On convergence of the Schulz-Snyder phase retrieval algorithm to local minima

Kerkil Choi, Aaron D. Lanterman, and Raviv Raich

School of Electrical and Computer Engineering Georgia Institute of Technology, Mail Code 0250, Atlanta, GA 30332

The Schulz-Snyder iterative algorithm for phase retrieval attempts to recover a nonnegative function from its autocorrelation by minimizing the I-divergence between a measured autocorrelation and the autocorrelation of the estimated image.

We illustrate that the Schulz-Snyder algorithm can become trapped in a local minimum of the I-divergence surface. To show that the estimates found are indeed local minima, sufficient conditions involving the gradient and Hessian matrix of the I-divergence are given. Then, we build a brief proof showing how an estimate that satisfies these conditions is a local minium. The conditions are used to perform numerical tests determining local minimality of estimates. Along with the tests, related numerical issues are examined, and some interesting phenomena are discussed.

OCIS codes: 100.5070, 100.3190, 100.3010

1. Introduction

The goal of phase retrieval is to infer Fourier phase information given only Fourier magnitude data.[1−12] Phase retrieval is quite challenging. Phase retrieval is equivalent to the problem of recovering a function from its autocorrelation. One fundamental problem, as mentioned by a reviewer, is that there may be multiple signals whose autocorrelations are the same. Moreover, for iterative algorithms intended to minimize an objective function, the algorithm may converge to local minima that do not correspond with a global minimum. In addition to these problems, some iterative phase retrieval algorithms may suffer from stagnation, where the algorithm gets stuck on an estimate that does not correspond to a local minimum of the objective function.[13] The stagnation problems in Fienup's algorithm, which seems to be the most widely known of the phase retrieval methods, have been discussed by Fienup and Wackerman.[13] This paper investigates the problem of convergence to local minima for a particular technique, which we call the Schulz-Snyder algorithm.

Schulz and Snyder[12] developed an iterative method for phase retrieval that works entirely in the spatial domain, instead of alternating between the frequency and spatial domains as in Fienup's algorithm. Although Schulz and Snyder did not claim that their algorithm would never be subject to convergence to local minima (as opposed to a global minimum), they were encouraged by the fact that no such problems were observed in any of their experiments. Unfortunately, we have found that for certain cases, *the Schulz-Snyder algorithm can converge to an incorrect solution.* The Schulz-Snyder algorithm is based on minimizing an information-theoretic distance.

We illustrate that incorrect solutions correspond to local minima on the surface of this objective function. This offers the hope of improving the algorithm by providing ways for it to escape local minima. To our knowledge, this paper is the first to report on and characterize the problem of convergence to local minima for this algorithm.

Since an analytic proof ensuring convergence of the iteration to a local minimum (as opposed to a saddle point) does not guarantee that a practical solution, obtained by the algorithm in a finite number of iterations, is a local minimum for sure, we take a numerical approach to confirming convergence to local minima. Our approach checks a set of sufficient conditions for a local minimum that involve the first and second derivatives of the objective function being minimized. All of our experiments use simulated data, so we can plug in the "truth" as the initial estimate to find a true global minimum, and then explore additional local minima.

This paper is organized as follows. Section 2 reviews the Schulz-Snyder algorithm and defines notation that will be used throughout this paper. Sufficient conditions for local minima of the objective function surface are presented in Section 3, while numerical experiments illustrating local minima are given in Section 4. Our discussion is then concluded in Section 5.

2. The Schulz-Snyder Algorithm

The Schulz-Snyder algorithm[12] is an iterative method for recovering nonnegative functions from their n-th order correlations. Here, we focus specifically on the $n = 2$ case of recovery from autocorrelations, which is equivalent to phase retrieval. The algorithm estimates images from their autocorrelations by minimizing an information-theoretic

distance between measured autocorrelation data and the autocorrelation of the estimated image. The distance used is Csiszar's I-divergence,[14] which is a generalization of the Kullback-Leibler distance.[15] It is defined as

$$D[S, R_f] = \sum_y [R_f(y) - S(y)] + \sum_y S(y) \ln \frac{S(y)}{R_f(y)}, \tag{1}$$

where $S = R_g$ is the autocorrelation of some true but unknown g that we want to reconstruct from S, and the autocorrelation of an estimate f is defined as

$$R_f(y) = \sum_x f(x) f(x + y). \tag{2}$$

The purpose of the algorithm is to minimize the following objective function:

$$\begin{aligned} J(f) &= D[S, R_f] \\ &= \sum_y [R_f(y) - S(y)] + \sum_y S(y) \ln \frac{S(y)}{R_f(y)} \end{aligned} \tag{3}$$

subject to the constraints

$$\begin{aligned} I(f) &= \sum_y f(y) = I(g) \\ f &\geq 0, \end{aligned} \tag{4}$$

where $I(f)^2 = \sum_y R_f(y)$, which results from Property 3.4 in Schulz and Snyder.[12] Note that $I(g)$ can be obtained even if g is not known.

The Schulz-Snyder algorithm for recovering a nonnegative function from its autocorrelation is specified by the iteration

$$\begin{aligned} f_{k+1}(x) &= f_k(x) \frac{1}{2I(f)} \sum [f_k(x + y) + f_k(x - y)] \frac{S(y)}{R_{f_k}(y)} \\ &= f_k(x) \frac{1}{I(f)} \sum f_k(x + y) \frac{[S(y) + S(-y)]}{2R_{f_k}(y)}. \end{aligned} \tag{5}$$

Note that if $f_0(x) = 0$ for some particular x, then $f_k(x) = 0$ for that x for all k. This provides a convenient way of incorporating support constraints if they are available.

3. Sufficient Conditions for Local Minima

This section develops criteria for determining whether or not the algorithm has indeed converged to a local minimum. In practice, it is not automatically guaranteed that the final estimate is a local minimum, even when the convergence of the algorithm, given a theoretically unlimited number of iterations, is assured by an analytic proof. This may be because the final estimate is a saddle point, or the algorithm may improve the estimates too slowly due to finite-precision numerical issues or the small curvature at the local minimum toward which the algorithm is headed. For these reasons, we suggest a set of sufficient conditions that can establish local minimality of a given estimate.

Generally speaking, it would be extremely difficult to tell whether or not a local minimum is a global minimum. For this reason, we conduct experiments where the true images are known, so that we may know for sure that plugging the "right answer" into the algorithm corresponds to a global minimum.

Before presenting the sufficient conditions, we define two sets that will be useful in later discussions.

Definition. *The two sets $\mathcal{S}_1$ and $\mathcal{S}_2$ are called* index sets *and defined as follows:*

$$\mathcal{S}_1 = \left\{ i \;:\; \frac{\partial J(f)}{\partial f(x_i)} \;=\; 0, \;\; x_i \in \chi \right\}, \tag{6}$$

$$\mathcal{S}_2 = \left\{ i \;:\; \frac{\partial J(f)}{\partial f(x_i)} \;>\; 0, \;\; x_i \in \chi \right\}, \tag{7}$$

where χ represents the two-dimensional set $\{1, 2, \ldots, N\} \times \{1, 2, \ldots, M\}$.

The I-divergence in the Schulz-Snyder algorithm is guaranteed to be non-increasing as iterations proceed.[12] Hence, if the algorithm has converged to a critical point, it cannot be a maximum; it must be either a minimum or a saddle point. Therefore, at convergence, $\mathcal{S}_1 \bigcup \mathcal{S}_2 = \mathcal{S}$, where $\mathcal{S}$ is the set of all the parameter indices.

The sufficient conditions for local minimality of a given estimate are given in the following theorem.

Theorem. (Sufficient conditions for a local minimum) *If an estimate f^* satisfies the following conditions:*

1. *The given estimate satisfies the Kuhn-Tucker conditions:*

$$\left.\frac{\partial J(f)}{\partial f(x)}\right|_{f=f^*} \begin{cases} = 0 & f^*(x) > 0 \\ \geq 0 & f^*(x) = 0. \end{cases} \tag{8}$$

2. *The Hessian matrix H whose (i,j)-th element is defined by*

$$H_{ij} = \left.\frac{\partial^2 J(f)}{\partial f(x_i)\partial f(x_j)}\right|_{f=f^*}, \; i \in \mathcal{S}_1, \; j \in \mathcal{S}_1 \tag{9}$$

is positive definite,

*then f^* is a local minimum. In other words, for any f that satisfies the given constraints (*i.e.*, nonnegativity) and is in the neighborhood of f^*, the following inequality is satisfied:*

$$D[S, R_{f^*+\Delta f}] - D[S, R_{f^*}] = J(f^* + \Delta f) - J(f^*) > 0, \tag{10}$$

where $\Delta f = f - f^*$.

Proof. See Appendix A. □

Although the conditions just mentioned are clear theoretically, they may involve various numerical issues in practice, such as determining when quantities are numerically zero (which influences construction of the two index sets) and the precision of the algorithm. Because such issues are quite problem-dependent and cannot be resolved by a single general rule, we address the issues based on problems under consideration in a reasonable way that takes experiments into consideration.

The first derivative of the I-divergence, which is implicitly embedded in the algorithm, has been derived by Schulz and Snyder:

$$\frac{\partial J(f)}{\partial f(x_i)} = 2I(f) - \sum_y \{f(x_i+y) + f(x_i-y)\}\frac{S(y)}{R_f(y)}. \tag{11}$$

In this equation, the x_i can be in either $\mathcal{S}_1$ or $\mathcal{S}_2$. The second partial derivative, derived in Appendix B, is given by

$$\begin{aligned}\frac{\partial^2 J(f)}{\partial f(x_i)\partial f(x_j)} &= 2 + \sum_y \{f(x_i+y) + f(x_i-y)\}\{f(x_j+y) + f(x_j-y)\}\frac{S(y)}{R_f(y)} \\ &\quad - \left(\frac{2S(x_j-x_i)}{R_f(x_j-x_i)}\right).\end{aligned} \tag{12}$$

4. Experiments

A. Preliminary Remarks

1. Convergence Criteria

Due to the buildup of numerical errors, an iterative algorithm implemented on a computer with finite-precision arithmetic may not *exactly* settle on a fixed-point solution. The algorithm may wander slightly about a point at which it is supposed to stabilize,

or take excruciatingly tiny steps towards the point. For these reasons, it is important to establish a meaningful stopping criterion before discussing other matters. Various choices appear in literature. We are tempted to choose I-divergence values in our stopping criterion. However, we have observed that the algorithm may make further improvements in solutions even while the numerically computed I-divergence wanders up and down slightly about a value. Hence, we choose a stopping criterion based on the pixel values themselves:

Stoppping Criterion. *The algorithm is halted when the following condition is satisfied:*

$$\max_i |f_k(x_i) - f_{k-1}(x_i)| < \epsilon, \tag{13}$$

where ϵ is a positive constant.

The choice of epsilon depends on the experiments to be performed. It may be related to precisions of the operations involved in the algorithm. Although the epsilon may be selected by analyzing the possible errors (or precisions), it would be quite difficult to estimate all the errors in advance. Since we know the "truth" in our experiments, we had the benefit of choosing appropriate epsilons by plugging the truth into the algorithm and seeing how far the estimate slides off of the truth due to numerical issues. This paper focuses on probing the limits of the algorithm, and hence we select our epsilons rather strictly. In practice, we would recommend epsilons that are are larger, i.e. looser, than those used in this study.

Table 1 shows the epsilon values used for the experiments in this study. In the following tables, the results of one experiment include the original image and esti-

mates, autocorrelations, gradient images, and line plots (if contained). For instance, the results of Exp. 1, abbreviating Experiment 1, are shown in the set of images in Fig. 1 through Fig. 3, and the results of Exp. 7 are shown in the set of images in Fig. 22 through Fig. 25.

Table 1. Choices of the epsilon for the experiments in this study

	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6	Exp. 7	Exp. 8
ϵ	10^{-13}	10^{-13}	10^{-7}	10^{-7}	10^{-7}	10^{-8}	10^{-7}	10^{-8}

Note that the chosen epsilons are not the same for all the experiments. For instance, in Exps. 1 and 2, the algorithm sometimes require far more iterations (compare Tables 5 and 6 with Tables 7 to 11) than the others, and we need a rather precise epsilon to convince ourselves that the algorithm has surely converged. Note that it is not necessarily true that we need smaller epsilons for larger parameter spaces; compare Exps. 1 and 2 with Exp. 3 (see Table 1). It is also not true that smaller epsilons always require the algorithm to take more iterations, as seen by comparing Exp. 3 with Exp. 6 (see Tables 7 and 10). The choice seems to depend on where a local minimum is located on the surface of the I-divergence function, which we cannot know until the algorithm converges. As just mentioned, it would be extremely difficult to predict what choices of the epsilon are appropriate ahead of time, since the choices may depend on different aspects of the algorithm in unfathomable ways. Therefore, we determine them by trial and error.

Numerical concerns may be sensitive to implementation issues. All our experiments were performed with MATLAB 6.5 from The MathWorks on a machine of Intel Pentium IV 2.0 GHz. The precision of the machine is 10^{-16}. This number is determined by selecting the maximum δ such that $1 + \delta = 1$ in a given machine. Any δ smaller than 10^{-16} added to 1 produces 1, and any δ larger than 10^{-16} added to 1 produces a larger number than 1, in our machine.

