $H^{AUX}(\omega)$. This prevents components of such frequencies from varying much in 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 can be observed by adding a noise term to G in Eq. (10). In general, $H^{AUX}(\omega)$ may be chosen to be proportional to higher powers of $H^{PSF}(\omega)$, as long as Condition (7) is maintained. Such a choice would increase numerical stability of the algorithm, but would decrease the rate of convergence.

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

Figure 4 shows the result of deblurring a monochrome image which was synthetically blurred by convolving it with a 7×7 blurring kernel.

Figure 5 shows the result of deblurring a monochrome 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.

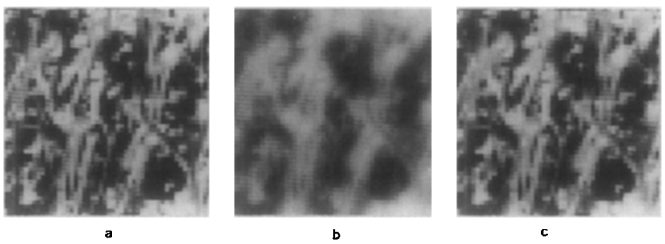

FIG. 4. Deblurring a 50×50 synthetically blurred monochrome image. (a) Original image. (b) Blurred image. (c) Restored image.

a b c

FIG. 5. Deblurring a 100 × 70 monochrome input image, with an arbitrary initial guess. (a) Blurred input image. (b) An arbitrary initial guess. (c) Deblurred image, which is independent of the initial guess.

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 iterations in a row. 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 for a noisy pixel to be fixed as its gray level usually differs from those of its neighboring pixels.

5.2. *Noise Reduction*

Referring back to the updating scheme of Eq. (4), the new value of a high-resolution pixel in each iteration is 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.

a b c

FIG. 6. Superresolution from ten 50 × 70 noisy monochrome images. (a) One of the 10 low-resolution images. (b) The initial guess: average of the low-resolution images after registration. (c) Improved resolution image.

Figure 6 shows the result of applying the algorithm to 10 monochrome low-resolution images, contaminated by zero-mean Gaussian noise of standard deviation 10. The sampling rate was doubled in each direction.

6. CONCLUDING REMARKS

Superresolution is shown to be feasible for monochrome and color image sequences, when the relative displacements can be computed, and with approximate knowledge of the imaging process.

An iterative algorithm for computing superresolution has been presented. It was shown that when the algorithm is applied to a single image without increasing the sampling rate, superresolution 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 when 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 and multiple motions in the image. 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, as well as blur that varies between sample locations, can be accommodated.

APPENDIX

The appendix contains proofs of Theorems 4.1 and 4.2. The following notation will be used:

- H^{PSF} and H^{AUX} denote the Fourier transforms of h^{PSF} and h^{AUX}, respectively.
- h denotes the expression

$$h = \frac{h^{\mathrm{PSF}} * h^{\mathrm{AUX}}}{c}.$$

- $H(\omega)$ denotes the expression

$$H(\omega) = \frac{H^{\mathrm{PSF}}(\omega) H^{\mathrm{AUX}}(\omega)}{c}. \tag{8}$$

THEOREM 4.1. *If*

$$\forall \omega \quad 0 < |1 - H(\omega)| < 1 \tag{9}$$

then the iterations of Eq. (5) *converge to the original image f.*

Proof. Mathematical manipulations on (5) yield

$$\begin{aligned} f^{(n+1)} &= f^{(n)} + (g - g^{(n)}) * \frac{h^{\mathrm{AUX}}}{c} \\ &= f^{(n)} + (g - f^{(n)} * h^{\mathrm{PSF}}) * \frac{h^{\mathrm{AUX}}}{c} \quad \text{(using (6))} \\ &= f^{(n)} * (\delta - h) + g * \frac{h^{\mathrm{AUX}}}{c} \\ &\cdot \\ &\cdot \quad \text{(unfolding the recursion)} \\ &\cdot \\ &= f^{(0)} * (\delta - h)^{*(n+1)} + g * \frac{h^{\mathrm{AUX}}}{c} * \sum_{j=0}^{n} (\delta - h)^{*j}, \end{aligned}$$

where

- δ denotes the unity pulse function centered at 0,
- a^{*n} denotes

$$\underbrace{a * a * \cdots * a}_{n \text{ times}} \quad (a^{*0} = \delta).$$

It is easier to analyze the behavior of the algorithm in the frequency domain (where convolutions are replaced by multiplications). Let F, $F^{(n)}$, and G denote the Fourier transforms of f, $f^{(n)}$, and g, respectively. Then switching to the Fourier domain we get

$$\begin{aligned} F^{(n+1)}(\omega) &= F^{(0)}(\omega)(1 - H(\omega))^{n+1} \\ &\quad + G(\omega) \frac{H^{\mathrm{AUX}}(\omega)}{c} \sum_{j=0}^{n} (1 - H(\omega))^{j}. \end{aligned} \tag{10}$$

