Blind deconvolution by means of the Richardson–Lucy algorithm

D. A. Fish, A. M. Brinicombe, and E. R. Pike

Department of Physics, King's College London, Strand, London WC2R 2LS, UK

J. G. Walker

Department of Electrical and Electronic Engineering, University Park, University of Nottingham, Nottingham NG7 2RD, UK

Received April 7, 1994; revised manuscript received August 8, 1994; accepted August 22, 1994

A blind deconvolution algorithm based on the Richardson–Lucy deconvolution algorithm is presented. Its performance in the presence of noise is found to be superior to that of other blind deconvolution algorithms. Results are presented and compared with results obtained from implementation of a Weiner filter blind deconvolution algorithm. The algorithm is developed further to incorporate functional forms of the point-spread function with unknown parameters. In the presence of noise the point-spread function can be evaluated with 1.0% error, and the object can be reconstructed with a quality near that of the deconvolution process with a known point-spread function.

1. INTRODUCTION

Blind deconvolution is the term given to an image-restoration technique in which complete knowledge of both the point-spread function (PSF) and the object are not available. Ayers and Dainty[1] proposed a scheme that essentially generalized the Feinup phase retrieval algorithm.[2] The technique is iterative, and *a priori* knowledge is limited to the nonnegativity of images. In each iteration one obtains estimates of the object and the PSF by simple inverse filtering. Davey *et al.*[3] proposed a similar scheme, but their algorithm assumed further *a priori* knowledge, i.e., that the object support was known. In their study a Weiner-type filter was used to obtain estimates of the object and the PSF. This method thus permitted better noise compensation.

In this paper a Weiner filter blind deconvolution algorithm is implemented and is compared with a new algorithm based on the Richardson–Lucy[4,5] deconvolution. The Richardson–Lucy algorithm has proved to be robust in the presence of noise; therefore we thought that a blind deconvolution algorithm based on this technique might have advantages over the Ayers–Dainty and the Davey–Lane–Bates algorithms. The results shown here confirm the high noise tolerance of our new algorithm.

To improve further the performance of this type of algorithm, we incorporated extra *a priori* knowledge by assuming a functional form for the PSF. It was thought that this method would produce better results because the number of unknowns is reduced from thousands of pixel values to a small number of parameters that describe the PSF. It is likely that blind deconvolution performed in this manner would find use in many areas in which it is not possible to know exactly how an optical system is aberrated but which could be characterized by a few free parameters. One example of such an application is in telescopes in space, where unknown fluctuations of mirrors, which are due to time-varying gravitational fields, do not permit exact knowledge of the PSF.

2. BLIND DECONVOLUTION BY THE RICHARDSON–LUCY ALGORITHM

The Richardson–Lucy deconvolution algorithm has become popular in the fields of astronomy and medical imaging. Initially it was derived from Bayes's theorem in the early 1970's by Richardson and Lucy.[4,5] In the early 1980's it was rederived by Shepp and Vardi[6] as an algorithm to solve positron emission tomography imaging problems, in which Poissonian statistics are dominant. Their method used a maximum-likelihood solution, which was found by use of the expectation maximization algorithm of Dempster *et al.*[7] The reason for the popularity of the Richardson–Lucy algorithm is its implementation of maximum likelihood and its apparent ability to produce reconstructed images of good quality in the presence of high noise levels. We therefore assumed that a blind form of this algorithm would have the same characteristics. A blind deconvolution algorithm similar to the one shown here was also developed by Holmes[8] by use of the expectation maximization algorithm of Dempster *et al.*[7]

We begin with a brief review of the Richardson–Lucy deconvolution method and then present the blind form of the algorithm. The Richardson–Lucy algorithm was developed from Bayes's theorem. Because it relates conditional probabilities the algorithm takes into account statistical fluctuations in the signal and therefore has the ability to reconstruct noisy images. Bayes's theorem is given by

$$P(x\,|\,y) = \frac{P(y\,|\,x)P(x)}{\int P(y\,|\,x)P(x)\mathrm{d}x}, \tag{1}$$

where $P(y\,|\,x)$ is the conditional probability of an event y, given event x. $P(x)$ is the probability of an event x, and $P(x\,|\,y)$ is the inverse conditional probability, i.e., the probability of event x, given event y. The probability $P(x)$ can be identified as the object distribution $f(x)$; the

0740-3232/95/010058-08$06.00

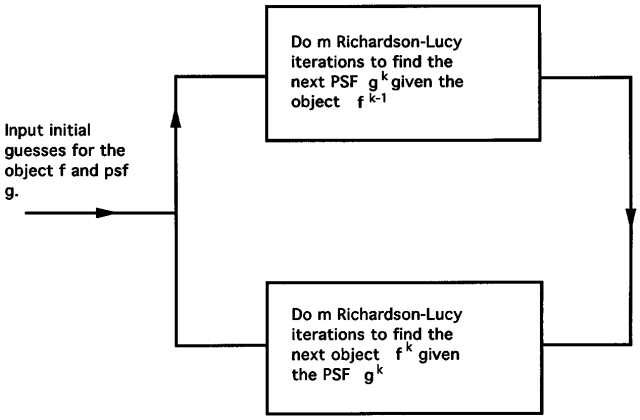


Fig. 1. Blind deconvolution based on the Richardson–Lucy algorithm.

conditional probability $P(y|x)$ can be identified as the PSF centered at x, i.e., $g(y, x)$; and the probability $P(y)$ can be identified as the degraded image or convolution $c(y)$. This inverse relation permits the derivation of the iterative algorithm

$$f_{i+1}(x) = \int \frac{g(y, x)c(y)\mathrm{d}y}{\int g(y, z)f_i(z)\mathrm{d}z} f_i(x), \tag{2}$$

where i is the iteration number. If an isoplanatic condition exists, then Eq. (2) can be written in terms of convolutions:

$$f_{i+1}(x) = \left\{\left[\frac{c(x)}{f_i(x) \otimes g(x)}\right] \otimes g(-x)\right\} f_i(x), \tag{3}$$

where $\otimes$ is the convolution operation. The PSF $g(x)$ is known, so one finds the object $f(x)$ by iterating Eq. (3) until convergence. An initial guess is required for the object $f_0(x)$ to start the algorithm. Then, in subsequent iterations, because of the form of the algorithm, large deviations in the guess from the true object are lost rapidly in initial iterations, whereas detail is added more slowly in subsequent iterations. Advantages of this algorithm include a nonnegativity constraint if the initial guess $f_0(x) \geq 0$. Also, energy is conserved as the iteration proceeds, which is easily seen by integration of both sides of Eq. (2) over x.

In the blind form of this algorithm two of these deconvolution steps are required. At the kth blind iteration it is assumed that the object is known from the $k-1$ iteration. The PSF $g^k(x)$ is then calculated for a specified number of Richardson–Lucy iterations, as in Eq. (4) below, where the i index represents the Richardson–Lucy iteration. This equation is essentially an inverse of Eq. (3), inasmuch as the object and the PSF have reverse roles, and it calculates the PSF from the object. Then $f^k(x)$ is calculated for the same number of Richardson–Lucy iterations. This is done with the PSF evaluated from the full iteration of Eq. (4). In this case the iteration is performed in the normal manner of Eq. (3), as shown in Eq. (5) below. The degraded image is again given as $c(x)$ in both Eqs. (4) and (5). The loop is repeated as required. One writes

$$g_{i+1}{}^k(x) = \left\{\left[\frac{c(x)}{g_i{}^k(x) \otimes f^{k-1}(x)}\right] \otimes f^{k-1}(-x)\right\} g_i{}^k(x), \tag{4}$$

$$f_{i+1}{}^k(x) = \left\{\left[\frac{c(x)}{f_i{}^k(x) \otimes g^k(x)}\right] \otimes g^k(-x)\right\} f_i{}^k(x). \tag{5}$$

The above equations are shown in one dimension; the extension for two-dimensional images is straightforward. Initial guesses are made for the object $f_0{}^0(x)$ and the PSF $g_0{}^0(x)$, and an algorithm loop of the form shown in Fig. 1 is performed. No positivity constraints are required because the above equations always ensure positivity. The algorithm is different from the Holmes[8] algorithm, as only two Richardson–Lucy iterations are performed within one blind iteration, one for an object evaluation and one for the PSF evaluation. It was found that the simulated images used did not perform well with this type of iteration but that when the number of Richardson–Lucy iterations within one blind iteration was increased to approximately ten a much better performance was obtained.

To test this algorithm against another blind deconvolution algorithm for comparison of performance purposes the Davey *et al.*[3] blind deconvolution algorithm with a Weiner filter was implemented. In the implementation used here the support constraint used by Davey *et al.* was not used because no support constraint was used for the Richardson–Lucy blind deconvolution algorithm.

A convolution was created from a Gaussian to model the PSF and a cross (the object); these can be seen in Fig. 2. All the images are 64×64 pixels. Photon noise was added to the image by generation of a random number lying on a Poisson distribution with the mean of the pixel value of the noiseless image, and the numbers that were generated for all the pixels then formed the noisy image. It was found that, with this type of image and approximately 1.5% noise (where the percentage value is the

(a) (b)

(c)

Fig. 2. (a) Simulated object, (b) Gaussian PSF, and (c) their convolution with 1.5% Poissonian noise.

Fig. 3. Blind deconvolution by the Weiner filter algorithm. Reconstructions of the object (left) and the PSF (right) with (a) zero noise and (b) 1.5% noise.

standard deviation divided by the intensity at the brightest point in the image), good reconstructions could be obtained. Figure 3 shows reconstructions by means of the blind Weiner filter algorithm; Fig. 3(a) shows the noiseless case, in which the images shown are the best-error object (left) and the PSF (right) after 400 iterations. The term best error refers to the least error between the original convolution and the convolution of the reconstructed object and the PSF. Figure 3(a) shows the case of 1.5% noise; the reconstructions have deteriorated, but the cross is still distinguishable. Noise levels much above this figure resulted in unrecognizable reconstructions.

The blind Richardson–Lucy algorithm performed far better on the same image. In Fig. 4(a) images are shown with 1.5% and 10.0% noise (left and right, respectively). Figures 4(b) and 4(c) show reconstructions for both these cases, respectively. It can immediately be seen that the performance of this algorithm is far superior to the previous algorithm. Good reconstructions are obtained at both noise levels. The algorithm was applied to many other images for which Gaussian PSF's were used, and it was found that as long as the blurring of the PSF was not too severe then reasonable reconstructions could generally be obtained, in some cases with noise levels as high as 15%.