2. Initial Estimates

Throughout the paper, *unless stated otherwise*, the algorithm is initialized with a rectangle with constant intensity plus a small amount of uniform random noise (we use "constant + 0.1×random noise"), which is scaled according to the constraint in Eq. (4). In our experiments, the selected constant is 1, and the uniform random noise takes values on between 0 and 1. This small amount of noise is helpful to prevent the algorithm from getting stuck on artificially symmetric images, as discussed in the right column of [12, p. 1269]. One might wonder if different realizations of this small additive noise would result in the algorithm converging to radically different answers. In all our experiments, we never found this to be the case. We found that significantly different initial estimates were needed to obtain different results.

The sizes of the initial estimates are chosen to be as small as possible to minimize accumulated computational errors resulting from a large number of parameters. The size of the initial images for all experiments except for Exp. 3 is chosen to be 16 × 16 pixels because images of the size manifest the effects that we want to show (i.e., local minima). On the other hand, the size for Exp. 3 is chosen to be 32 × 32 pixels

because initial images smaller than that did not produce the effects that we want to illustrate.

When the initial estimates are put into the algorithm, they are zero-padded for ease of implementation; we implement the correlation operations in the Schulz-Snyder algorithm with fast Fourier transforms. The sizes of all the original true images are the same as those of the corresponding initial images. In the images shown throughout the paper, we show only a subset of the full image to illustrate detail; surrounding pixels may be considered to be zero. Brighter pixels represent smaller values, and darker pixels represent larger values. Also, all the images in this paper are scaled to fit the full dynamic range when they are displayed.

We show the gradient image of the initial estimate for each experiment in Figs. 3(b), 6(b), 9(b), 12(b), 16(b), 20(b), 24(b) and 28(b). The gradient images are also scaled to use the full dynamic range, hence it may be difficult to compare pixel values from image to image. Hence, Table 2 shows the range of the gradient values. Furthermore, we add a column containing the numbers of the gradient values between -0.1 and 0.1. The ranges and numbers may vary slightly with the choice of initial images, but not significantly. In Table 2, f_0 denotes an initial estimate, and ∇ denotes gradient. The elements of the set $\mathcal{A}$ are the gradient values between -0.1 and 0.1, and the symbol $|\mathcal{A}|$ denotes the cardinality of $\mathcal{A}$.

Functions shifted and/or rotated by 180 degrees have exactly the same autocorrelations as the original function. Therefore, the final estimates of some experiments are shifted and/or rotated by 180 degrees from their original versions. (*e.g.*, see Figs. 7 and 10).

Table 2. The range of gradient values for the initial images , and the number of the gradient values between -0.1 and 0.1.

	$\max \nabla f_0$	$\min \nabla f_0$	$\lvert\mathcal{A}\rvert$
Exp. 1	58.66	-39.81	1
Exp. 2	86.33	-17.62	2
Exp. 3	305.99	254.64	0
Exp. 4	27.76	-27.14	1
Exp. 5	35.51	-32.58	2
Exp. 6	36.55	-28.74	1
Exp. 7	31.52	-20.75	2
Exp. 8	34.08	-20.79	2

3. Numerical Zero

Another issue we need to be careful about is the construction of the index sets $\mathcal{S}_1$ and $\mathcal{S}_2$. It may be controversial to say whether a gradient should be considered numerically zero or not, as pointed out by one of the reviewers. Hence, we take into account not only the gradient values but also the values of the final estimates. Recall that if the algorithm has converged, then nonzero-valued pixels of an estimate are supposed to have zero-gradient values. Therefore, we construct $\mathcal{S}_1$ with the "nonzero" pixels. Determination of the nonzeroness of pixels may also be controversial. We conservatively consider pixels whose values are greater than 10^{-4} to be the "nonzero" pixels.

Now, the gradient values corresponding to the nonzero pixels should be zero in theory. However, numerical limitations keep the values from becoming exactly zero. Table 3 shows the maximum and minimum of the gradient values computed on the index set $\mathcal{S}_1$ in each experiment. As seen on the table, the gradient values are close to zero; if we run the algorithm further, then the values become smaller.

Table 3. Maximum and minimum of the gradient values for the final estimate corresponding to the index set $\mathcal{S}_1$ in each experiment.

	$\max \nabla f(\mathcal{S}_1)$	$\min \nabla f(\mathcal{S}_1)$
Exp. 1	1.2037×10^{-12}	5.1563×10^{-12}
Exp. 2	4.0412×10^{-13}	9.3425×10^{-13}
Exp. 3	9.3342×10^{-8}	9.7413×10^{-8}
Exp. 4	1.6614×10^{-7}	8.0929×10^{-7}
Exp. 5	2.0553×10^{-7}	3.4158×10^{-7}
Exp. 6	3.8334×10^{-8}	2.8034×10^{-6}
Exp. 7	8.8482×10^{-8}	1.0549×10^{-7}
Exp. 8	3.2379×10^{-6}	3.5768×10^{-9}

To convince ourselves that the construction of the index set $\mathcal{S}_1$ is correct, we investigate the gradient values computed on the zero-value index set $\mathcal{S}_2$ to see if there are any values with gradient values in the ranges given in Table 3. Table 4 shows the maximums of the estimated values corresponding to the index set $\mathcal{S}_2$, along with the

associated gradient values. The estimated values corresponding to the index set $\mathcal{S}_2$ are sufficiently small to claim their zeroness, and the associated gradient values are large enough to claim their nonzeroness. With low probability, there might be some pixels whose values are close to zero with gradient values close to zero, according to the theory of calculus. However, we have not met such pixels in any of our experiments. None of the gradient values associated with the index set $\mathcal{S}_2$ fell in the ranges given in Table 3.

Table 4. Maximum of the values of the final estimate corresponding to the index set $\mathcal{S}_2$, along with the associated gradient value in each experiment.

	$\max f(\mathcal{S}_2)$	$\nabla\{\max f(\mathcal{S}_2)\}$
Exp. 1	1.7509×10^{-9}	5.2072×10^{-5}
Exp. 2	6.3692×10^{-12}	1.8345×10^{-4}
Exp. 3	4.5767×10^{-149}	8.8046×10^{-3}
Exp. 4	4.7416×10^{-166}	6.5779×10^{-3}
Exp. 5	6.0390×10^{-167}	2.5424×10^{-2}
Exp. 6	9.0652×10^{-9}	1.2225×10^{-4}
Exp. 7	4.4466×10^{-323}	5.2912×10^{-2}
Exp. 8	1.1884×10^{-7}	3.3057×10^{-3}

4. *Positive Definiteness*

Positive definiteness of the Hessian matrices is tested by checking the eigenvalues of the matrices. The eigenvalues are computed with the MATLAB command "eig." Even though we computed all the eigenvalues associated with each matrix, we only show the minimum and maximum eigenvalues for brevity.

B. *Radically Different Results*

1. *Experiment 1*

Figure 1(a) shows an original image, which is quite simple, and Fig. 1(c) shows the final estimate of the original image reconstructed by the Schulz-Snyder algorithm. As mentioned, the final estimate is obtained by halting the algorithm when the given stopping criterion is met. The estimate looks fairly different from the original image. Because the estimate stayed visually changeless even after many more iterations, we suspected that the estimate was one of the local minima. To confirm this conjecture, we test the sufficient conditions discussed in the preceding section. Table 5 shows I-divergence values for the original true image, the initial estimate, and the final estimate, denoted by $D[S, S]$, $D[S, R_{f_0}]$, and $D[S, R_{f_k}]$, respectively, where f_k denotes the final estimate, and k is the number of iterations when the final estimate is obtained. The I-divergence value for the original true function is supposed to be zero. However, due to numerical errors, it moved about the I-divergence value given in Table 1 as iterations proceeded; the value is recorded after one iteration. Note that the I-divergence value for the final estimate is relatively large in comparison with the I-divergence value for the truth, even though the image size is small. The value of

the norm used in the stopping criterion, $\max_i |f_k(x_i) - f_{k-1}(x_i)|$, is given. The information on the Hessian matrix described in the sufficient conditions is also contained. Based on the minimum eigenvalue of the matrix, the Hessian matrix formed from the set $\mathcal{S}_1$ is positive definite. The size of the Hessian matrix is also given with other information.

Table 5. Selected data from Exp. 1.

Quantity	Value
$D[S, S]$	6.0646×10^{-13}
$D[S, R_{f_0}]$	3220.7138
$D[S, R_{f_k}]$	4.5947
k	308320
$\max_i \|f_k(x_i) - f_{k-1}(x_i)\|$	9.9920×10^{-14}
size(H)	120×120
max{eigenvalues(H)}	484.8513
min{eigenvalues(H)}	0.02472

Figures 2(a) and 2(c) show the autocorrelations corresponding to Fig. 1(a) and 1(c) respectively. Note that the autocorrelations in Fig. 2(a) and 2(c) look almost the same. As mentioned by a reviewer, there may be more than one signal that has the *exact* same autocorrelation. To convince the reader that this experiment does not fall into such a case of non-uniqueness, we show the difference of the two autocorrelations.

The two images possess clear differences. Because the image is scaled for display, we also include the range of the values of the difference image in the caption. For the same purposes, we hereafter show difference images along with all autocorrelations.

Figures 3(a), 3(b), and 3(c) show the gradient images of the I-divergence computed at the true image, the initial estimate, and the final estimate, respectively. The gradient images are given to provide information on what the gradient looks like initially and how it has changed. However, since the images are scaled, they may be hard to compare if one is concerned with exact values. In such cases, one can refer to Tables 2, 3, and 4.

2. Experiment 2

Another interesting experiment was performed on this local minimum. We tried initializing the algorithm by a convex combination of the original image and the local minimum shown in Fig. 1(c):

$$f_{initial} = (1 - \alpha) f_{local} + \alpha f_{original}, \tag{14}$$

where $0 \leq \alpha \leq 1$. A small number 10^{-15} was added to the local minimum and the original image to make sure that all pixels would be allowed to take nonzero values and not be forced to always be zero.

Figure 4(c) shows the final estimate when $\alpha = 0.4$, and Fig. 4(b) shows what the initial image looks like. Even though this new estimate looks somewhat similar to the original image in Fig. 4(a), it shows distinctively different features on the edges. For example, some pixels near the center of the sector in the final estimate are erased out. Also, the two ends of the arc of the fan have lost their features. Figs. 5(a) and

5(c) show the autocorrelations of the images. Note that, although the original image and final estimate are quite different, the autocorrelations of these images are very similar. As before, to confirm that the new estimate is another local minimum, we tested the sufficient conditions and show some selected data from Exp. 2 in Table 2. Again, the I-divergence value computed at the final estimate is relatively large. The Hessian matrix is positive definite in terms of its eigenvalues. The associated gradient images are shown in Fig. 6.

Table 6. Selected data from Exp. 2.

Quantity	Value
$D[S, S]$	7.5830×10^{-13}
$D[S, R_{f_0}]$	716.3421
$D[S, R_{f_k}]$	6.3621
k	118113
$\max_i \lvert f_k(x_i) - f_{k-1}(x_i)\rvert$	9.9962×10^{-14}
size(H)	122×122
max{eigenvalues(H)}	493.0640
min{eigenvalues(H)}	0.0558

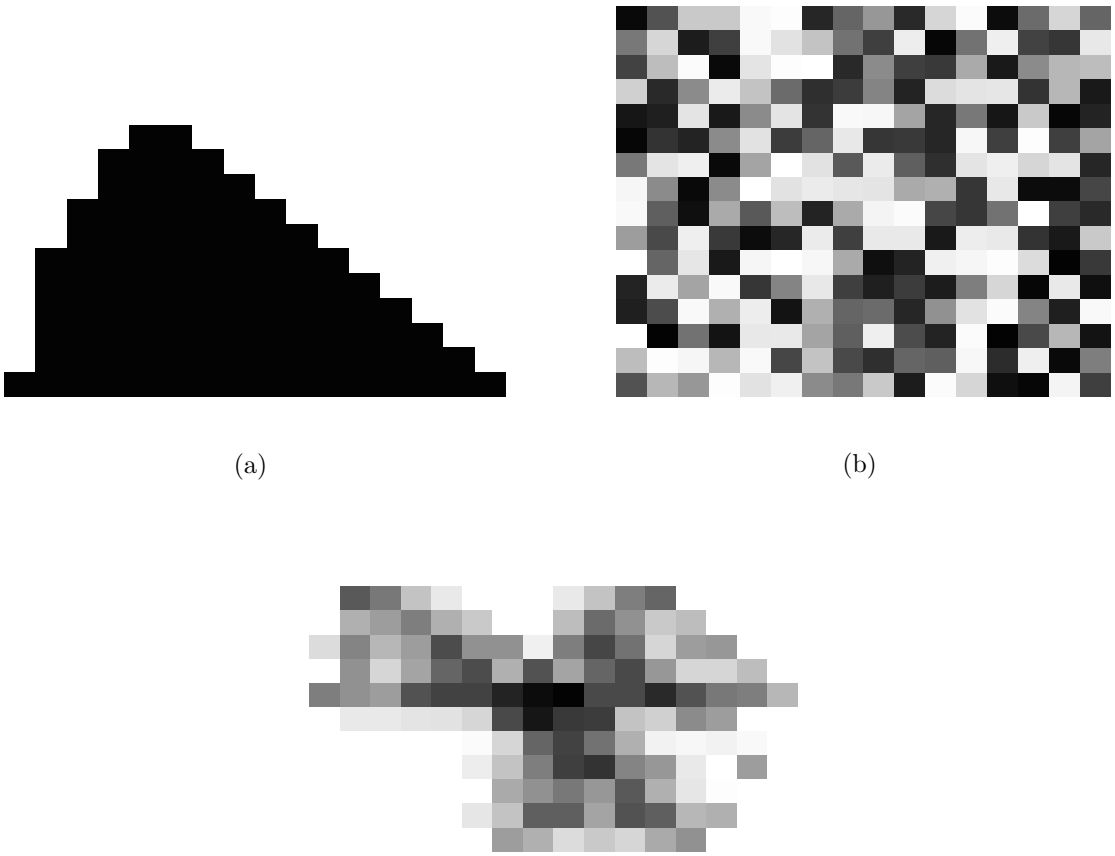
3. Experiment 3

Figure 7 shows another example of a local minimum. Figure 7(a) show the original true image consisting of a set of vertical lines to the right of a set of horizontal lines.