Since (according to Condition (9)) $H^{\mathrm{PSE}}(\omega) \neq 0$ and $H^{\mathrm{AUX}}(\omega) \neq 0$,

$$\begin{aligned} F^{(n+1)}(\omega) &= F^{(0)}(\omega)(1 - H(\omega))^{n+1} \\ &\quad + G(\omega) \frac{H^{\mathrm{AUX}}(\omega)}{c} \frac{(1 - (1 - H(\omega))^{n+1})}{H(\omega)} \\ &= F^{(0)}(\omega)(1 - H(\omega))^{n+1} \\ &\quad + \frac{G(\omega)}{H^{\mathrm{PSF}}(\omega)} (1 - (1 - H(\omega))^{n+1}). \end{aligned}$$

As $|1 - H(\omega)| < 1$ (9),

$$\lim_{n \to \infty} (1 - H(\omega))^{n} = 0. \tag{11}$$

We therefore have

$$\lim_{n \to \infty} F^{(n+1)}(\omega) = \frac{G(\omega)}{H^{\mathrm{PSF}}(\omega)}. \tag{12}$$

Since by definition (6)

$$G(\omega) = F(\omega) H^{\mathrm{PSFPS1A}}(\omega), \tag{13}$$

from (12) and (13) we get

$$\lim_{n \to \infty} F^{(n+1)}(\omega) = F(\omega), \tag{14}$$

For (14) to entail $\lim_{n \to \infty} f^{(n)} = f$, it is sufficient that the convergence of $F^{(n)}$ to F be uniform. Since our images are discrete and of finite number of samples, this condition holds. (In the continuous case, Condition (9) must be slightly altered to $\varepsilon < |1 - H(\omega)| < 1 - \varepsilon$, for some $\varepsilon > 0$.) ■

THEOREM 4.2. *Given Condition* (9), *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 (11) confirms the exponential speed of convergence. The absence of $F^{(0)}(w)$ in Eq. (12) confirms that the choice of initial guess does not affect the convergence. ■

REFERENCES

1. H. C. Andrews and B. R. Hunt (Eds.), *Digital Image Restoration*, Prentice–Hall, Englewood Cliffs, NJ, 1977.
2. D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice–Hall, Englewood Cliffs, NJ, 1982.
3. E. De Castro and C. Morandi, Registration of translated and rotated images using finite Fourier transforms, *IEEE Trans. Pattern Anal. Mach. Intelligence* **9**, No. 5, 1987, 700–703.
4. H. Gharavi and A. Tabatabai, Application of quadrature mirror filtering to the coding of monochrome and color images, in *International IEEE Conference on Acoustic, Speech, and Signal Processing*, pp. 2384–2487, 1987.
5. R. C. Gonzalez, Image enhancement and restoration, in *Handbook of Pattern Recognition and Image Processing* (T. Y. Young and K. S. Fu, Ed.), pp. 191–213, Academic Press, San Diego, 1986.
6. D. Gross, Super resolution from sub-pixel shifted pictures, Master's thesis, Tel-Aviv University, Oct. 1986.
7. D. A. Hayner and W. K. Jenkins, The missing cone problem in computer tomography, in *Advances in Computer Vision and Image Processing* (T. S. Huang, Ed.), Vol. 1, pp. 83–114, JAI Press, Greenwich, CT, 1984.
8. B. K. P. Horn and E. J. Weldon, Direct methods for recovering motion, *Int. J. Comput. Vision* **2**, No. 1, 1988, 51–76.

9. T. S. Huang (Ed.), *Image Enhancement and Restoration*, JAI Press, New York, 1986.
10. T. S. Huang and R. Y. Tsai, Multi-frame image restoration and registration, in *Advances in Computer Vision and Image Processing* (T. S. Huang, Ed.), Vol. 1, pp. 317–339, JAI Press, Greenwich, CT, 1984.
11. R. A. Hummel, B. Kimia, and S. W. Zucker, Deblurring gaussian blur, *Comput. Vision, Graphics Image Process.* **38,** 1987, 66–80.
12. D. Keren, S. Peleg, and R. Brada, Image sequence enhancement using sub-pixel displacements, in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 742–746, Ann Arbor, MI, June 1988.
13. B. D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, in *International Joint Conference on Artificial Intelligence*, pp. 674–679, Vancouver, Aug. 1981.
14. S. Negahdaripour and B. K. P. Horn, Direct passive navigation, *IEEE Trans. Pattern Anal. Mach. Intelligence* **9,** No. 1, 1987, 168–176.
15. A. N. Netravlia and B. G. Haskell, *Digital Pictures: Representation and Compression*, Plenum, New York, 1988.
16. S. Peleg, D. Keren, and L. Schweitzer, Improving image resolution using subpixel motion, *Pattern Recognit. Lett.*, 1987, 223–226.
17. A. Rosenfeld (Ed.), *Multiresolution Image Processing and Analysis*, Springer-Verlag, Berlin/New York, 1984.
18. H. Shvayster and S. Peleg, Inversion of picture operators, *Pattern Recognit. Lett.* **5,** 1985, 49–61.