3. SEMIBLIND DECONVOLUTION

As mentioned in Section 1, further *a priori* information could be incorporated by assuming knowledge of the form of the PSF. In a real situation it may be known that a telescope suffers from spherical aberration, but because of time-varying factors such as the changing gravitational field that exists around a telescope in orbit the extent of this aberration may not be known. This situation would reduce the number of unknown variables in the deconvolution from perhaps thousands of pixel values to one or two unknown constants. We have termed this approach semiblind deconvolution.

A. Weiner Semiblind Algorithm

This algorithm used the blind deconvolution with a Weiner filter as its basis. The only part of the algorithm altered was the image-plane constraints on the PSF. In the blind algorithm the constraint was just nonnegativity. This constraint was replaced by a least-squares-fitting procedure. Initially convolutions were created with Gaussians, so Gaussians of varying widths were compared with the evaluated PSF. The Gaussian giving the least error in fitting was then chosen as the PSF, and the next object guess was evaluated with this PSF.

To illustrate how well this algorithm performed in the absence of noise, Fig. 5 shows the reconstructions of the object and the PSF at every iteration. In this particular case the images are all 128×128 pixels. This algorithm converged within three iterations and produced a perfect reconstruction of the satellite object. When we tried to

Fig. 4. Blind deconvolution by the Richardson–Lucy algorithm. (a) Convolutions with 1.5% (left) and 10.0% (right) noise. (b) Reconstructions of the object (left) and the PSF (right) at the 1.5% noise level. (c) Reconstructions of the object (left) and the PSF (right) at the 10.0% noise level.

Fig. 5. Semiblind deconvolution by a Weiner filter-based algorithm. (a) True object (left), random starting guess of the object (center), and noiseless convolution (right). (b) Object (left) and PSF (right) from the first iteration. (c) Object (left) and PSF (right) from the second iteration. (d) Object (left) and PSF (right) from the third iteration.

reconstruct with noisy images, however, the algorithm always converged on the delta-function solution, i.e., a Gaussian of smallest possible width was evaluated as the PSF. Even with noise values less than 0.1% the algorithm performed poorly. We therefore tried using the Richardson–Lucy algorithm.

B. Semiblind Deconvolution by the Richardson–Lucy Algorithm

The semiblind form of the algorithm took as its basis the blind algorithm. A number of blind iterations were performed, and then a least-squares fit on the function evaluated as the PSF was found. A PSF was then created with the fitting parameters, and then another series of blind iterations was performed, with this PSF being used as the starting point. This procedure was then repeated for a specified number of iterations. Initially, simple one-variable PSF forms were chosen, i.e., Gaussians of unknown width.

In some cases the results for this algorithm showed remarkable noise tolerance. In Fig. 6 results are shown for semiblind deconvolution on a series of point sources.

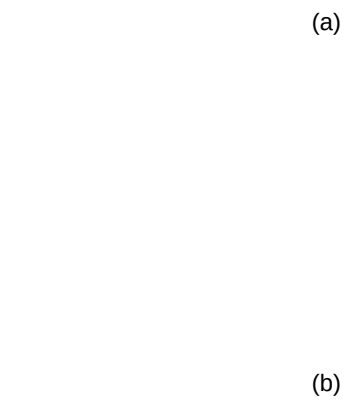


Fig. 6. Semiblind deconvolution by the Richardson–Lucy-based algorithm. (a) Object (left) and convolution (right) with 20.0% noise. (b) Reconstruction of the object (left) and the fitted Gaussian PSF (right).

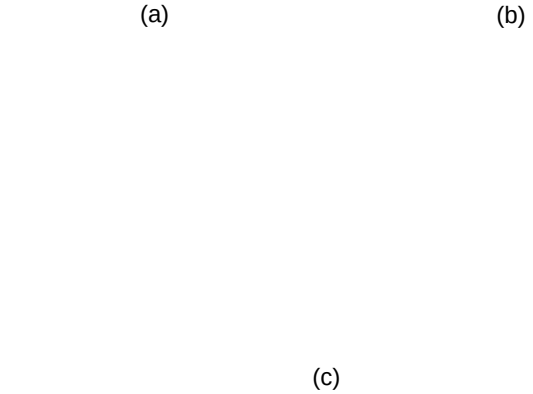


Fig. 7. Comparison of Richardson–Lucy semiblind deconvolution with standard deconvolution algorithms. (a) Reconstruction by semiblind deconvolution with a 0.1-pixel step width. (b) Reconstruction by Fourier regularization. (c) Reconstruction by the Richardson–Lucy algorithm.

The image contained approximately 20.0% noise. The reconstruction shown is good, considering the noise level.

The algorithm was also tried on the noisy image of the cross used earlier for the pure blind deconvolution research. Although the PSF was fitted in each iteration with a Gaussian of the correct size, the results were not good; in fact, the pure blind deconvolution results were better. Therefore it was decided that a Gaussian fitting process should be performed after the blind deconvolution and then a specified number of Richardson–Lucy iterations performed with the guessed Gaussian. The step width in Gaussian fitting was obviously important: with the step width of 1 pixel for the Gaussian radius at the $1/e$ height, the correct PSF width of 3 pixels was guessed.

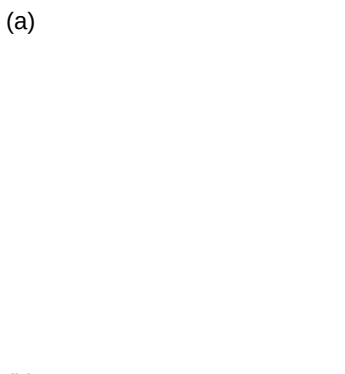


Fig. 8. Many-variable semiblind deconvolution. (a) Object (left) and PSF (right). (b) Convolution with 1.0% noise.

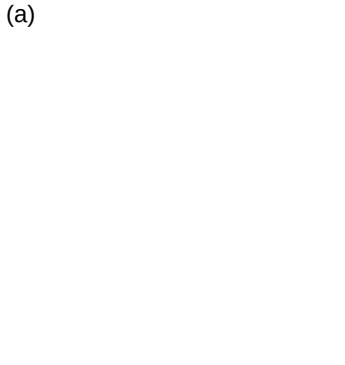
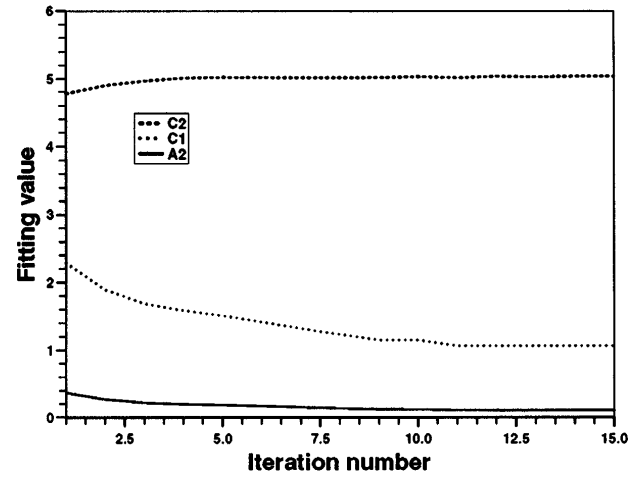
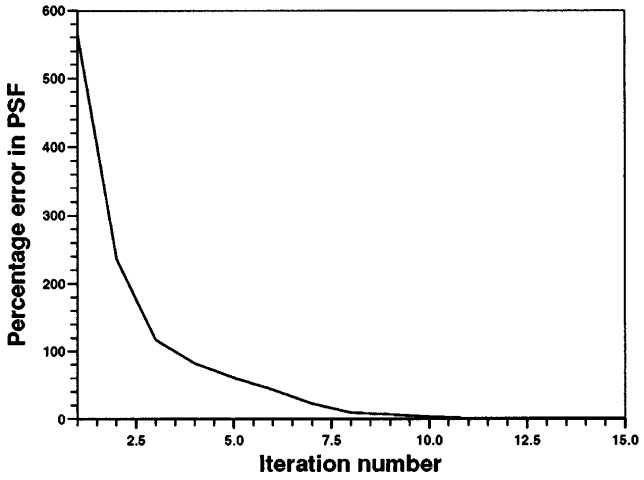


Fig. 9. Reconstructions of the image shown in Fig. 8. (a) Richardson–Lucy deconvolution after 1000 iterations. (b) Semiblind deconvolution after 15 iterations: object (left) and PSF (right).



(a)



(b)

Fig. 10. Error graphs for the 1.0% noise image shown in Fig. 8. (a) Fitting parameters A_2, C_1, C_2 with iteration number. (b) Percentage error in the PSF with iteration number.

At a step width of 0.1 pixel the Gaussian width obtained was 3.2 pixels. The results for both cases are similar and are shown in Fig. 7(a) for the 0.1-pixel case.

The results show that the slight error made in finding the width of the Gaussian PSF does not make the reconstructions significantly worse. This is probably due to the high level of noise on the image, which results in the loss of a large amount of information. To show the impressiveness of these results straightforward deconvolutions with a Weiner filter and the Richardson–Lucy algorithm are shown in Figs. 7(b) and 7(c). It can be seen that the semiblind deconvolution results are comparable with the usual methods of deconvolution.

To extend this research to cases in which it may be used in realistic situations, more than one fitting variable may be needed to describe the PSF accurately. To test this possibility, a simple PSF was created that had the functional form

$$y(r) = \sum_k \left[\frac{A_k r^2 \exp(1.0)}{{C_k}^2} + B_k \right] \exp\left(-\frac{r^2}{{C_k}^2} \right), \qquad (6)$$

where r is the radius and the variables A_k, B_k, C_k were given the values

$$A_1 = 0.0, \qquad A_2 = 0.1,$$
$$B_1 = 1.0, \qquad B_2 = 0.0,$$
$$C_1 = 1.0, \qquad C_2 = 5.0.$$

Then the variables A_2, C_1, C_2 were allowed to change their values, so that the PSF was a Gaussian plus a Gaussian times its radius squared. Incorrect values for these variables were introduced into the program, and a PSF was created. Then, as before, a blind deconvolution process evaluated a new object and a new PSF. A PSF with the functional form given above and with free variables A_2, C_1, C_2 was fitted to the evaluated PSF by a Levenberg–Marquardt[9] nonlinear least-squares-fitting routine. This routine returned new values for A_2, C_1, C_2, and the process was repeated for a specified number of iterations.