Figure 7(c) shows the corresponding final estimate. Figure 8 shows the associated autocorrelations. Note that the autocorrelations are quite similar, but the original image and final estimate have interesting differences. There does not seem to be a definite space between the two sets of lines in the reconstruction; in addition, the lines near the border of the two line sets become "dotted." We do not have an intuitive explanation for this intriguing behavior. To confirm that the final estimate is a local minimum, we again perform the tests on the sufficient conditions. Table 3 contains information about Exp. 3. The related gradient images are shown in Fig. 9.

Table 7. Selected data from Exp. 3.

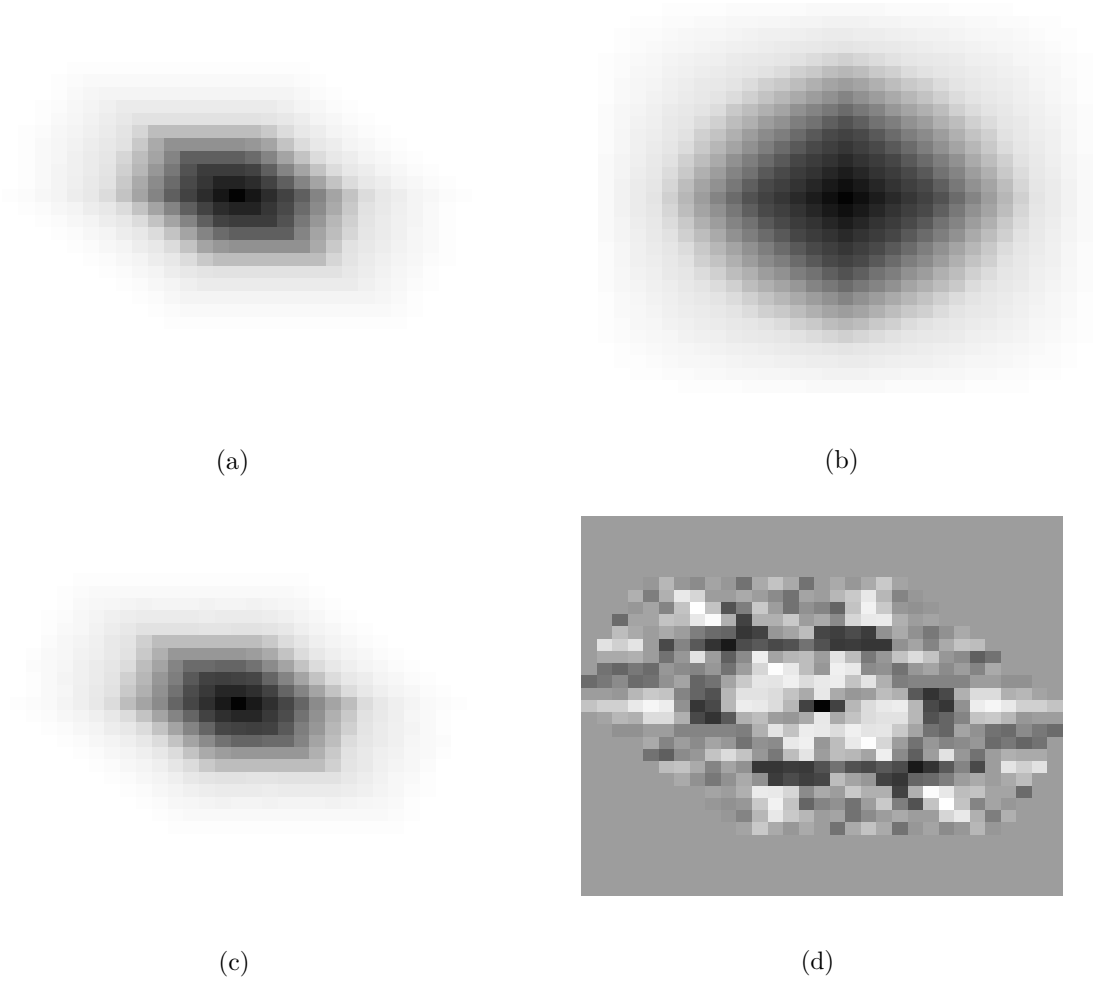
Quantity	Value
$D[S, S]$	8.9336×10^{-12}
$D[S, R_{f_0}]$	18021.1460
$D[S, R_{f_k}]$	45.0530
k	38210
$\max_i \lvert f_k(x_i) - f_{k-1}(x_i)\rvert$	9.9993×10^{-8}
size(H)	307×307
max{eigenvalues(H)}	1236.4256
min{eigenvalues(H)}	0.0327



(a) (b)

(c)

Fig. 1. (a) Original image. (b) Initial estimate. (c) Final estimate.



(a) (b)

(c) (d)

Fig. 2. Images associated with Fig. 1. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-1.37 1.84].

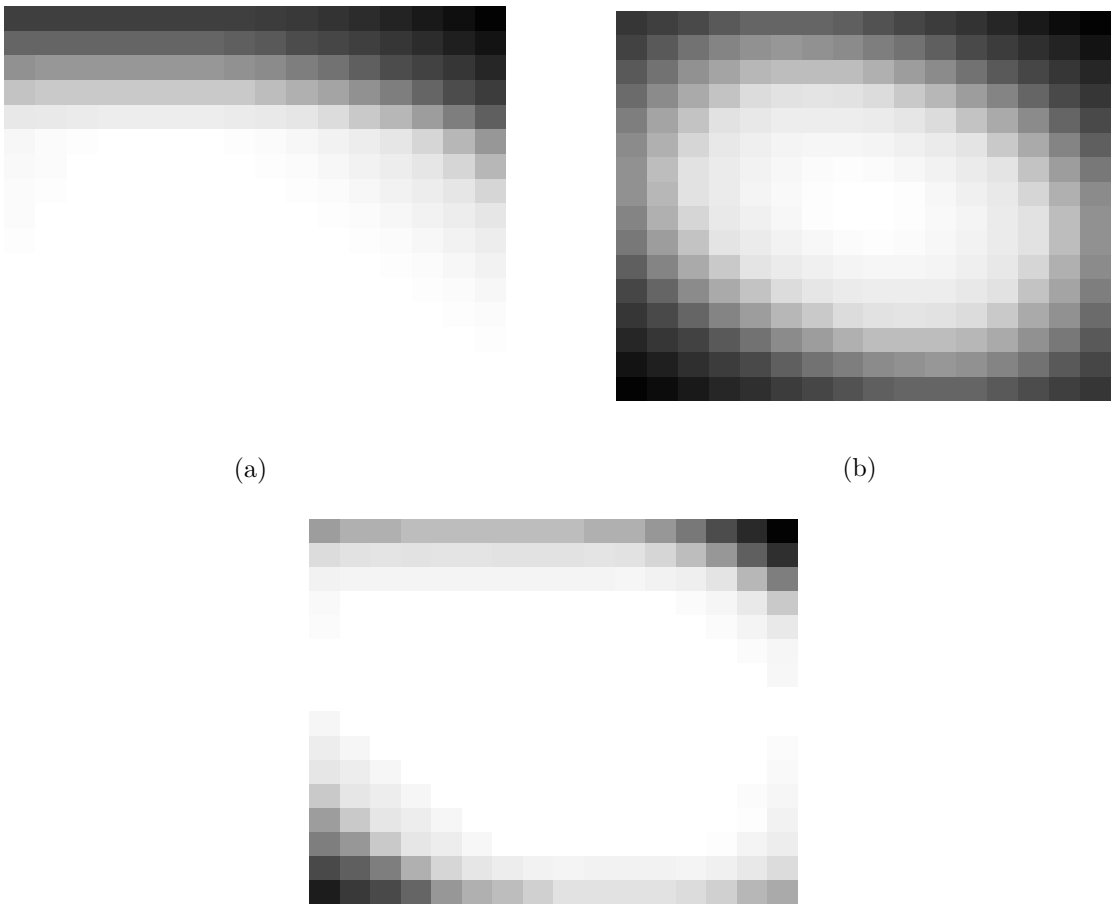
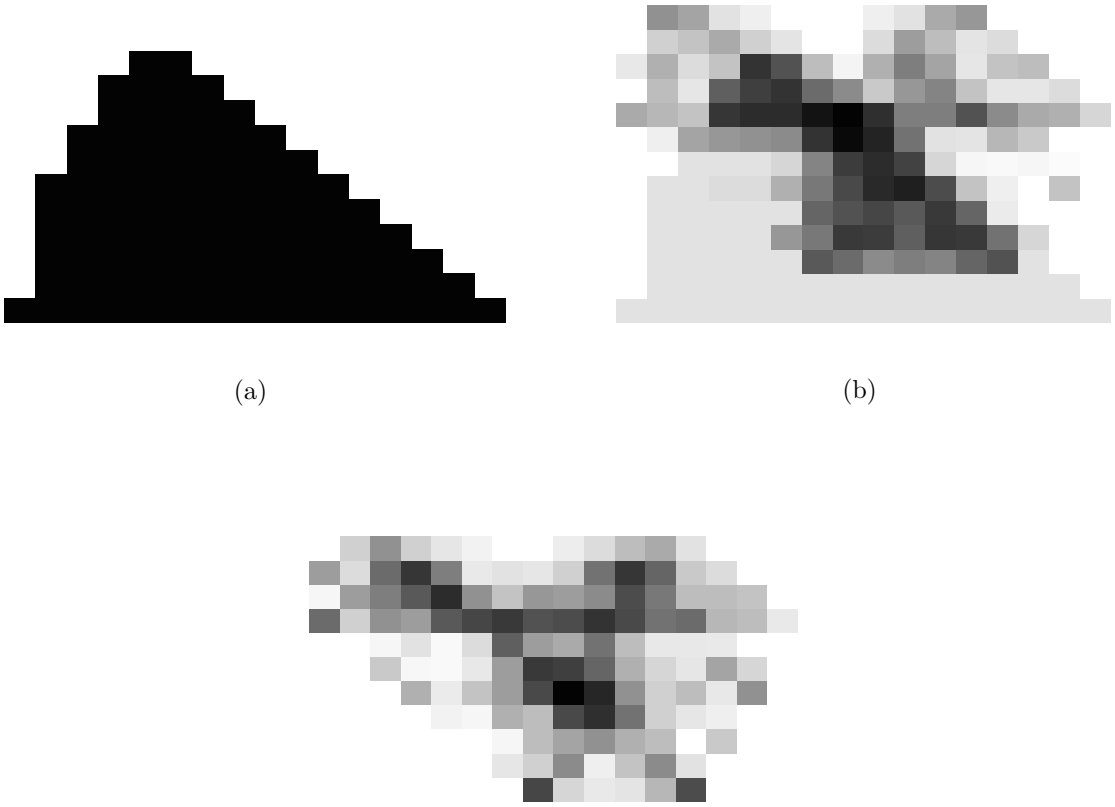


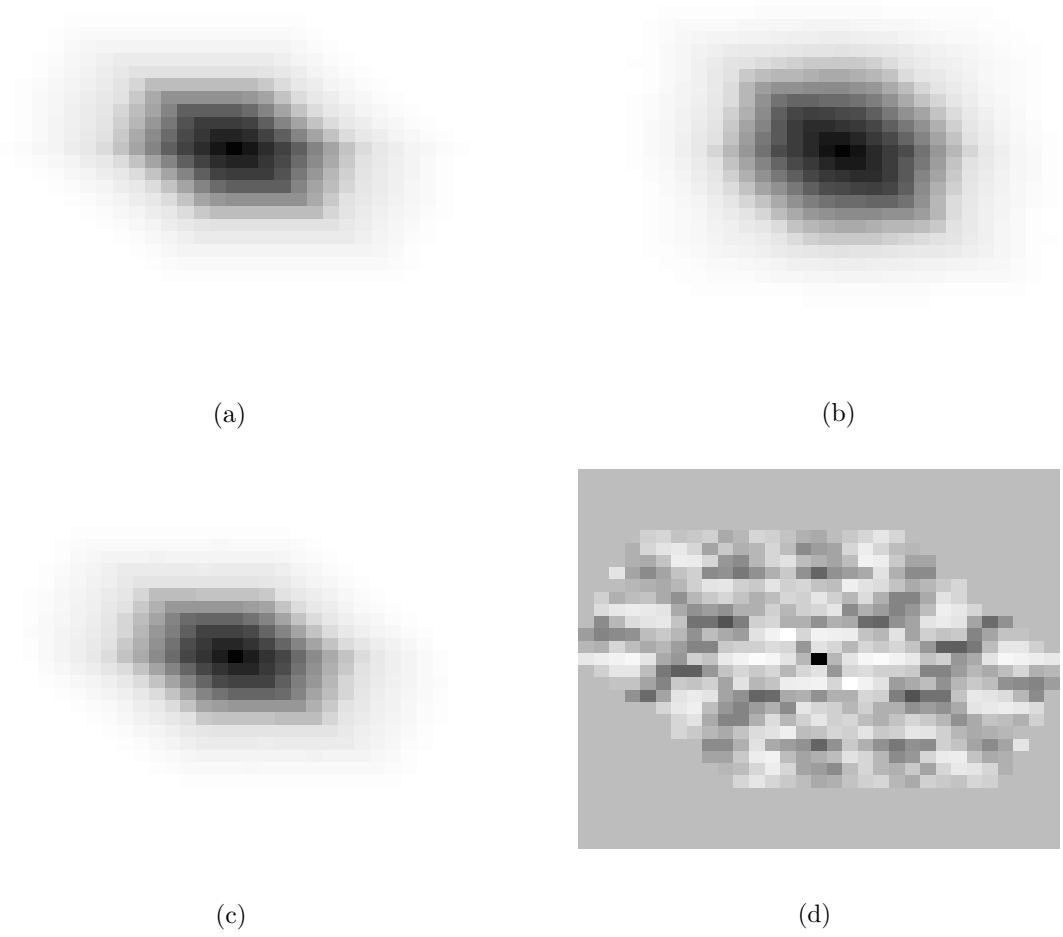
Fig. 3. Images associated with Fig. 1. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.