The simulated object and the PSF used to illustrate this algorithm are shown in Fig. 8(a), and a 1.0% noise convolution is shown in Fig. 8(b) (the object used was the cross shown above). To compare the results of this semiblind deconvolution algorithm a Richardson–Lucy deconvolution was performed with 1000 iterations with the known PSF. The result of this process is shown in Fig. 9(a) and can be compared with the results of 15 iterations of the semiblind deconvolution algorithm shown in Fig. 9(b). The results compare well.

In Fig. 10(a) the variation of the fitting parameters A_2, C_1, C_2 with iteration number are shown. The starting values introduced into the program were $A_2 = 0.5$, $C_1 = 3.0$, $C_2 = 7.0$, giving a 74.0% error in the PSF. It

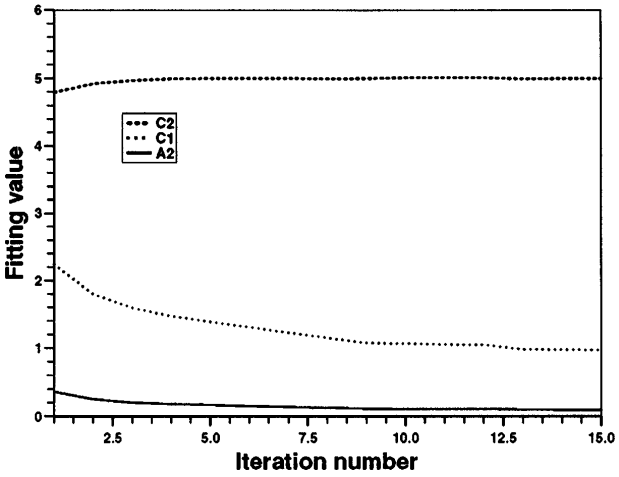
(a) (b)

(c)

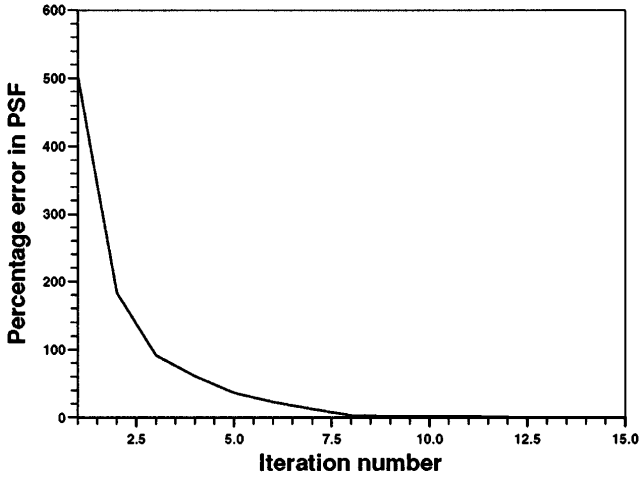
Fig. 11. Reconstructions of an image with 4.0% noise. (a) 4.0% noise image. (b) Richardson–Lucy deconvolution after 1000 iterations. (c) Semiblind deconvolution after 15 iterations: object (left) and PSF (right).

can be seen that at this noise level the algorithm is converging on the correct values. This convergence is highlighted by Fig. 10(b), which shows the percentage error in the PSF with iteration number. This result is slightly false because the true PSF will not be known in a real situation, but the figure shows the convergent properties of the algorithm, which do not occur in cases such as the Ayers–Dainty, the blind Weiner, and the blind Richardson–Lucy algorithms. In the case of the blind Richardson–Lucy algorithm, the image shown in Fig. 8(b) was used for comparison with the results of the semiblind algorithm. It was found that the algorithm did not converge to the correct values for the fitting parameters, and in fact the algorithm eventually diverged. The final values for the fitting parameters from the semiblind algorithm were $A_2 = 0.102$, $C_1 = 1.06$, $C_2 = 5.03$, with an overall error in the PSF evaluation of 1.09%.

At 2.0%, 3.0%, and 4.0% noise levels similar results were obtained, and convergence was seen at the correct values of the fitting variables. The results of the 4.0% noise image are shown in Fig. 11. This figure shows the Richardson–Lucy deconvolution after 1000 iterations and the semiblind deconvolution after 15 iterations. Again the results compare quite well. Figure 12 shows the variation in the fitting parameters and the percentage error in the PSF with iteration number. Again convergence is evident. When the same image with 6.0% noise was tried, the results were not so good. Figure 13 shows the variation of fitting parameter and the percentage error in the PSF, and it can be seen that convergence is reached after eight iterations and that the algorithm then starts to diverge. It therefore appears that the algorithm has a certain noise tolerance.

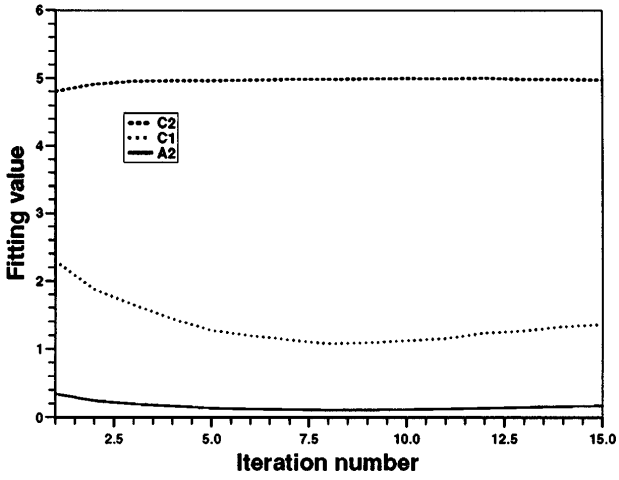


(a)

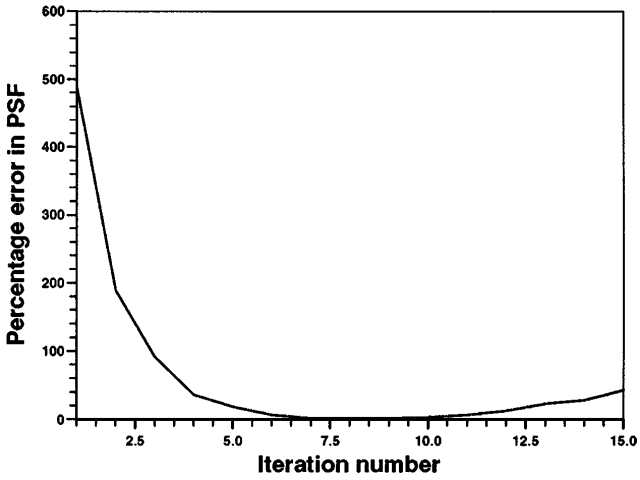


(b)

Fig. 12. Error graphs for the 4.0% noise image. (a) Fitting parameters A_2, C_1, C_2 with iteration number. (b) Percentage error in the PSF with iteration number.



(a)



(b)

Fig. 13. Error graphs for a 6.0% noise image. (a) Fitting parameters A_2, C_1, C_2 with iteration number. (b) Percentage error in the PSF with iteration number.

4. CONCLUSIONS

A blind deconvolution algorithm has been presented here that is based on the Richardson–Lucy algorithm. The algorithm presented is similar to that presented by Holmes,[8] but the implementation given here seems to have a more stable performance on the images chosen. The noise tolerance of the present algorithm is also far better than that of algorithms such as the Ayers–Dainty[1] and the Weiner filter algorithms, used for comparison purposes in this paper.

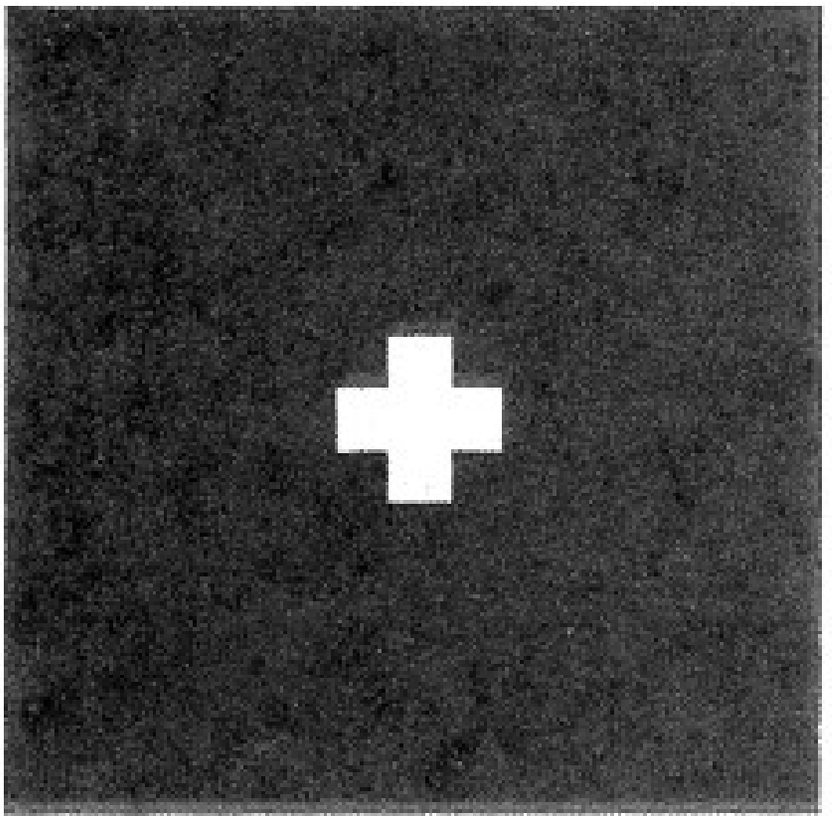
In many real situations it may be the case that some knowledge of the PSF can be obtained. Therefore functional forms for the PSF's were chosen with a number of unknown variables. It was found that accurate deconvolutions of a quality near that of a deconvolution with full knowledge of the PSF can be made. It is hoped that this research can be extended to real images with PSF's containing unknown amounts of aberration, with the algorithm evaluating both the aberration coefficients and the object.

ACKNOWLEDGMENTS

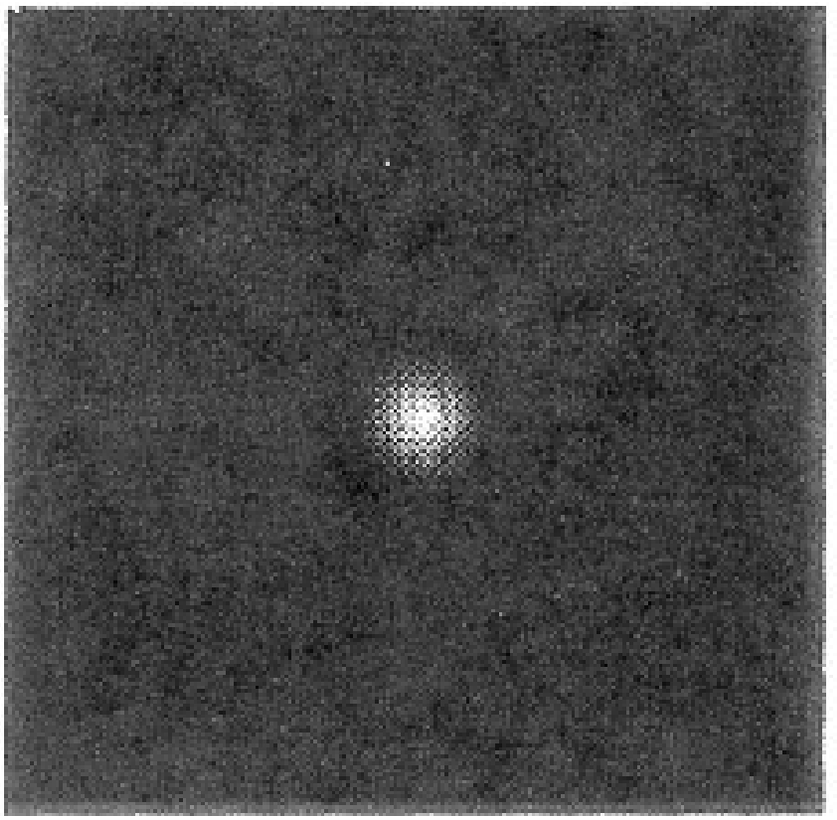
The authors are grateful to the U.S. Army Research Office for supporting this research under a project entitled "Diffraction limited imaging using aberrated optics," grant DAAL03-92-G-0142.

REFERENCES

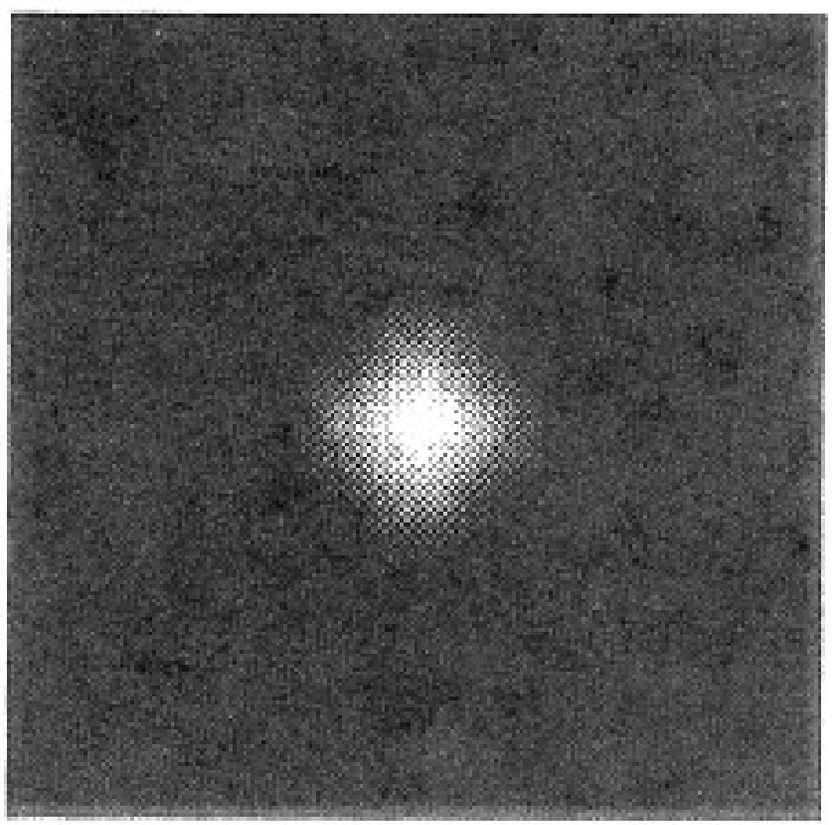
1. G. R. Ayers and J. C. Dainty, "Iterative blind deconvolution method and its applications," Opt. Lett. **13,** 547–549 (1988).
2. J. R. Feinup, "Phase retrieval algorithms: a comparison," Appl. Opt. **21,** 2758–2769 (1982).
3. B. L. K. Davey, R. G. Lane, and R. H. T. Bates, "Blind deconvolution of noisy complex-valued image," Opt. Commun. **69,** 353–356 (1989).
4. W. H. Richardson, "Bayesian-based iterative method of image restoration," J. Opt. Soc. Am. **62,** 55–59 (1972).
5. L. B. Lucy, "An iterative technique for the rectification of observed distributions," Astron. J. **79,** 745–754 (1974).
6. L. A. Shepp and Y. Vardi, "Maximum likelihood reconstructions for emission tomography," IEEE Trans. Med. Imaging **MI-1,** 113–122 (1982).
7. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," J. R. Stat. Soc. **39,** 1–38 (1977).
8. T. J. Holmes, "Blind deconvolution of quantum-limited incoherent imagery: maximum likelihood approach," J. Opt. Soc. Am. A **9,** 1052–1061 (1992).
9. W. H. Press, B. P. Flannery, J. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: the Art of Scientific Computing* (Cambridge U. Press, Cambridge, 1988).



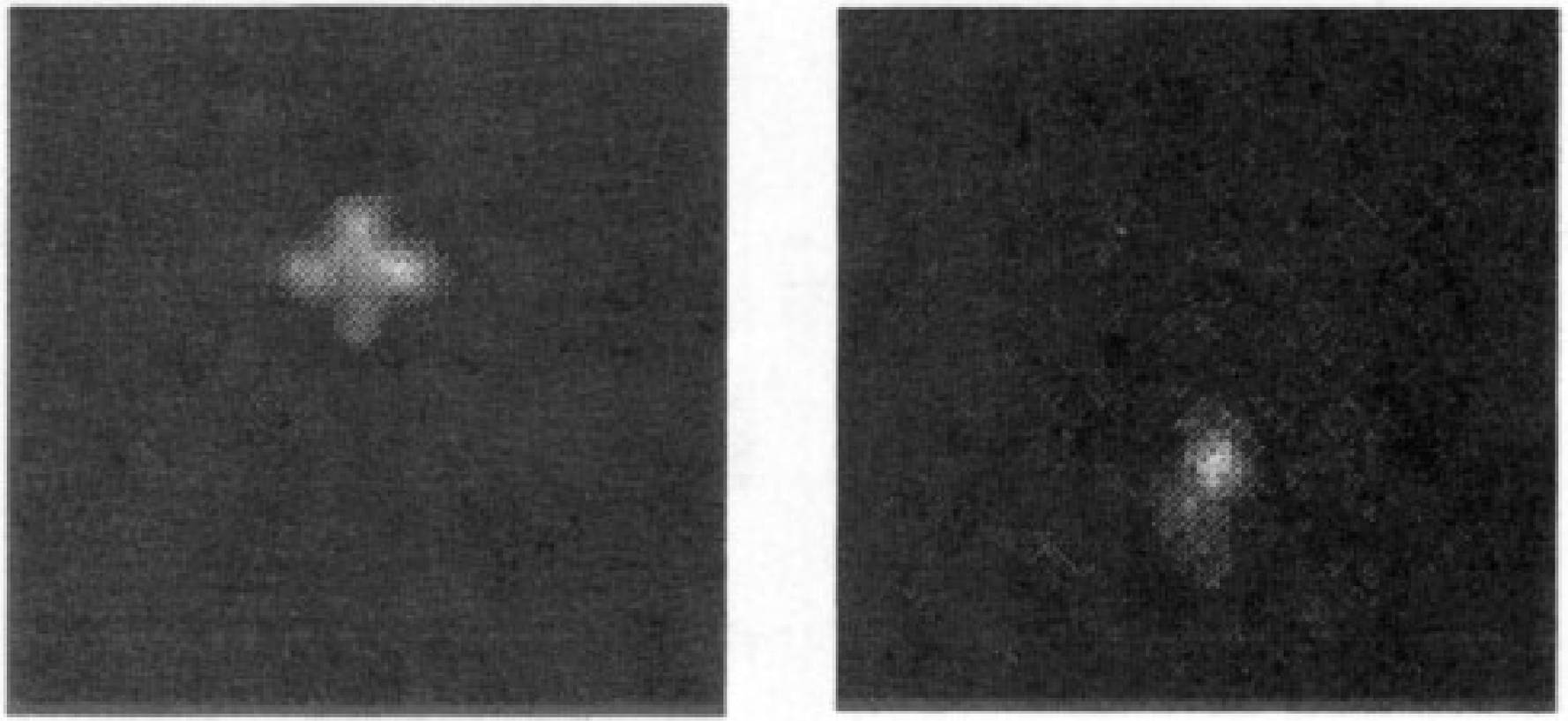
(a)