(a) (b)

(c)

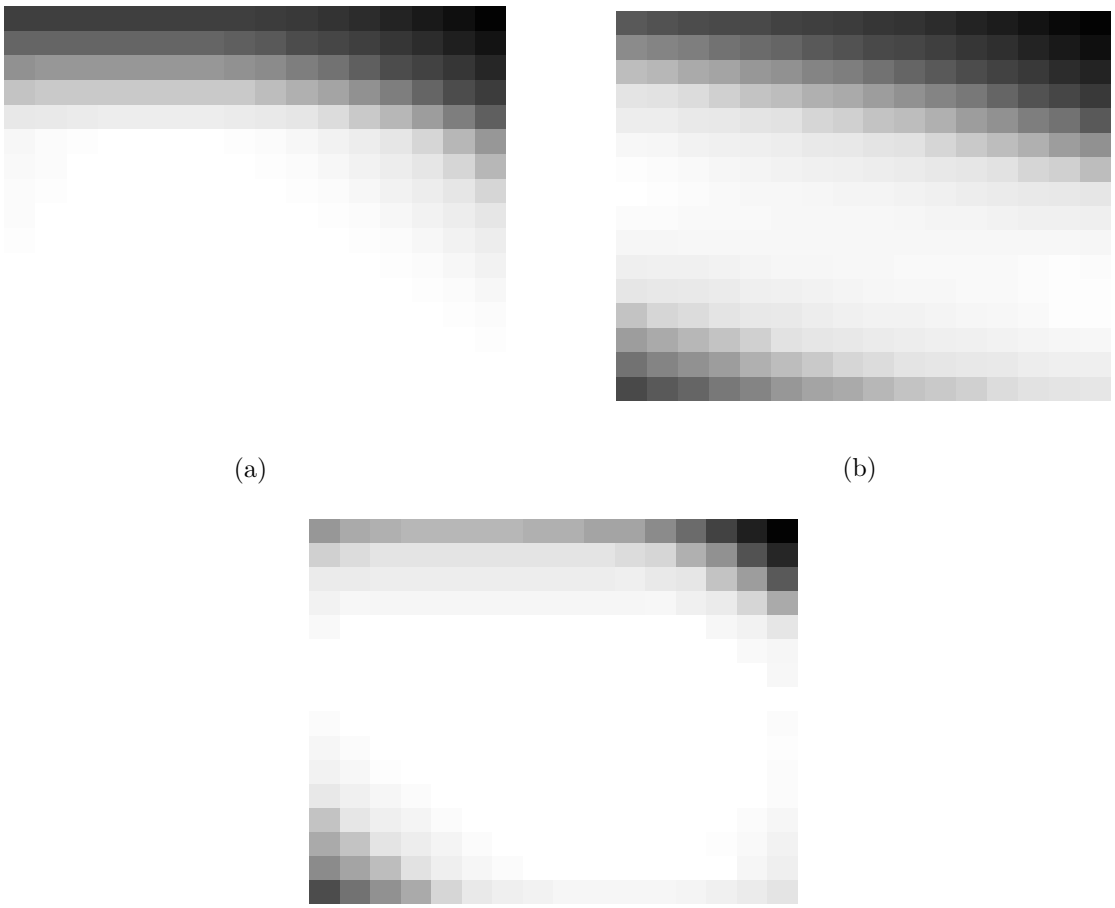
Fig. 4. (a) Original image. (b) Initial estimate. (c) Final estimate.



(a) (b)

(c) (d)

Fig. 5. Images associated with Fig. 4. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-1.39 2.64].



(a) (b)

(c)

Fig. 6. Images associated with Fig. 4. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.

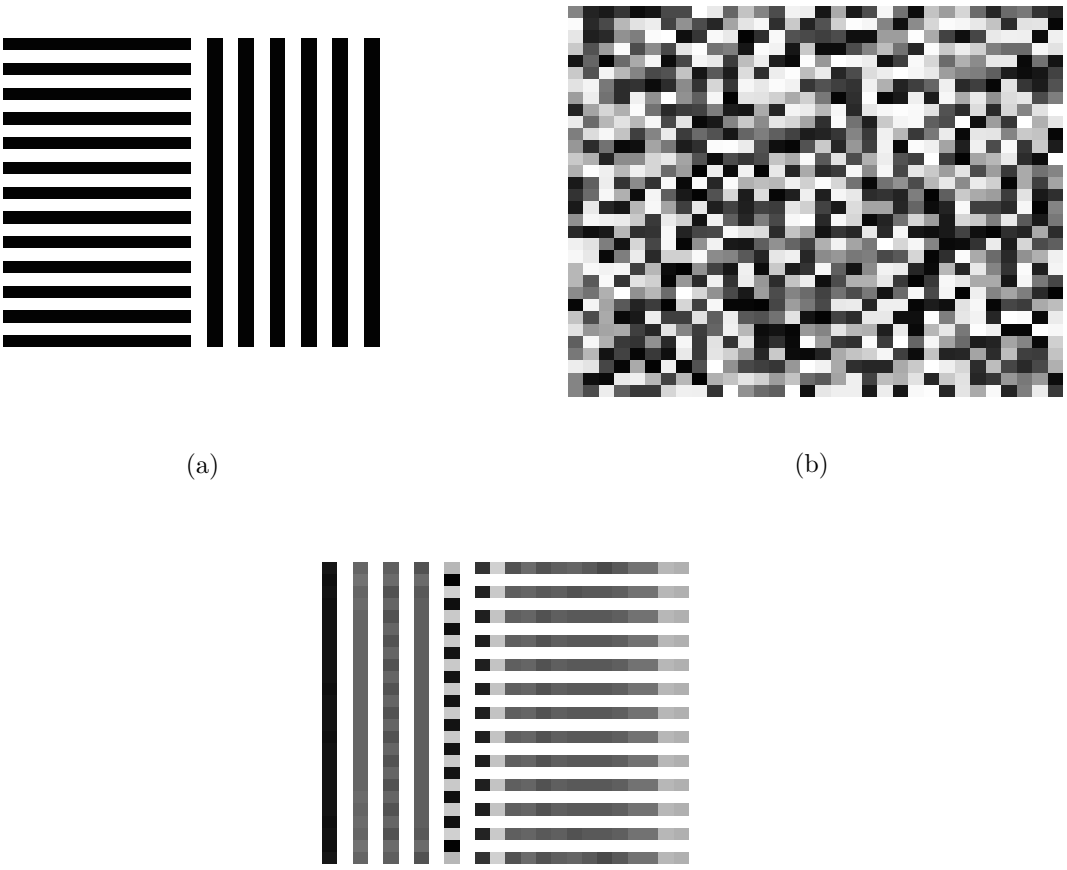


Fig. 7. (a) Original image. (b) Initial estimate. (c) Final estimate.

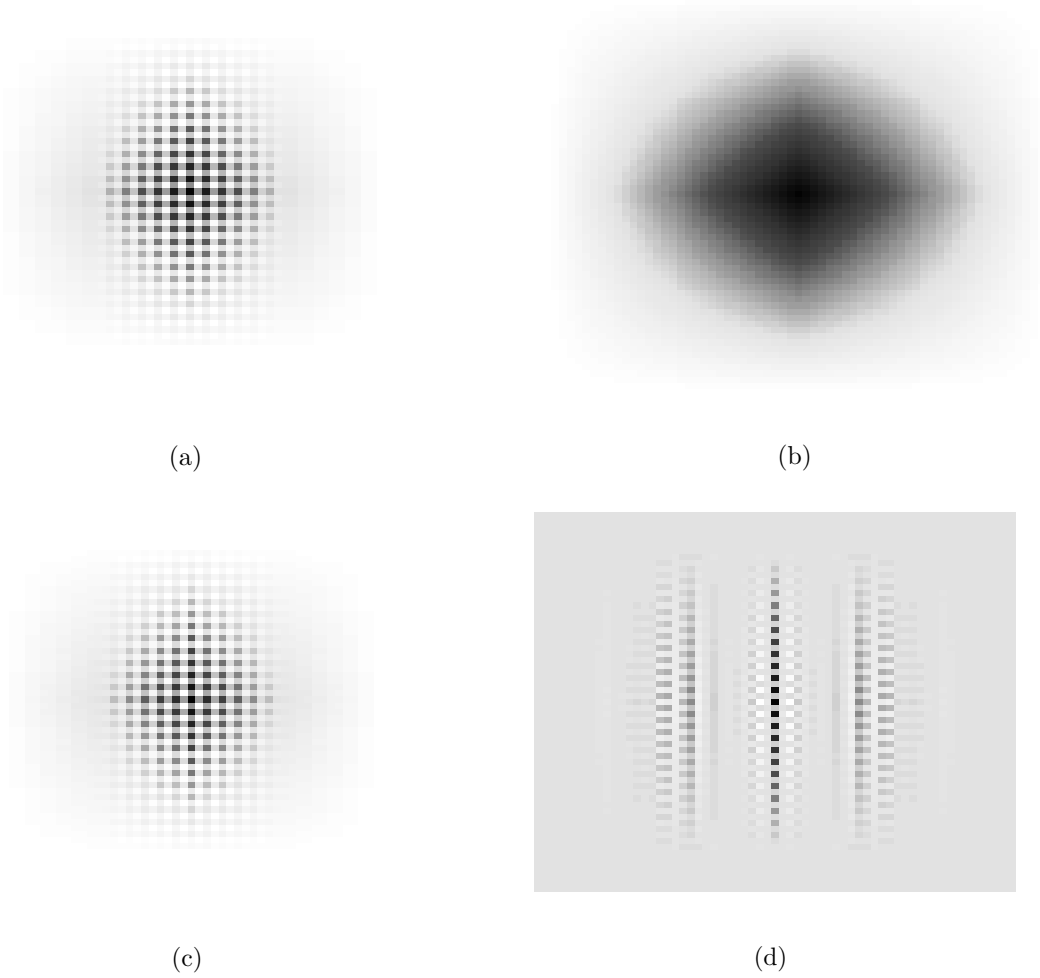
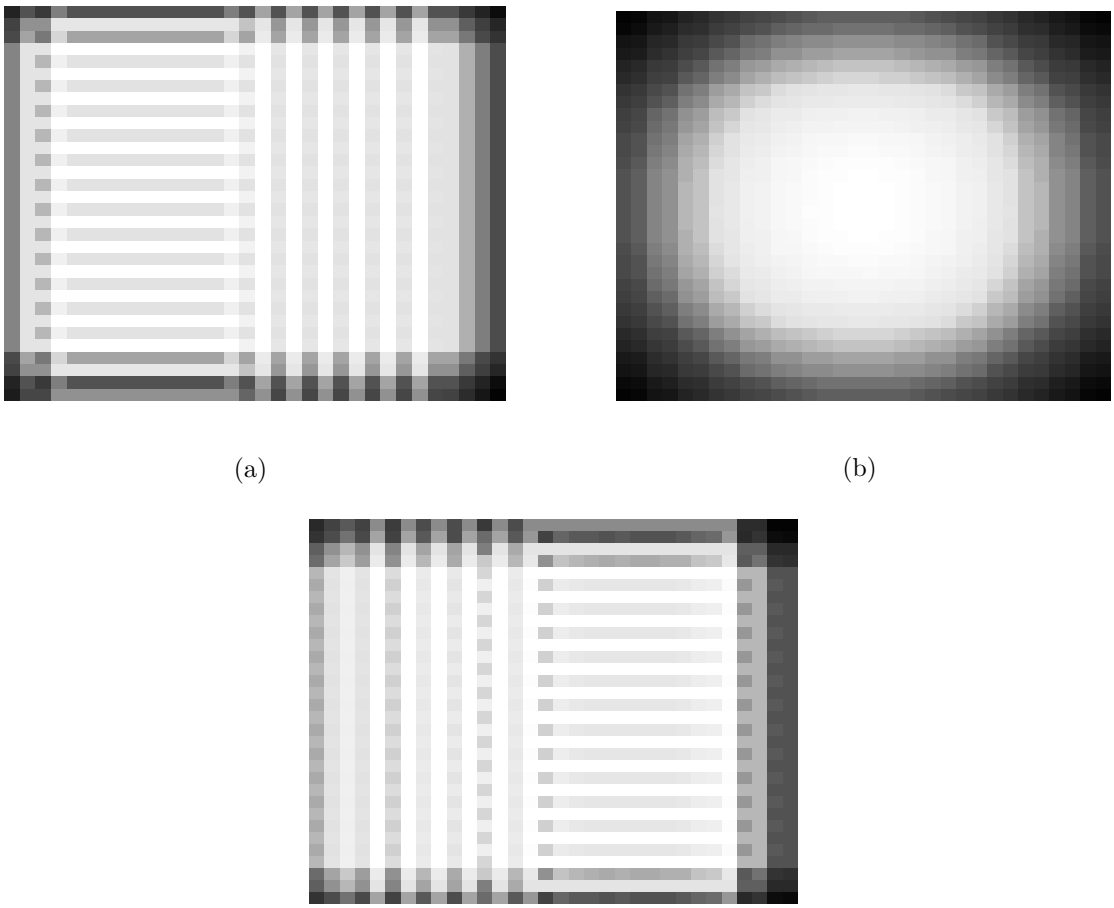


Fig. 8. Images associated with Fig. 7. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-7.56 2.19].



(a) (b)

(c)

Fig. 9. Images associated with Fig. 7. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.

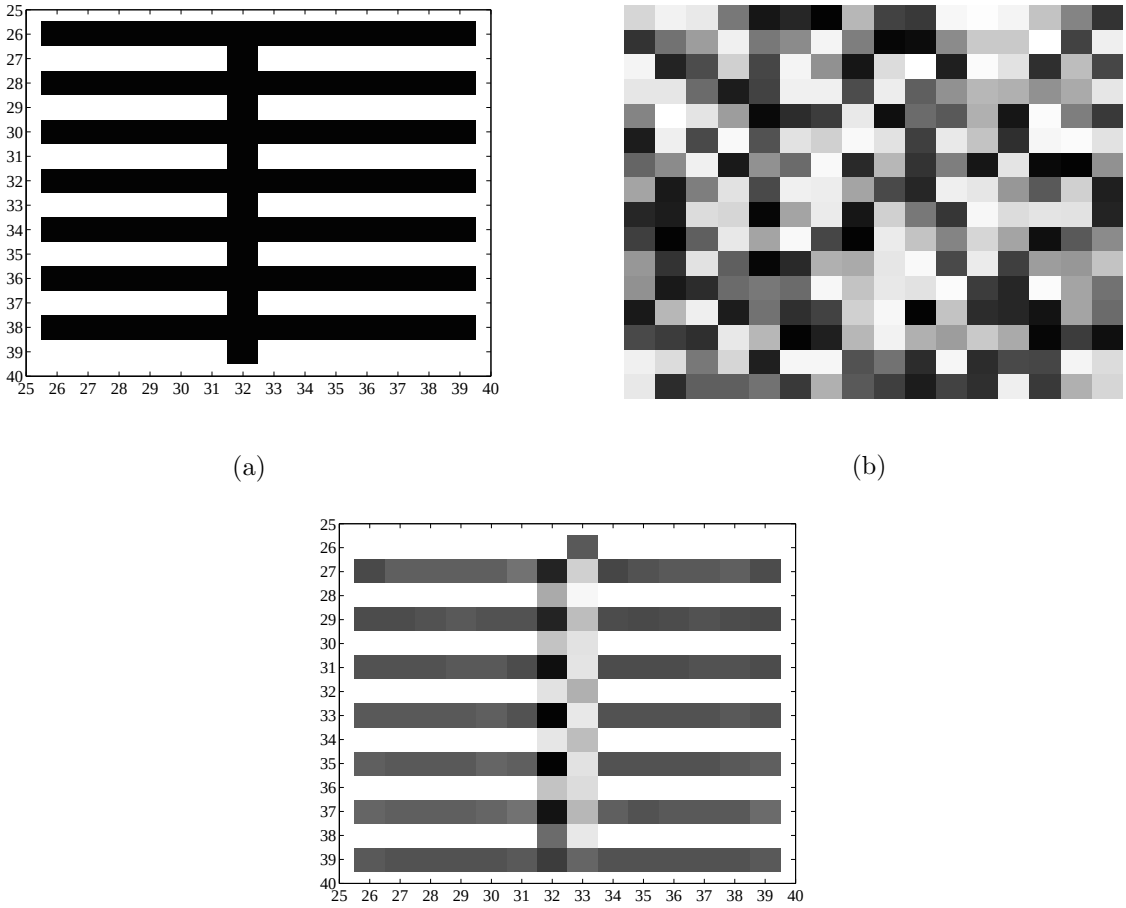
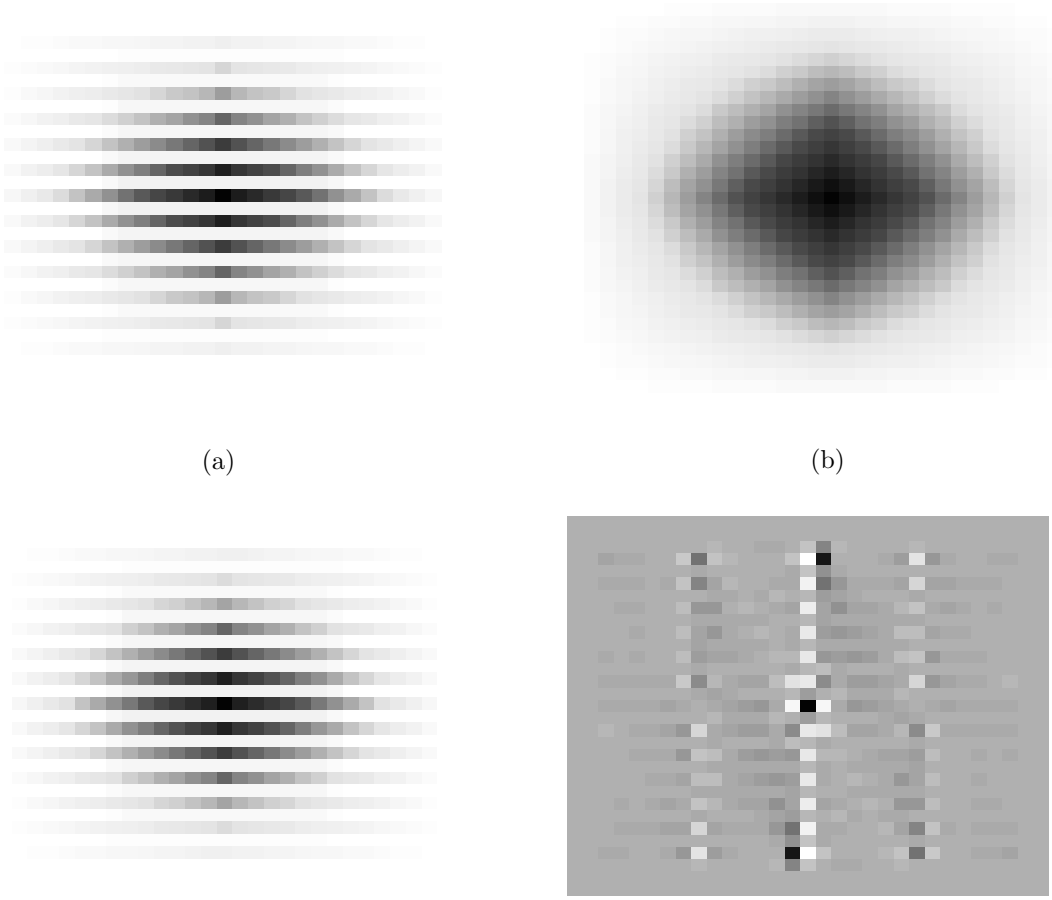


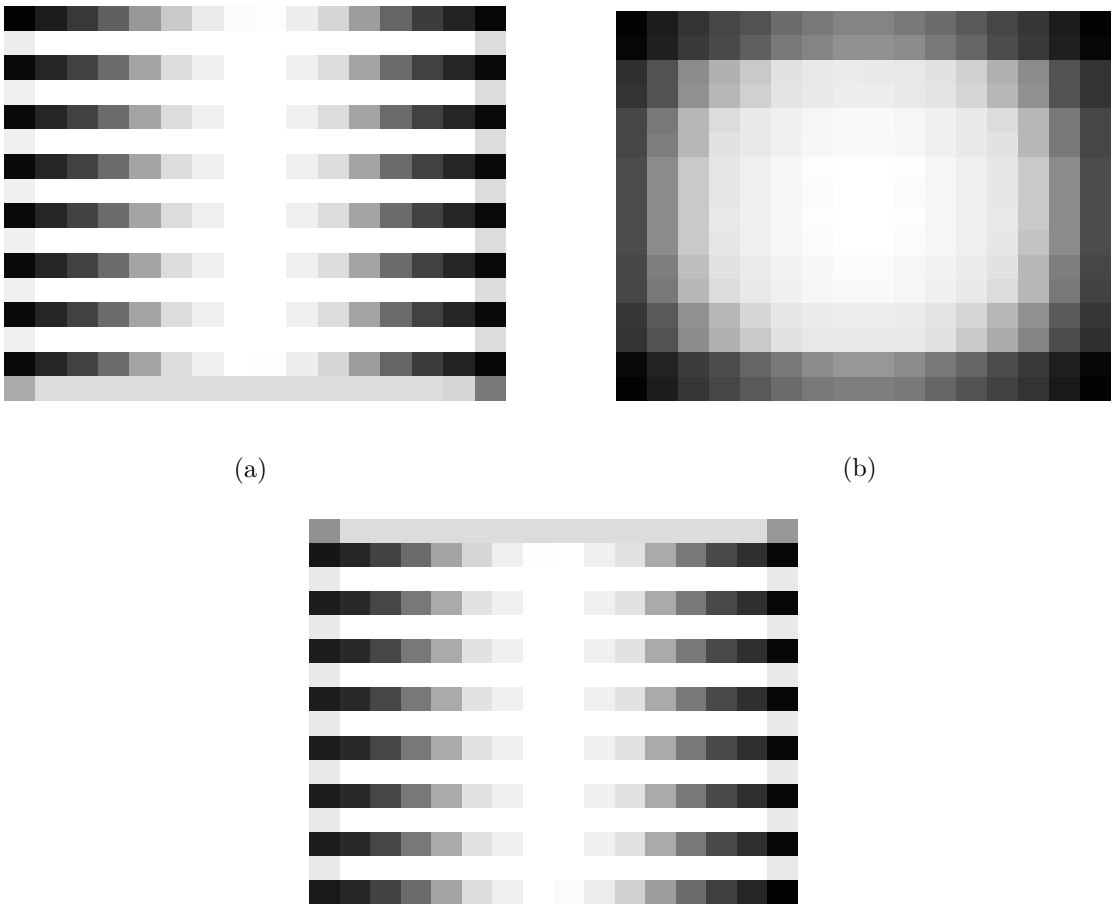
Fig. 10. (a) Original image. (b) Initial estimate. (c) Final estimate.



(a) (b)

(c) (d)

Fig. 11. Images associated with Fig. 10. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-0.58 0.91].



(a) (b)

(c)

Fig. 12. Images associated with Fig. 10. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.