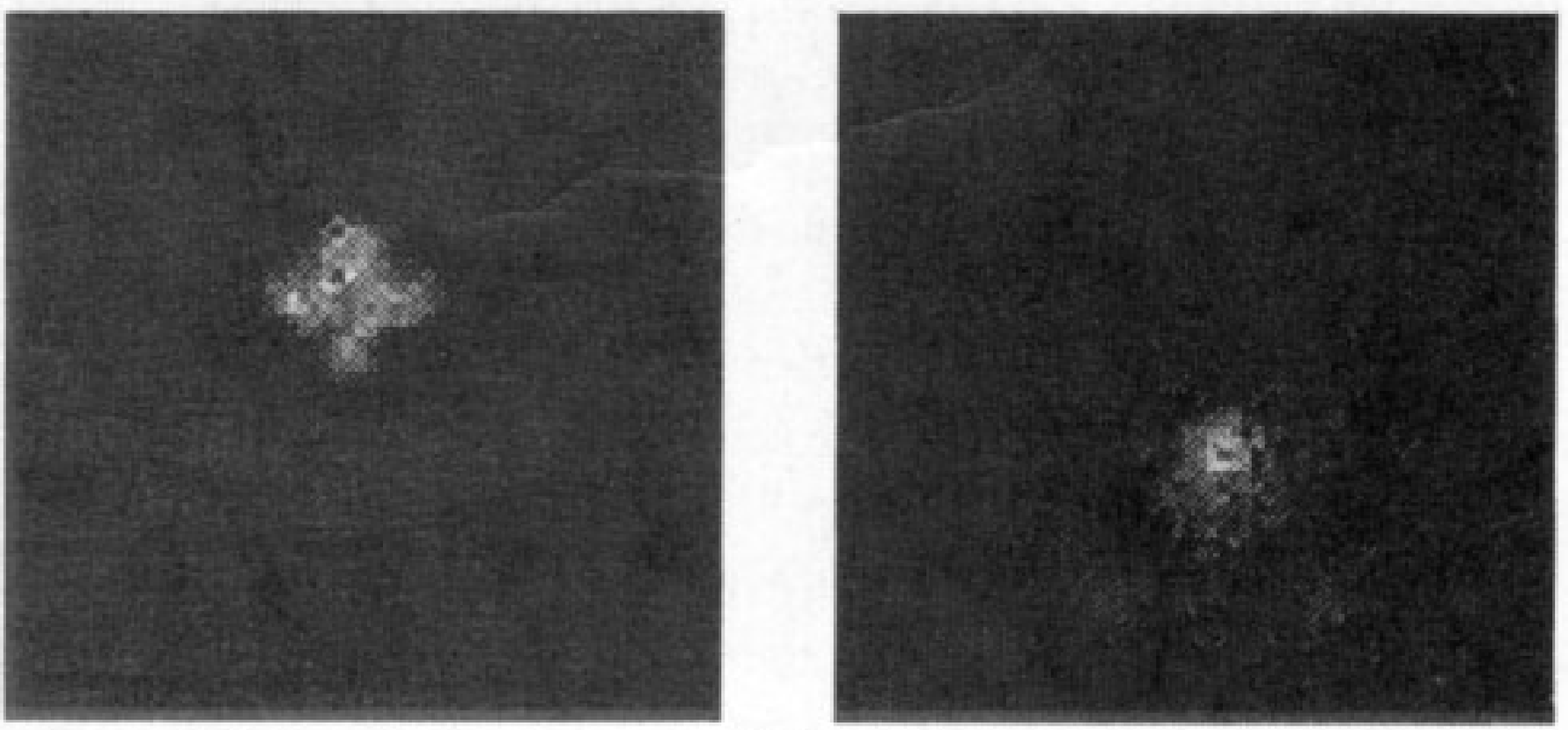
(b)



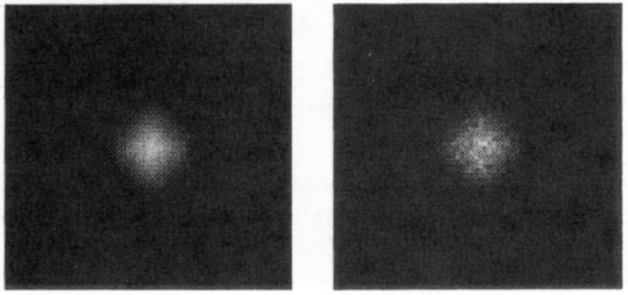
(c)



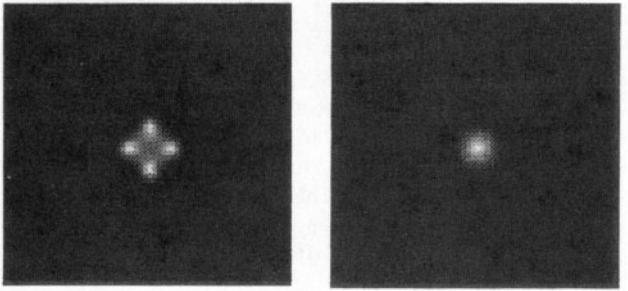
(a)



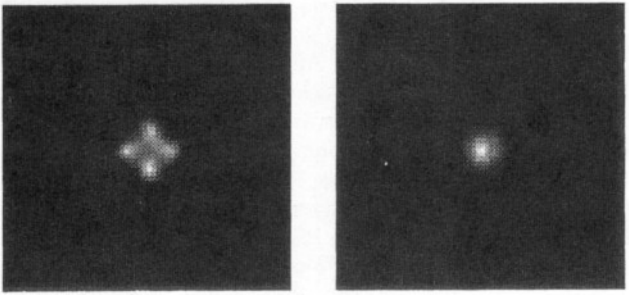
(b)



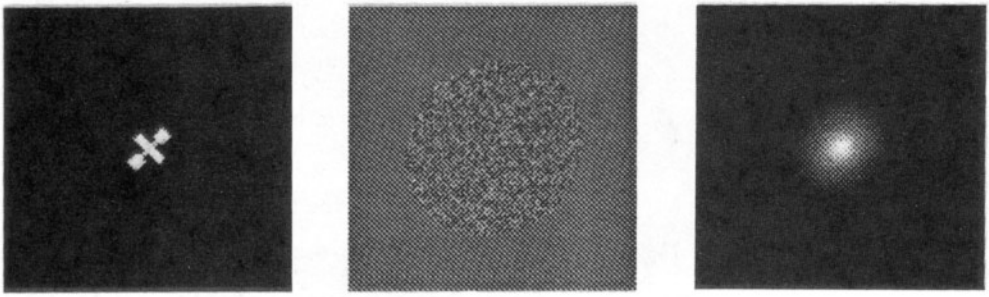
(a)



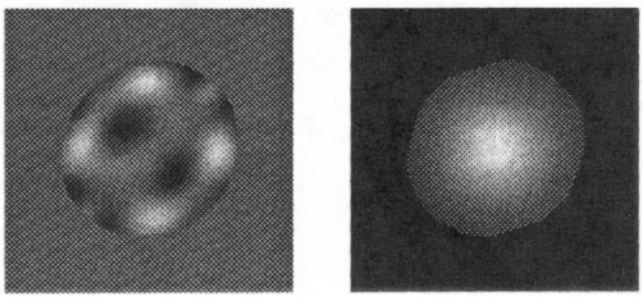
(b)



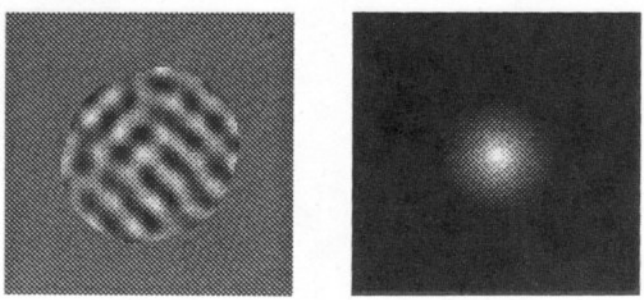
(c)



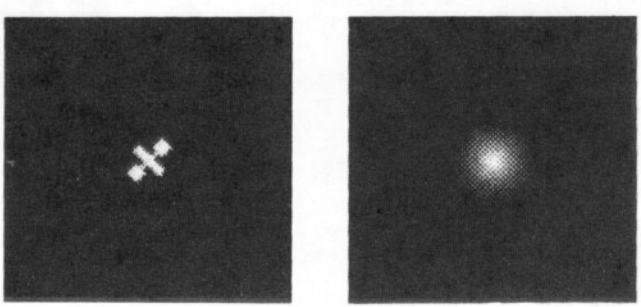
(a)



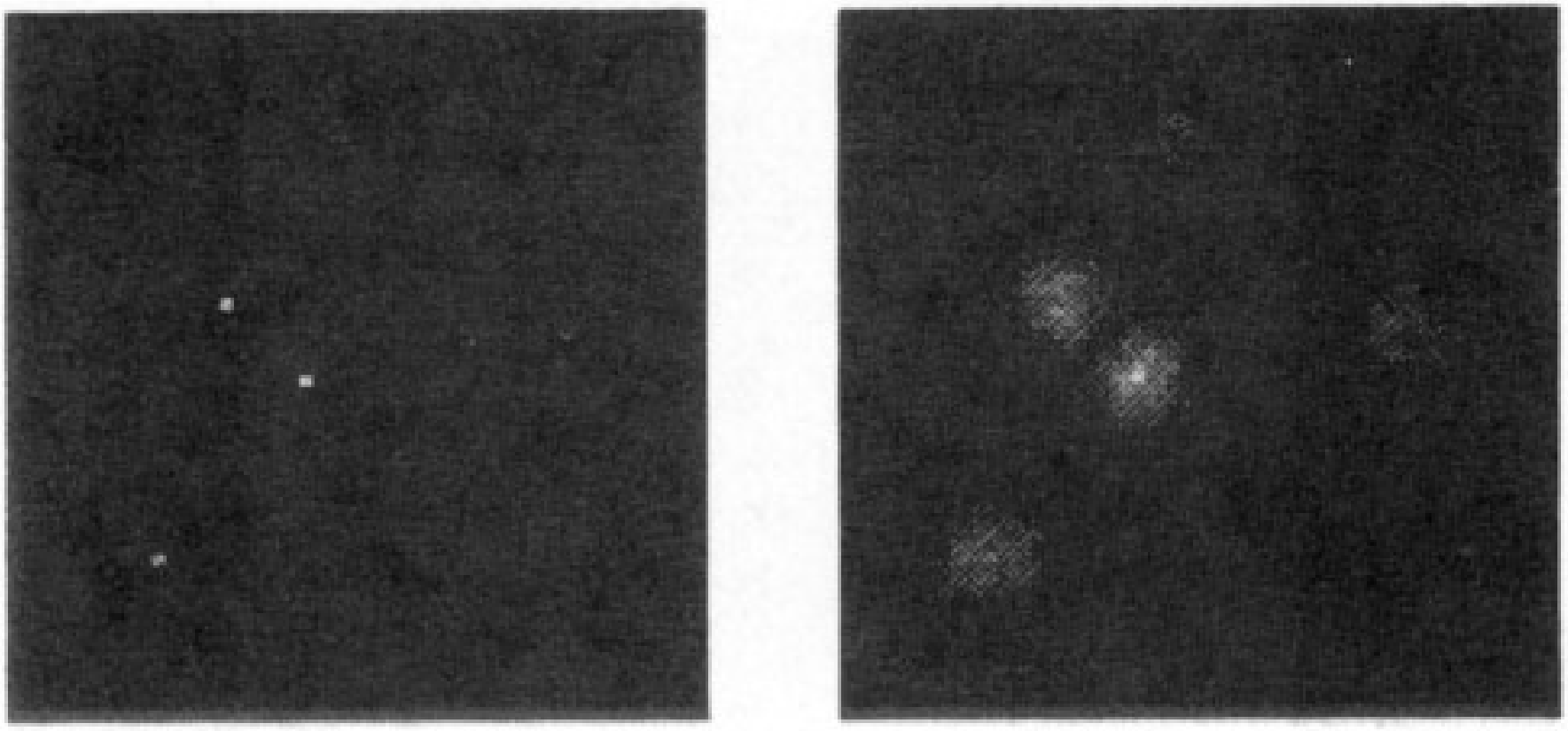
(b)