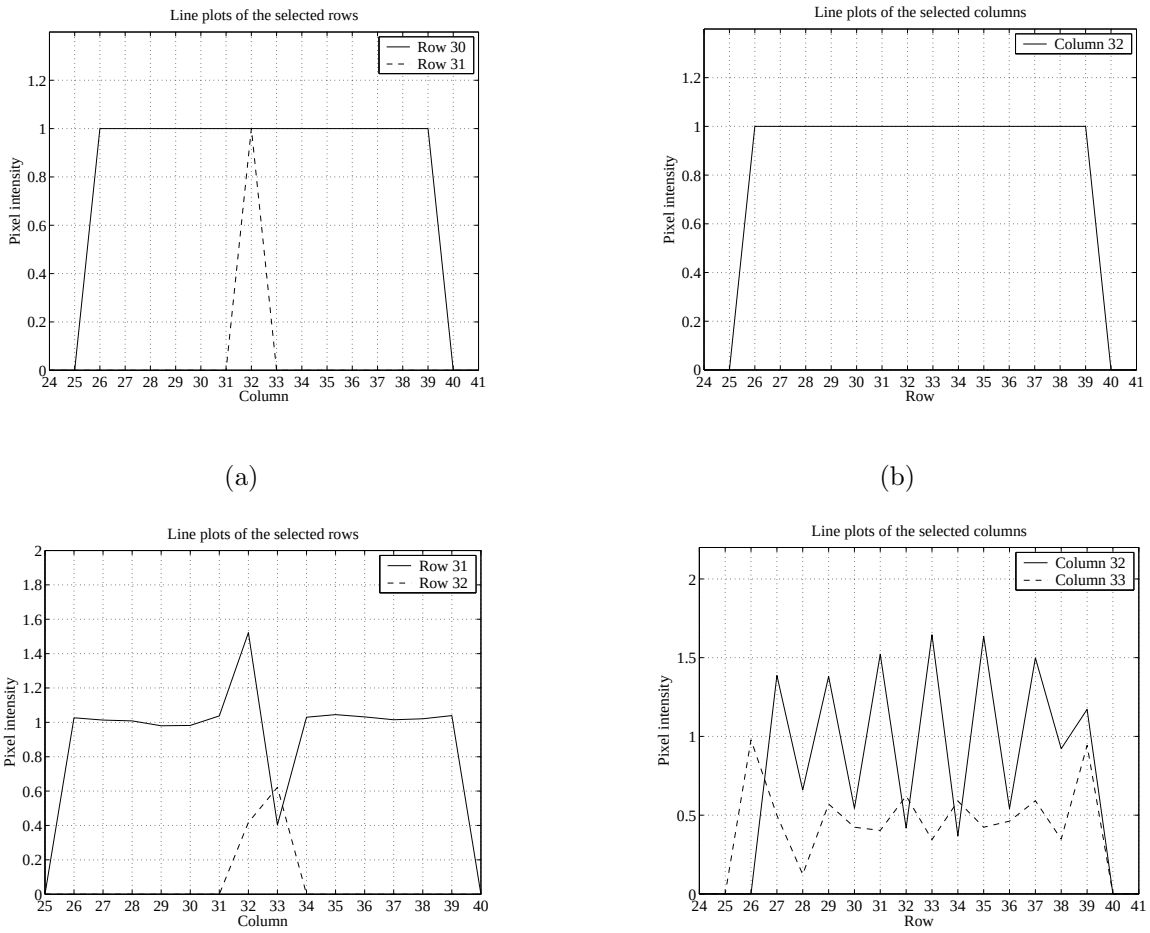


Fig. 13. Line plots associated with Fig. 10 are shown. The rows and columns are selected such that all other lines are common in the overall trend with one of the rows or columns. (a) Line plots of some selective rows of the original image. (b) Line plots of some selective columns of the original image. (c) Line plots of some selective rows of the final estimate. (d) Line plots of some selective columns of the final estimate.

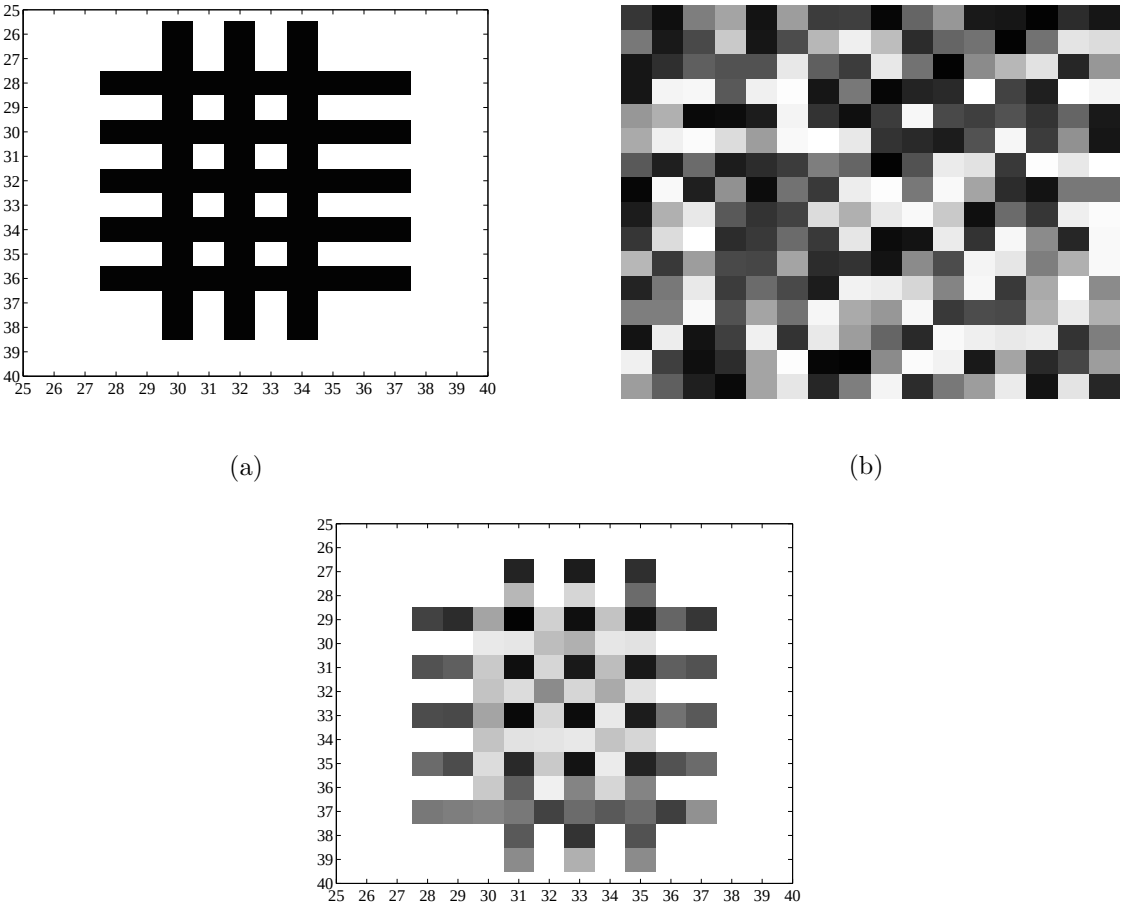


Fig. 14. (a) Original image. (b) Initial estimate. (c) Final estimate.

C. Mild Artifacts

The experiments described in the rest of this section were originally performed to try to analyze the resolution of the Schulz-Snyder algorithm. As our research progressed, it became apparent that resolution in the usual sense was not necessarily an issue (at least in the noiseless cases considered in this paper); points and lines are often reconstructed without any apparent blurring. However, two adjacent lines may some-

times appear blurred together. Sometimes, artifacts in such cases may have apparent tendencies as in the experiments to be discussed in Section D. Nonetheless, these mild artifacts, in most cases, are quite unpredictable.

1. Experiment 4

Figure 10(a) shows a vertical line superimposed on seven alternating horizontal lines. Figure 10(c) shows the corresponding final estimate. The estimate shows a somewhat weird reconstruction; the vertical line is spread to the left, except for the protruding part. Figure 13 shows line plots of some selected rows and columns. As we may see in Figs. 13(c) and 13(d), the ways that the vertical lines are reconstructed are quite different for the two different columns, and Row 31 is also reconstructed strangely. On the other hand, the gradients and autocorrelations of the original image and final estimate look very similar as shown in Figs. 11 and 12. As before, the sufficient conditions were tested on the final estimate to see if it has converged. Related data are given in Table 8.

2. Experiment 5

The unpredictability of such phenomena becomes more obvious with three vertical lines spaced by one pixel superimposed on five horizontal lines. Figure 14(a) shows the original true image, and Fig. 14(c) shows the final estimate of the original image. The reason that we remove a few horizontal lines is to emphasize the effects of the vertical line reconstruction more than the horizontal line reconstruction. Note that, in the final estimate, we obtain a 6×6 block that looks like a checkerboard. However, pixel values in the block are somewhat randomly distributed. Figure 17 supports this

Table 8. Selected data from Exp. 4.

Quantity	Value
$D[S, S]$	6.2433×10^{-13}
$D[S, R_{f_0}]$	1581.9542
$D[S, R_{f_k}]$	0.1572
k	57720
$\max_i \lvert f_k(x_i) - f_{k-1}(x_i)\rvert$	9.9997×10^{-8}
size(H)	111×111
max{eigenvalues(H)}	469.1720
min{eigenvalues(H)}	0.0574

observation. In this case, the corresponding autocorrelations still look very similar, but the gradients show a small difference; compare the top and bottom rows of Figs 16(a) and 16(c). The autocorrelations for this experiment are shown in Fig. 15. Table 9 shows the related data illustrating that the final estimate satisfies the sufficient conditions and hence is a local minimum.

3. Experiment 6

Experiment 6 illustrates a local minima which bears an overall resemblance to the correct answer, but with some unusual minor artifacts. Figure 18(a) shows the original true image consisting of three vertical lines spaced by three pixels superimposed on five alternating lines. Figure 18(c) is the final estimate of the original pattern. By

Table 9. Selected data from Exp. 5.

Quantity	Value
$D[S, S]$	5.900×10^{-13}
$D[S, R_{f_0}]$	1645.0375
$D[S, R_{f_k}]$	1.0190
k	15108
$\max_i \lvert f_k(x_i) - f_{k-1}(x_i)\rvert$	9.9979×10^{-8}
size(H)	86×86
max{eigenvalues(H)}	353.1094
min{eigenvalues(H)}	0.1541

carefully observing Row 35, we are able to see the vertical lines are spread among more than two bins. Figures 21(c) and 21(d) show line plots of some interesting rows and columns in addition to Row 35. The line plots shown in the figures are fairly different than the line plots in Figs. 17(c) and 17(d). Besides, the columns in Fig. 21(d) have a less predictable pattern than seen in, for instance, Fig. 25(d).

Interestingly, the corresponding gradients look quite similar, unlike in Exp. 5. Also, the corresponding autocorrelations look quite similar, as expected, and are shown in Fig. 19. As before, Table 10 shows related data from Exp. 6 concerning the testing of the sufficient conditions.

Table 10. Selected data from Exp. 6.

Quantity	Value
$D[S, S]$	5.0976×10^{-13}
$D[S, R_{f_0}]$	1982.5427
$D[S, R_{f_k}]$	1.2542
k	101158
$\max_i \lvert f_k(x_i) - f_{k-1}(x_i)\rvert$	9.9999×10^{-9}
size(H)	110×110
max{eigenvalues(H)}	454.0213
min{eigenvalues(H)}	0.0543

D. Straddle Loss Effects

Unlike the experiments in the latter part of Section C, the experiments discussed in this section show consistent behaviors.

1. Experiment 7

Figure 22(a) shows the original image of a vertical line superimposed on seven horizontal lines, but without any protruding parts as in Fig. 10(a). Figure 22(c) shows the final estimate of the original image. As can be seen, all the alternating rows (*e.g.*, Rows 26 and 28, or Rows 27 and 29) show similar tendencies, even though the values are fluctuating about 1 in the reconstructed horizontal lines. The two reconstructed columns also show a similar tendency. In contrast to the reconstructions of

the columns in Exps. 4, 5, and 6, the columns reconstructed in this experiment have a somewhat clear pattern (compare Column 33 in Figs. 13(d) and 25(d)). Note that the reconstructions of the vertical lines are spread between only two bins. In this experiment, both the autocorrelations and gradients of the original image and final estimate look extremely close (see Figs 23 and 24). The data showing that the final estimate is a local minimum are given in Table 11.

Table 11. Selected data from Exp. 7.

Quantity	Value
$D[S, S]$	8.5242×10^{-13}
$D[S, R_{f_0}]$	3716.3495
$D[S, R_{f_k}]$	0.0807
k	94240
$\max_i \lvert f_k(x_i) - f_{k-1}(x_i)\rvert$	9.9999×10^{-8}
size(H)	110×110
max{eigenvalues(H)}	474.7070
min{eigenvalues(H)}	0.0032

We believe the spreading is due to a *straddle loss* effect such as that seen when one applies an FFT to a sinusoid whose frequency does not fall exactly on one of the FFT bins, and the energy from that sinusoid is split between two adjacent bins (see, for instance, Ref. [16, p. 165]).

2. Experiment 8

We performed another experiment with a pattern of two vertical lines superimposed on seven horizontal lines. The original image is shown in Fig. 26(a), and the final estimate is shown in Fig. 26(c). Table 12 supports the fact that the final estimate is a local minimum based on the given data. The reconstructions of the vertical lines are spread between only two bins as in Exp. 7, which is a clear straddle loss effect. This is obviously different than the effects in Exp. 6, whose original image is also composed of two vertical lines with some alternating horizontal lines. The line plots shown in Fig. 29 show consistent patterns and manifest the straddle loss effects along Row 30 (see Figs. 29(c) and 29(d)). The autocorrelations and gradients are shown in Figs. 27 and 28, respectively.

Recall that translating and/or mirroring an image does not change its autocorrelation function; hence, an algorithm trying to reconstruct an image from its autocorrelation may reconstruct a shifted version of the original image. In fact, the I-divergence surface has many global minima, some of which correspond to versions of the image shifted by an integer number of pixels. We suspect that in the above experiments, where a small amount of directional blurring is observed, the algorithm is trying to reconstruct the image translated a half-pixel to the right of its original location, resulting in the straddle loss effect as the algorithm tries to put the vertical line in between pixel values. The most surprising aspect of these experiments is that no straddle loss effects were observed for the repeated horizontal lines (and similarly for the cases with repeated vertical lines, which are not shown); we do not have an

Table 12. Selected data from Exp. 8.