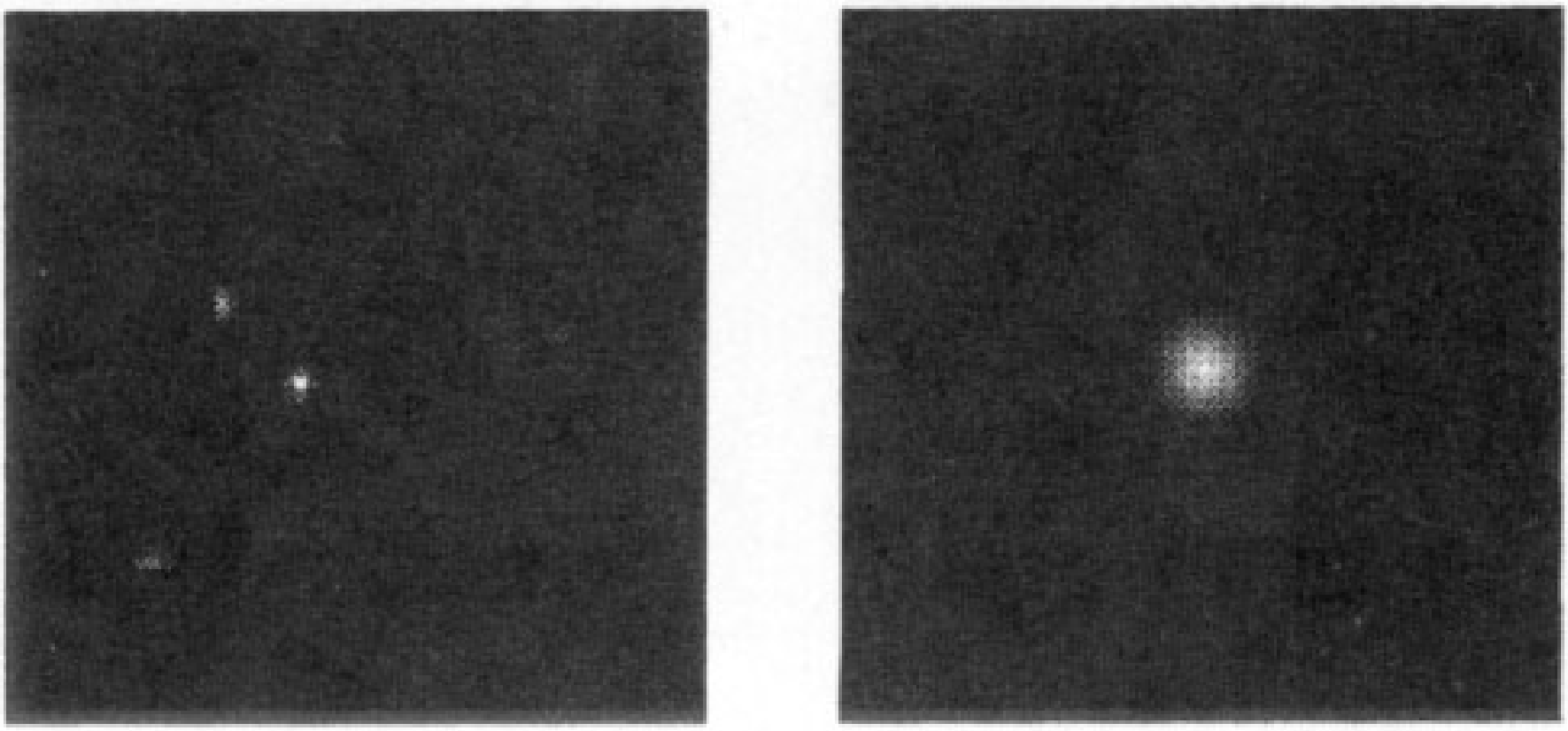
(c)



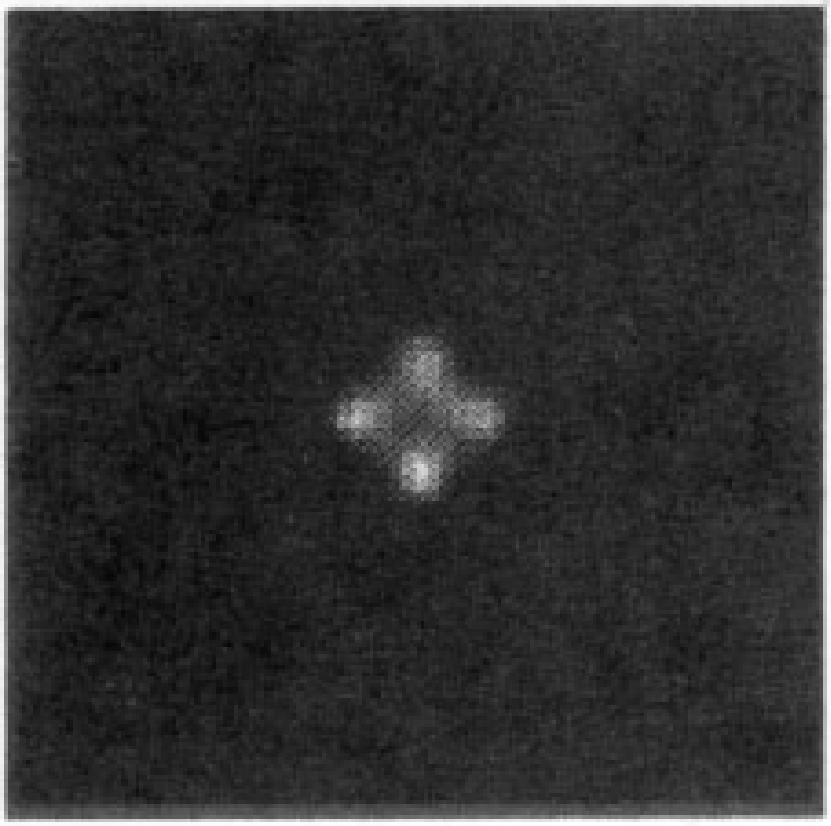
(d)



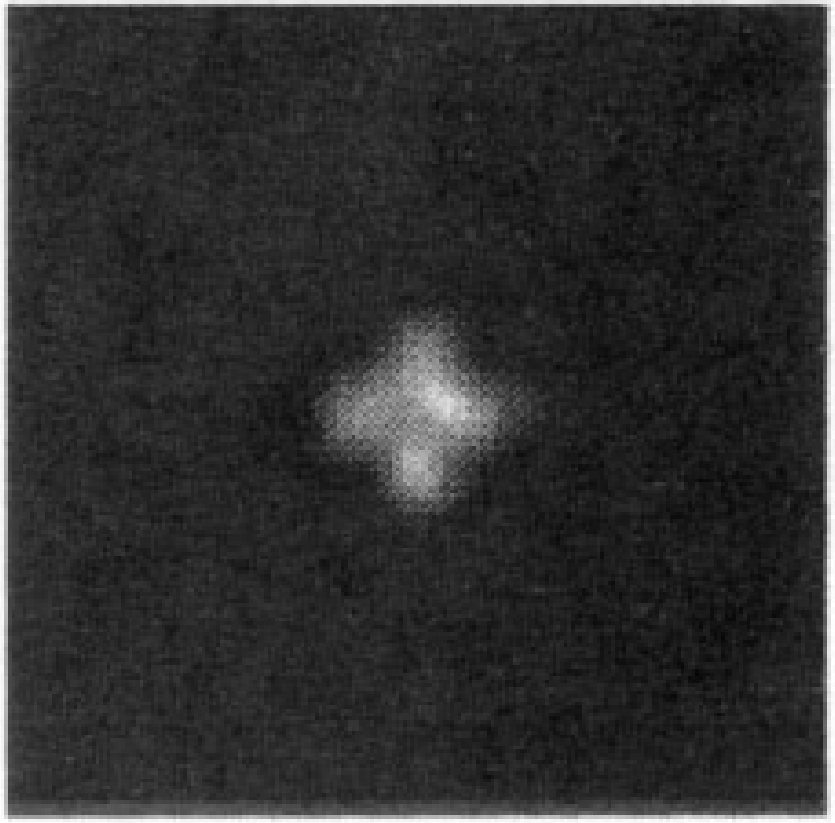
(a)



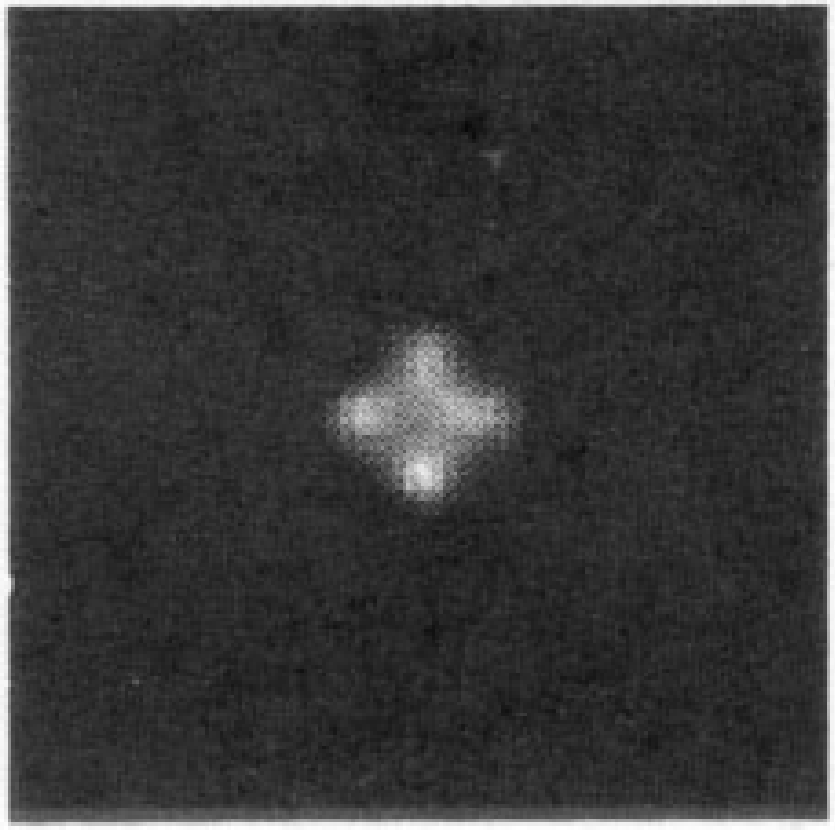
(b)