Quantity	Value
$D[S, S]$	5.7178×10^{-13}
$D[S, R_{f_0}]$	2685.6983
$D[S, R_{f_k}]$	0.0292
k	257827
$\max_i \lvert f_k(x_i) - f_{k-1}(x_i)\rvert$	9.9999×10^{-9}
size(H)	122×122
max{eigenvalues(H)}	522.9897
min{eigenvalues(H)}	0.0022

explanation for this intriguing behavior at present.

3. Experiment 9

In a final experiment, we initialized the algorithm with the true image with a constant ϵ added to every pixel in a rectangle surrounding the true image, where the true image is taken to be the pattern in Fig. 26(a) from Experiment 8. The support of the constant rectangle is the same as the support for the initial uniform estimate used in the experiment that resulted in Fig. 26(c). Using a small value of $\epsilon = 10^{-6}$, the algorithm converges to the true answer without any blurring, in the sense that the vertical lines have value 1 and the valleys are practically zero. When ϵ is boosted to 0.5, the straddle loss effect is again observed; however, this time, the two vertical

lines are quite distinguishable. The value of two peaks in the estimated image is 0.8, and the value of two valleys is 0.2; this can be thought of as an answer lying between the correct result and the block result in Fig. 26(c). To save space, we do not show the plots for this experiment since they provide little information beyond the textual description and are largely redundant with the preceding plots.

5. Conclusions

We have derived the second derivatives of the I-divergence between a measured autocorrelation and an estimated autocorrelation in terms of the image estimate, and used that to illustrate that the Schulz-Snyder algorithm may converge to local minima of the I-divergence surface. In some cases, the local minima correspond to shifts of the true image. Sometimes this shift may consist of a fraction of a pixel width. Such local minima may be thought of as "close" to a true global minimum, and are not that objectionable. In other cases, the local minima may bear little resemblance to the true object.

The Schulz-Snyder algorithm bears a striking resemblance to the Richardson-Lucy[17] algorithm for reconstructing an image from a blurred version of that image. The Richardson-Lucy algorithm can be shown to minimize the I-divergence between the measured blurred image and a blurred version of the estimated image.[18] In the Richardson-Lucy case, one can show that the objective function has a unique global minimum.[18] We have shown that the Schulz-Snyder scenario is far more complex. Not only are the distinct global minima due to the invariance of the autocorrelation function to shift, but there may also be numerous local minima in which the algorithm

may become trapped. To our knowledge, this paper is the first to report these effects concerning the Schulz-Snyder algorithm.

Modifications to the Schulz-Snyder algorithm must be found which will allow it to break out of local minima. In our experiments, we had the advantage of knowing what the true images were; in practice, it may be difficult to tell if a found minimum is a local minimum or one of the true global minima. The Schulz-Snyder algorithm is promising, since even without these modifications, it has still been shown to converge in many situations. Given this promise, we hope this paper will inspire the phase retrieval community to explore this path.

We were drawn to the Schulz-Snyder iteration due to its analogy with multiplicative deblurring iterations that are popular within the optics community, such as the Richardson-Lucy iteration. As suggested by a reviewer, a useful avenue for future work would be to compare the Schulz-Snyder iteration with some other traditional optimization techniques such as Newton's methods, trust region methods, and conjugate gradient methods.

The authors may be reached by e-mail at

kerkil@ece.gatech.edu, lanterma@ece.gatech.edu

.

Appendix A: Proof of Theorem

Let f^* denote an estimate satisfying the Kuhn-Tucker conditions given in Eq. (8). For an arbitrary f that satisfies the nonnegativity constraint and is in the neighborhood

of f^*, we want to show that

$$J(f^* + \Delta f) - J(f^*) > 0. \tag{15}$$

To do so, define $\Delta f = f - f^*$ and start with a Taylor expansion formula:

$$\begin{aligned} J(f^* + \Delta f) &= J(f^*) + \sum_i \frac{\partial J(f^*)}{\partial f(x_i)} \Delta f(x_i) \\ &+ \frac{1}{2} \sum_i \sum_j \frac{\partial^2 D[S, R_{f^*}]}{\partial f(x_i) \partial f(x_j)} \Delta f(x_i) \Delta f(x_j) + o(\|\Delta f(x_i)\|^2), \end{aligned} \tag{16}$$

where $o(\cdot)$ is defined as in [19, p. 861]. We now move $D[S, R_{f^*}]$ to the left-hand side, and regroup the parameters on the right-hand side according to the sets $\mathcal{S}_1$ and $\mathcal{S}_2$. Then, we obtain

$$\begin{aligned} J(f^* + \Delta f) - J(f^*) &= \sum_{i \in \mathcal{S}_1} \frac{\partial J(f^*)}{\partial f(x_i)} \Delta f(x_i) + \sum_{i \in \mathcal{S}_2} \frac{\partial J(f^*)}{\partial f(x_i)} \Delta f(x_i) \\ &+ \frac{1}{2} \sum_{i \in \mathcal{S}_1} \sum_{j \in \mathcal{S}_1} \frac{\partial^2 J(f^*)}{\partial f(x_i) \partial f(x_j)} \Delta f(x_i) \Delta f(x_j) \\ &+ \frac{1}{2} \sum_{i \in \mathcal{S}_1} \sum_{j \in \mathcal{S}_2} \frac{\partial^2 J(f^*)}{\partial f(x_i) \partial f(x_j)} \Delta f(x_i) \Delta f(x_j) \\ &+ \frac{1}{2} \sum_{i \in \mathcal{S}_2} \sum_{j \in \mathcal{S}_1} \frac{\partial^2 J(f^*)}{\partial f(x_i) \partial f(x_j)} \Delta f(x_i) \Delta f(x_j) \\ &+ \frac{1}{2} \sum_{i \in \mathcal{S}_2} \sum_{j \in \mathcal{S}_2} \frac{\partial^2 J(f^*)}{\partial f(x_i) \partial f(x_j)} \Delta f(x_i) \Delta f(x_j) \\ &+ o(\|\Delta f(x_i)\|^2). \end{aligned} \tag{17}$$

The index set $\mathcal{S}_2$ may be empty. However, we assume $\mathcal{S}_2$ is non-empty for generality. The proof can be easily modified for the case of empty $\mathcal{S}_2$, though. Now, by the definition of the index sets, the first term on the right-hand side is zero, and the second term is positive because all the parameters in the set $\mathcal{S}_2$ are on the boundary, and hence the $\Delta f(x_i)$ are always positive. We now inspect the fourth through the

sixth terms on the right-hand side. Since we have only two $\Delta f(x_i)$ terms, we can always relate the two terms in a trivially linear way. In other words, for any $\Delta f(x_i)$ and $\Delta f(x_j)$ such that $i \in \mathcal{S}_1$ and $j \in \mathcal{S}_2$, there always exists a constant c_{ij} such that $\Delta f(x_i) = c_{ij}\Delta f(x_j)$. Using this relation, we conclude that

$$\begin{aligned} \frac{1}{2}\sum_{i\in\mathcal{S}_1}\sum_{j\in\mathcal{S}_2}\frac{\partial^2 J(f^*)}{\partial f(x_i)\partial f(x_j)}\Delta f(x_i)\Delta f(x_j) &= \frac{1}{2}\sum_{i\in\mathcal{S}_1}\sum_{j\in\mathcal{S}_2}\frac{\partial^2 J(f^*)}{\partial f(x_i)\partial f(x_j)}c_{ij}[\Delta f(x_j)]^2 \\ &= o(\|\Delta f(x_i)\|). \end{aligned} \tag{18}$$

Therefore, the fourth term is dominated by the second term on the right-hand side of Eq. (17) for sufficiently small $||\Delta f(x_i)||$. Hence, we can neglect the fourth term. The same reasoning easily proves that the fifth and sixth terms are also negligible. As we have seen so far, all the terms on the right-hand side except for the third term are either positive or negligible. In the meantime, the second condition in Eq. (9) guarantees positivity of the third term. Therefore, we finally obtain the inequality in Eq. (15). This proves the theorem.

Appendix B: Derivation of The Second Partial Derivative

We start with the first derivative:

$$
\begin{aligned}
\frac{\partial^2 J(f)}{\partial f(x_i)\partial f(x_j)} &= \frac{\partial}{\partial f(x_j)}\left[2\sum_x f(x) - \sum_y \{f(x_i+y)+f(x_i-y)\}\frac{S(y)}{R_f(y)}\right] \\
&= 2 - \sum_y \frac{\partial}{\partial f(x_i)}\{f(x_i+y)+f(x_i-y)\}\frac{S(y)}{R_f(y)} \\
&= 2 - \sum_y S(y)\Bigg\{\frac{f'(x_i+y)R_f(y) - f(x_i+y)R'_f(y)}{R_f^2(y)} \\
&\quad + \frac{f'(x_i-y)R_f(y) - f(x_i-y)R'_f(y)}{R_f^2(y)}\Bigg\} \\
&= 2 - \sum_y S(y)\Bigg\{\frac{\delta(x_i+y-x_j)}{R_f(y)} - \frac{f(x_i+y)\{f(x_j+y)+f(x_j-y)\}}{R_f^2(y)} \\
&\quad + \frac{\delta(x_i-y-x_j)}{R_f(y)} - \frac{f(x_i-y)\{f(x_j+y)+f(x_j-y)\}}{R_f^2(y)}\Bigg\} \\
&= 2 + \sum_y \{f(x_i+y)f(x_j+y)+f(x_i+y)f(x_j-y)\}\frac{S(y)}{R_f(y)} \\
&\quad + \sum_y \{f(x_i-y)f(x_j+y)+f(x_i-y)f(x_j-y)\}\frac{S(y)}{R_f(y)} \\
&\quad - \left(\frac{S(x_j-x_i)}{R_f(x_j-x_i)} + \frac{S(x_i-x_j)}{R_f(x_i-x_j)}\right) \\
&= 2 + \sum_y \{f(x_i+y)+f(x_i-y)\}\{f(x_j+y)+f(x_j-y)\}\frac{S(y)}{R_f(y)} \\
&\quad - \left(\frac{2S(x_j-x_i)}{R_f(x_j-x_i)}\right). \qquad (19)
\end{aligned}
$$

The last equality holds because an autocorrelation is symmetric.

Acknowledgment

We would like to thank Dr. Timothy Schulz of Michigan Technological University for reviewing this research. Also, we would like to thank the two reviewers for various helpful comments on this work, which led to a much more precise paper.

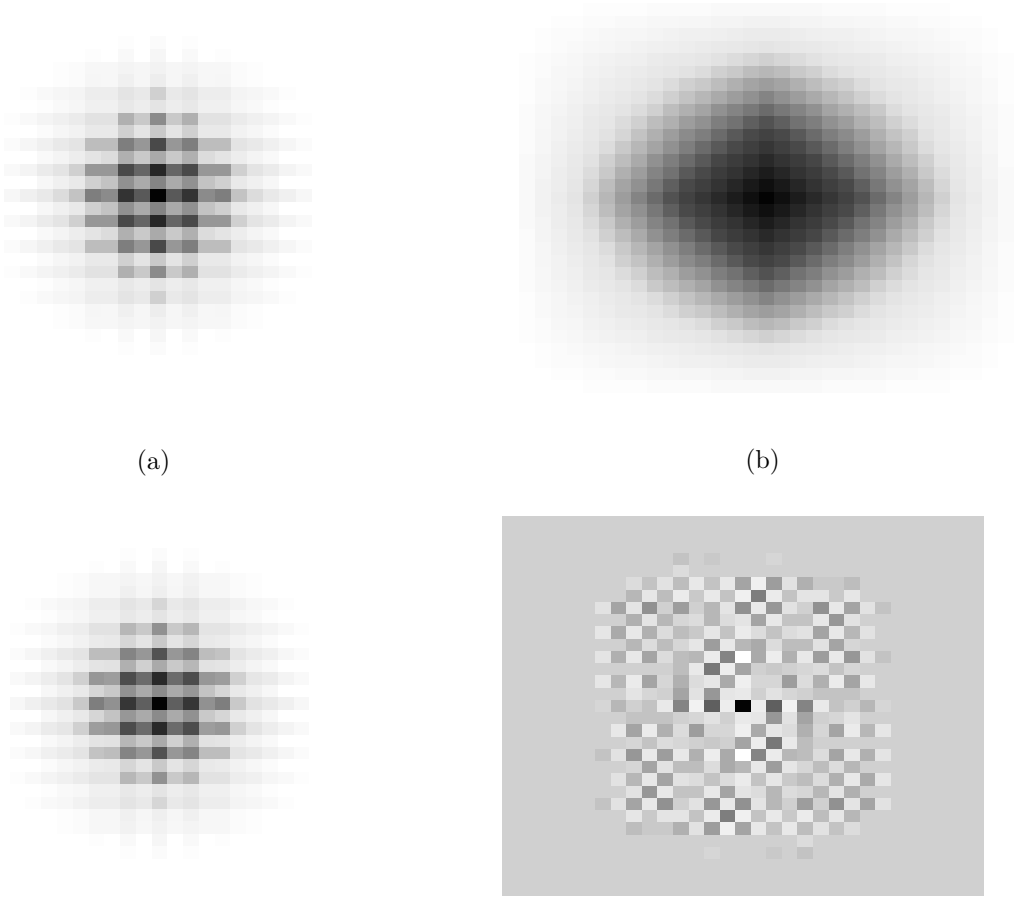
References

1. J. R. Fienup, "Reconstruction of an object from the modulus of its Fourier transform," Opt. Lett. **3,** 27-29 (1978).
2. J. R. Fienup, "Space object imaging through the turbulent atmosphere," Opt. Eng. **18,** 529-534 (1979).
3. J. R. Fienup, "Phase retrieval algorithms: a comparison," Appl. Opt. **21,** 2758-2769 (1982).
4. G. H. Stout and L. H. Jenson, *X-Ray Structure Determination* (Macmillan, London, 1968).
5. J. C. Dainty, J. R. Fienup, "Phase retrieval and image reconstruction for astronomy," *Image Recovery: Theory and Applications*, H. Stark, ed. (Academic, New York, 1987).
6. C. Y. C. Liu and A. W. Lohmann, "High resolution image formation through the turbulent atmosphere ," Opt. Commun. **8,** 372-377 (1973).
7. P. J. Napier and R. H. T. Bates, "Inferring phase information from modulus information in two-dimensional aperture synthesis," Astron. Astrophys. Suppl. **15,** 427-430 (1974).
8. B. R. Frieden and D. G. Currie, "On unfolding the autocorrelation function," J. Opt. Soc. Am. **66**, 1111(A) (1976).
9. R. H. T. Bates, "Fourier phase problems are uniquely solvable in more than one dimension. I: underlying theory," Optik **61**, 247-262 (1982); K. L. Garden and R. H. T. Bates, "II: One-dimensional considerations," Optik **62**, 131-142 (1982); W.