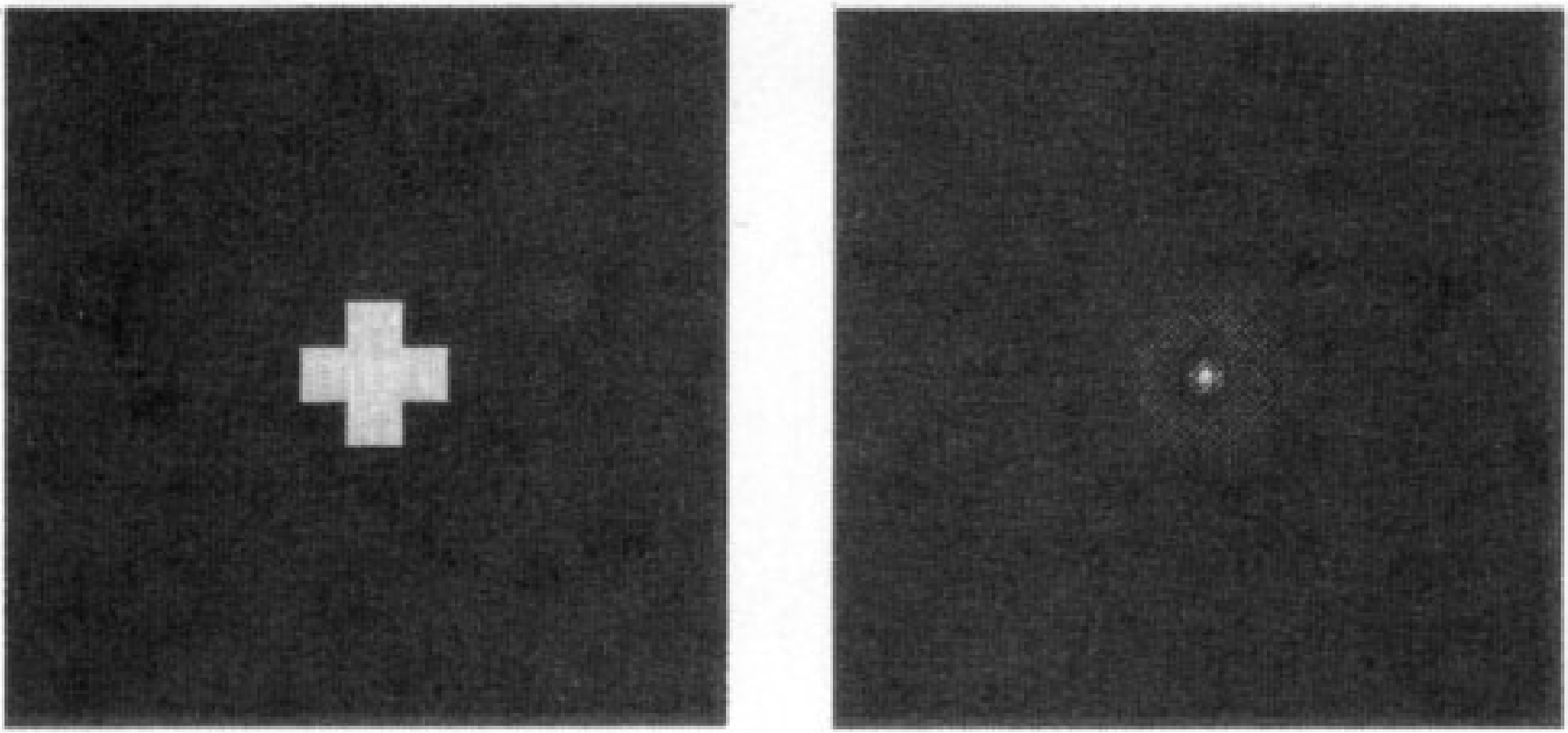
(a)



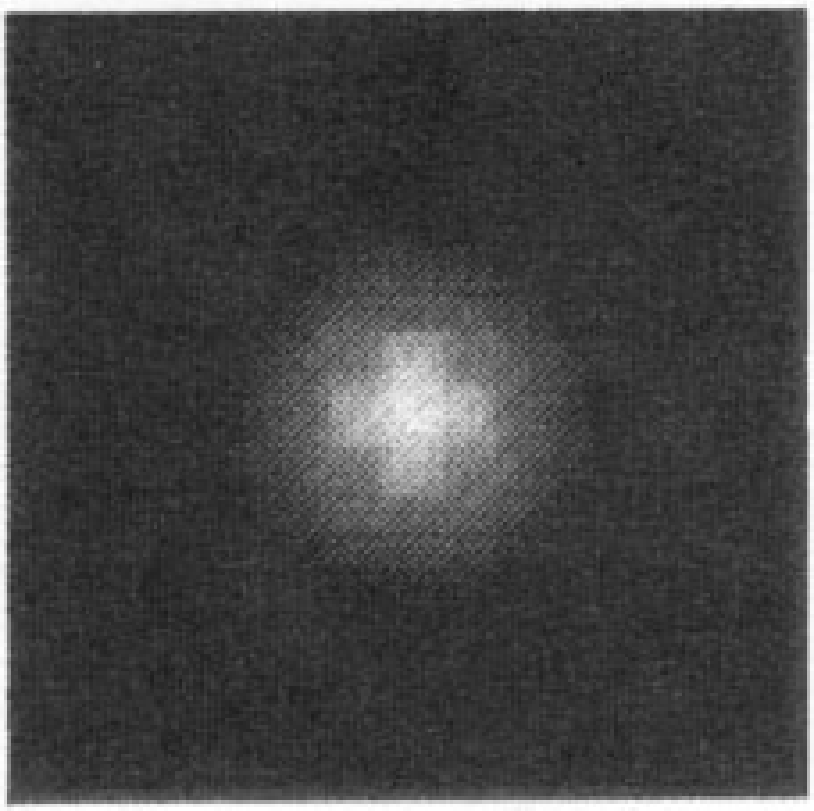
(b)



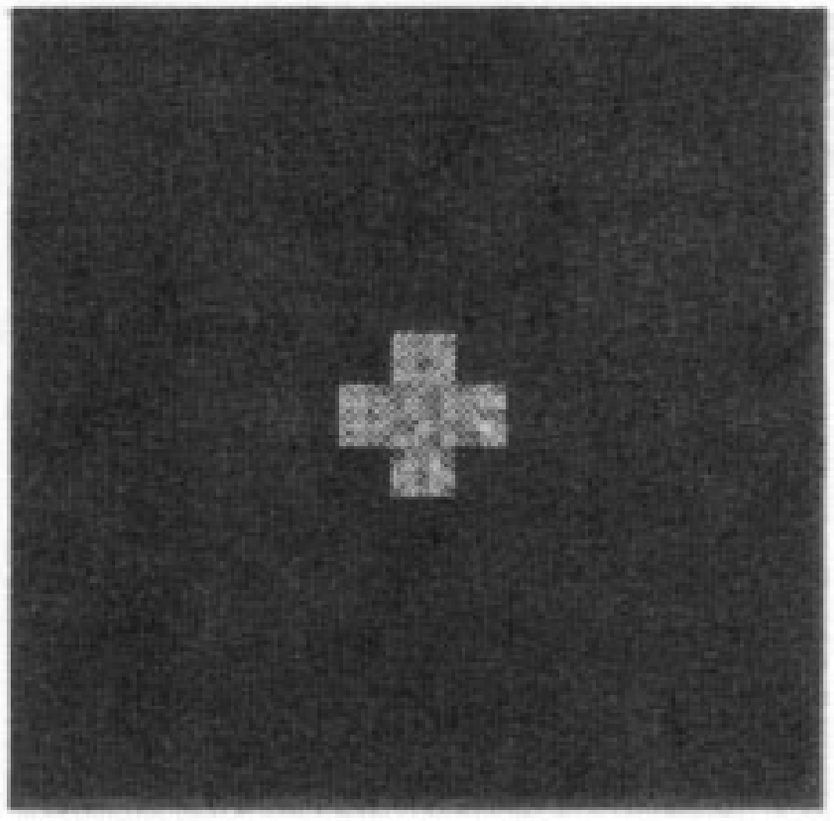
(c)



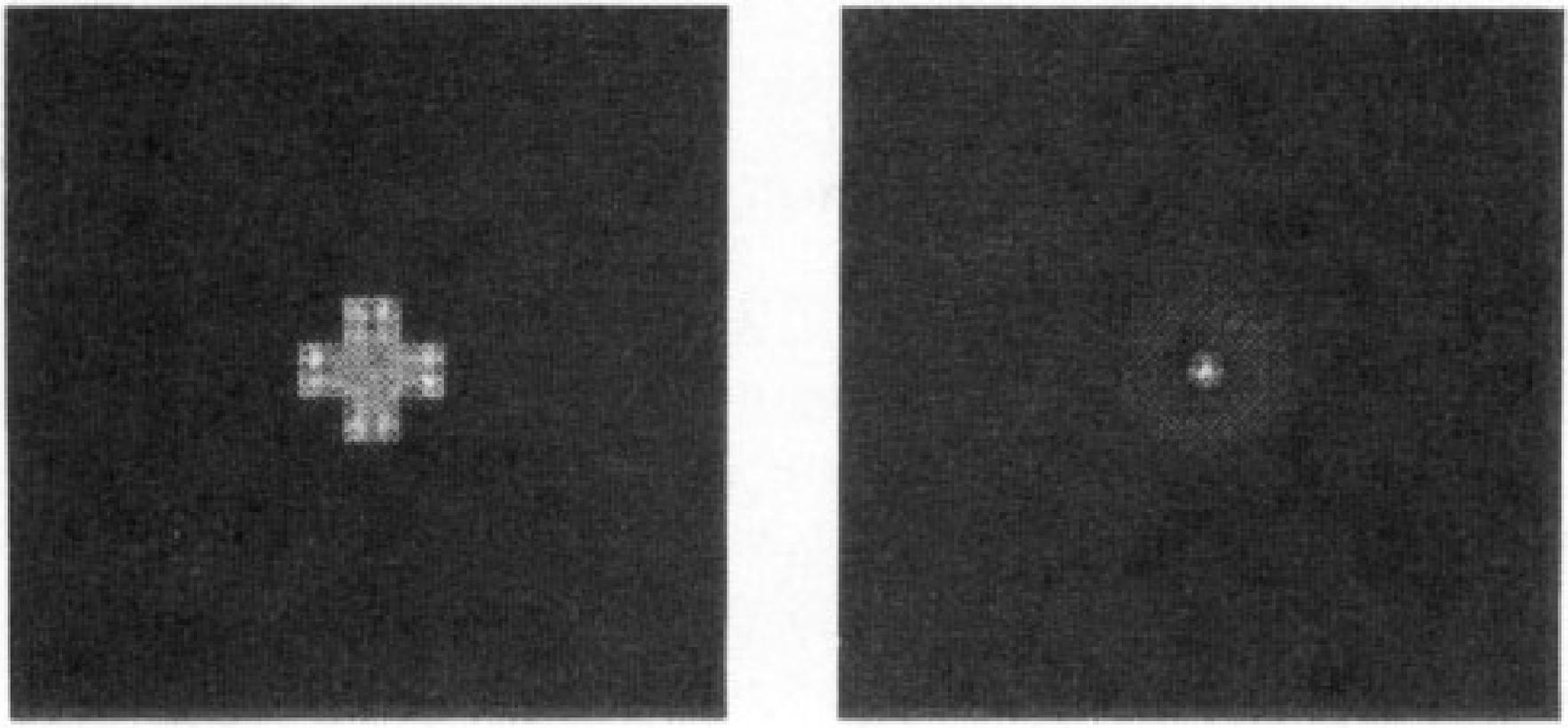
(a)



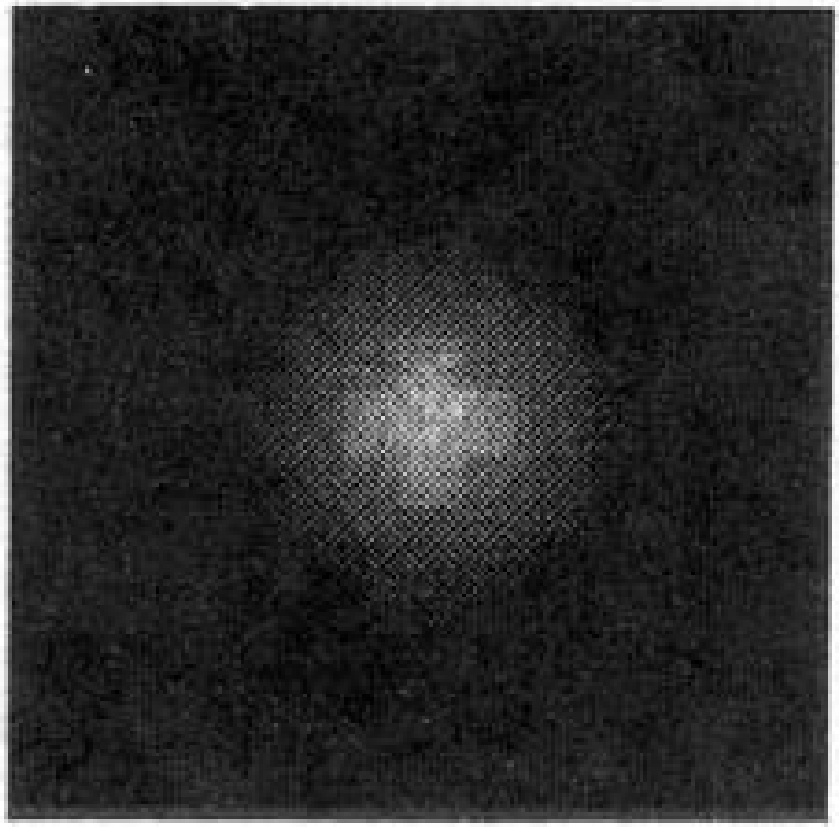
(b)



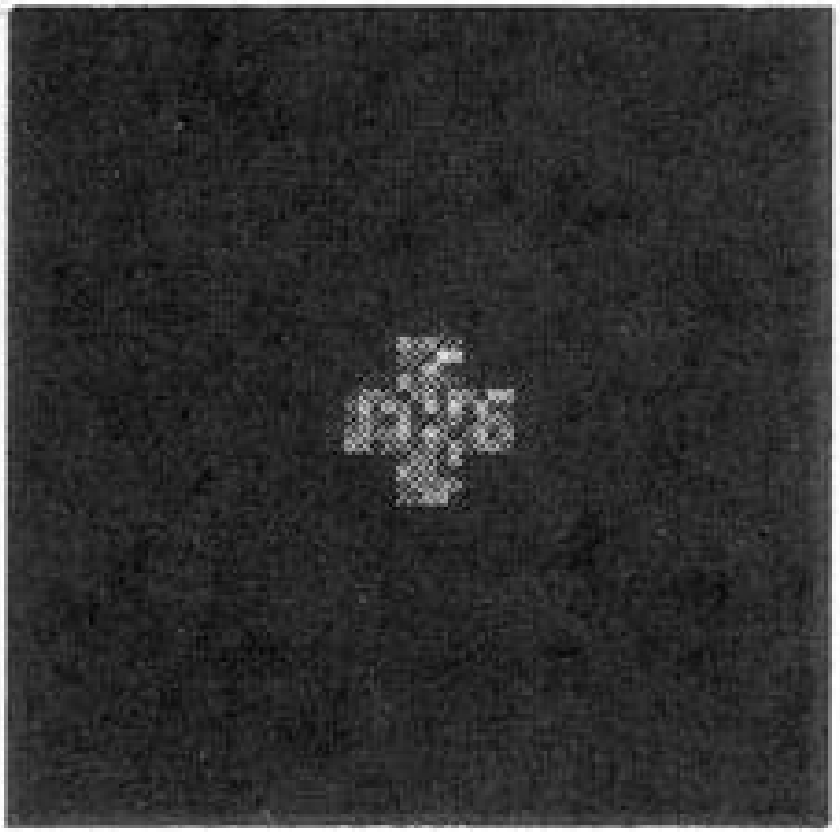
(a)



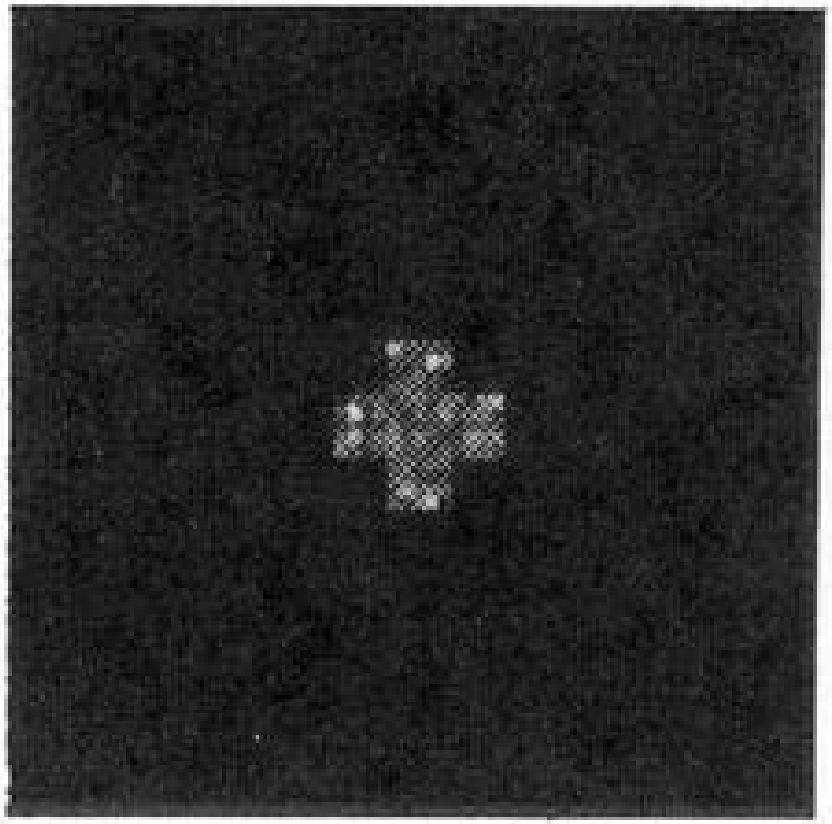
(b)

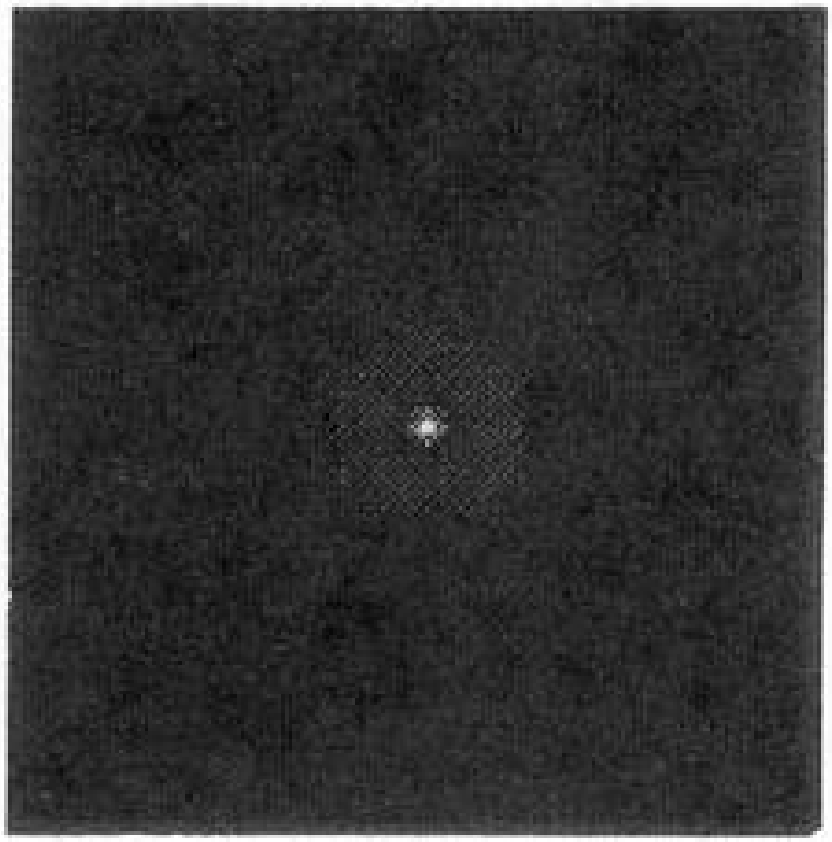


(a)



(b)





(c)