R. Fright and R. H. T. Bates, "III: Computational examples for two dimension," Optik **62**, 219-230 (1982).

10. A. Levi and H. Stark, "Image restoration by the method of generalized projections with application to restoration from magnitude," J. Opt. Soc. Am. A **1**, 932-943 (1984).
11. T. J. Schulz and D. L. Snyder, "Imaging a randomly moving object from quantum-limited data: applications to image recovery from second- and third-order auto-correlations," J. Opt. Soc. Am. A **8**, 801-807 (1991).
12. T. J. Schulz and D. L. Snyder, "Image recovery from correlations," J. Opt. Soc. Am. A **9**, 1266–1272 (1992).
13. J. R. Fienup and C. C. Wackerman, "Phase-retrieval stagnation problems and solutions," J. Opt. Soc. Am. A **3**, 1897-1907(1986).
14. I. Csiszár, "Why least squares and maximum entropy? — an axiomatic approach to inverse problems," Ann. Stat. **19**, 2033–2066 (1991).
15. S. Kullback, *Information Theory and Statistics* (Wiley, New York, 1959).
16. G. Morris and L. Harkness, *Airborne Pulsed Doppler Radar* (Artech House, Boston, 1996).
17. L. Lucy, "An Iterative Technique for the Rectification of Observed Distributions," The Astronomical Journal **79**, 745–754, (1974).
18. D. L. Snyder, T.J. Schulz, and J.A. O'Sullivan, "Deblurring subject to nonnegativity constraints," IEEE Trans. Signal Process. **40**, 1143–1150, (1992).
19. T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for signal*

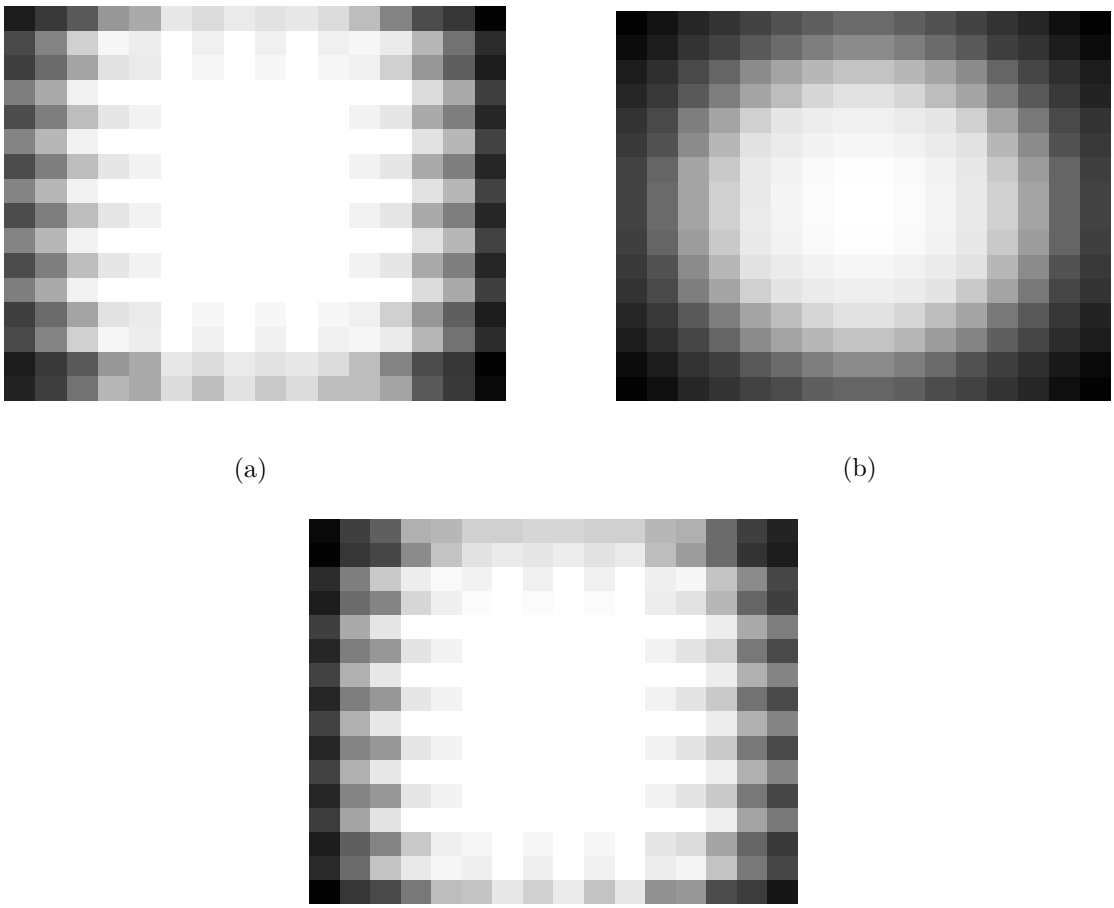
processing (Prentice Hall, New Jersey, 1999).



(a) (b)

(c) (d)

Fig. 15. Images associated with Fig. 14. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-0.80 1.80].



(a) (b)

(c)

Fig. 16. Images associated with Fig. 14. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.

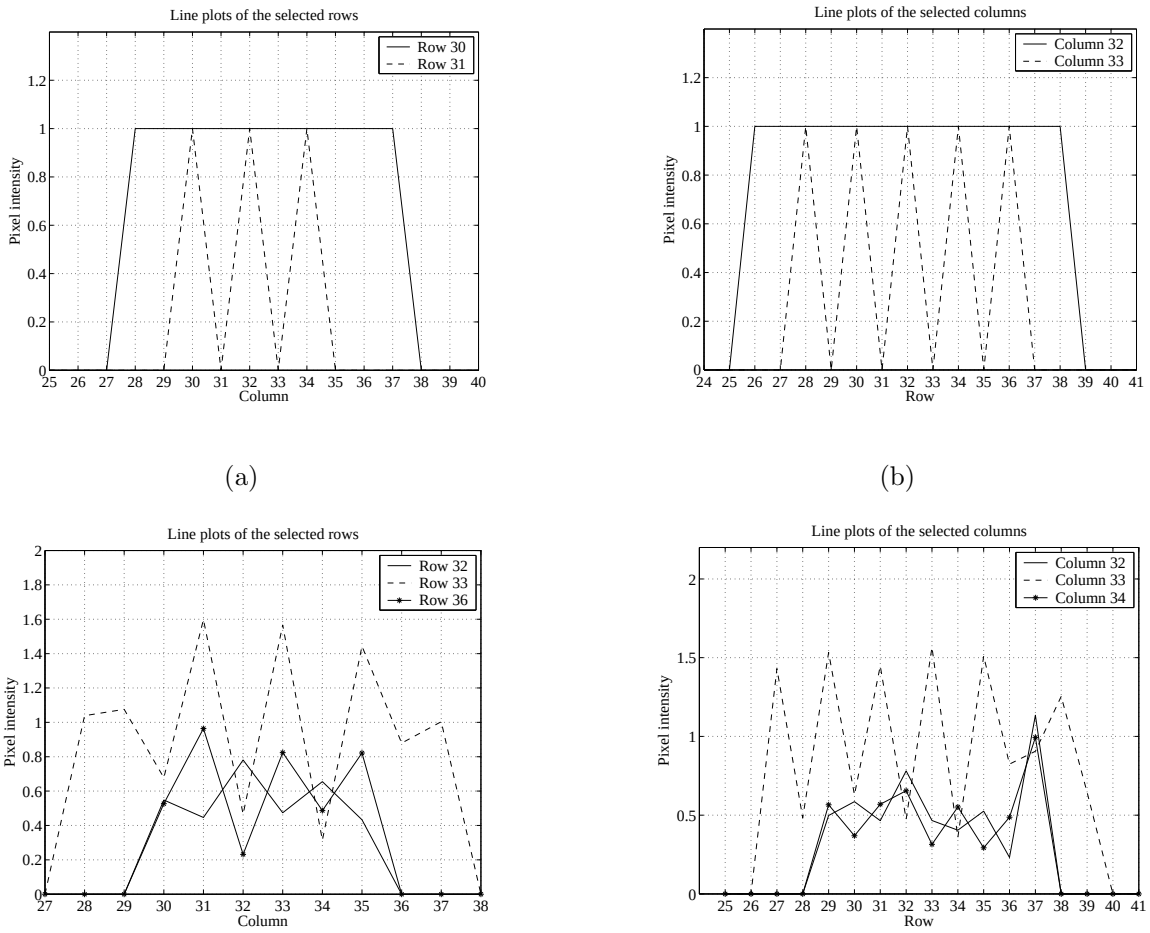
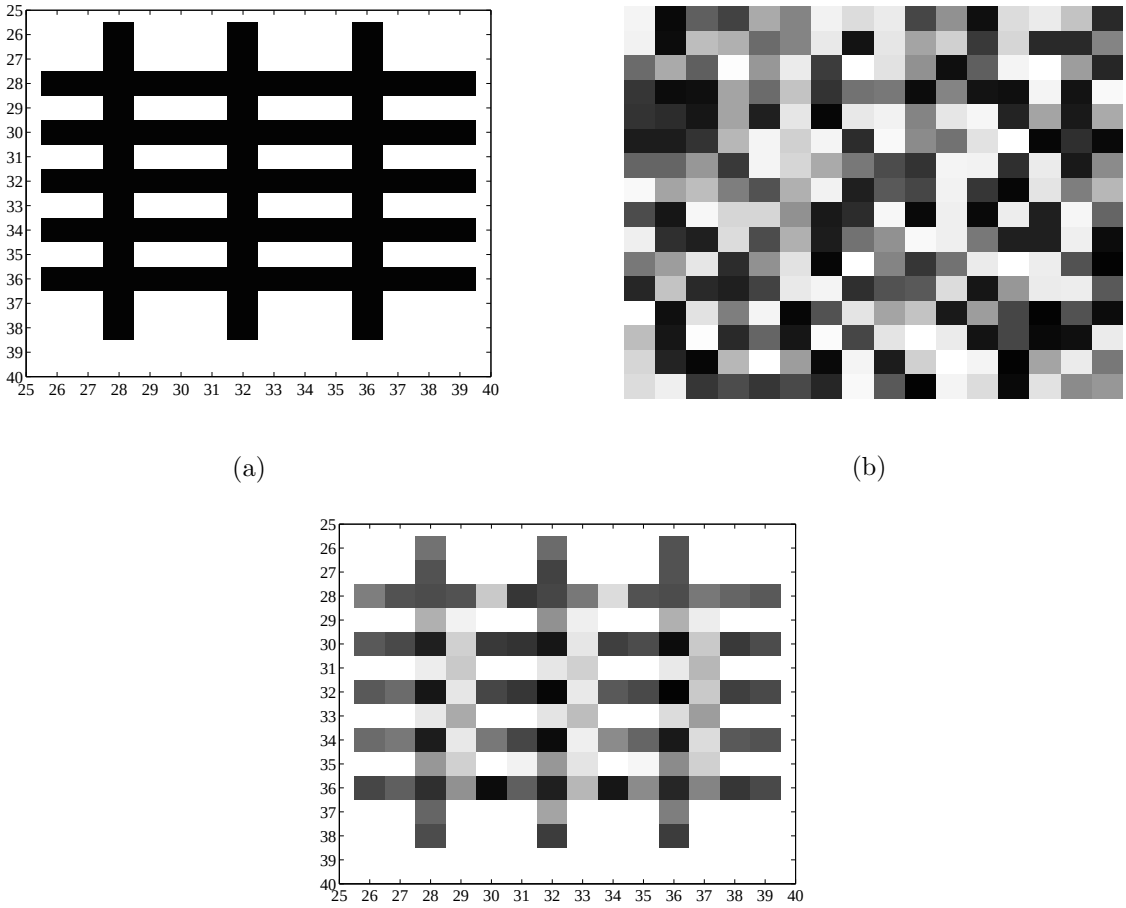


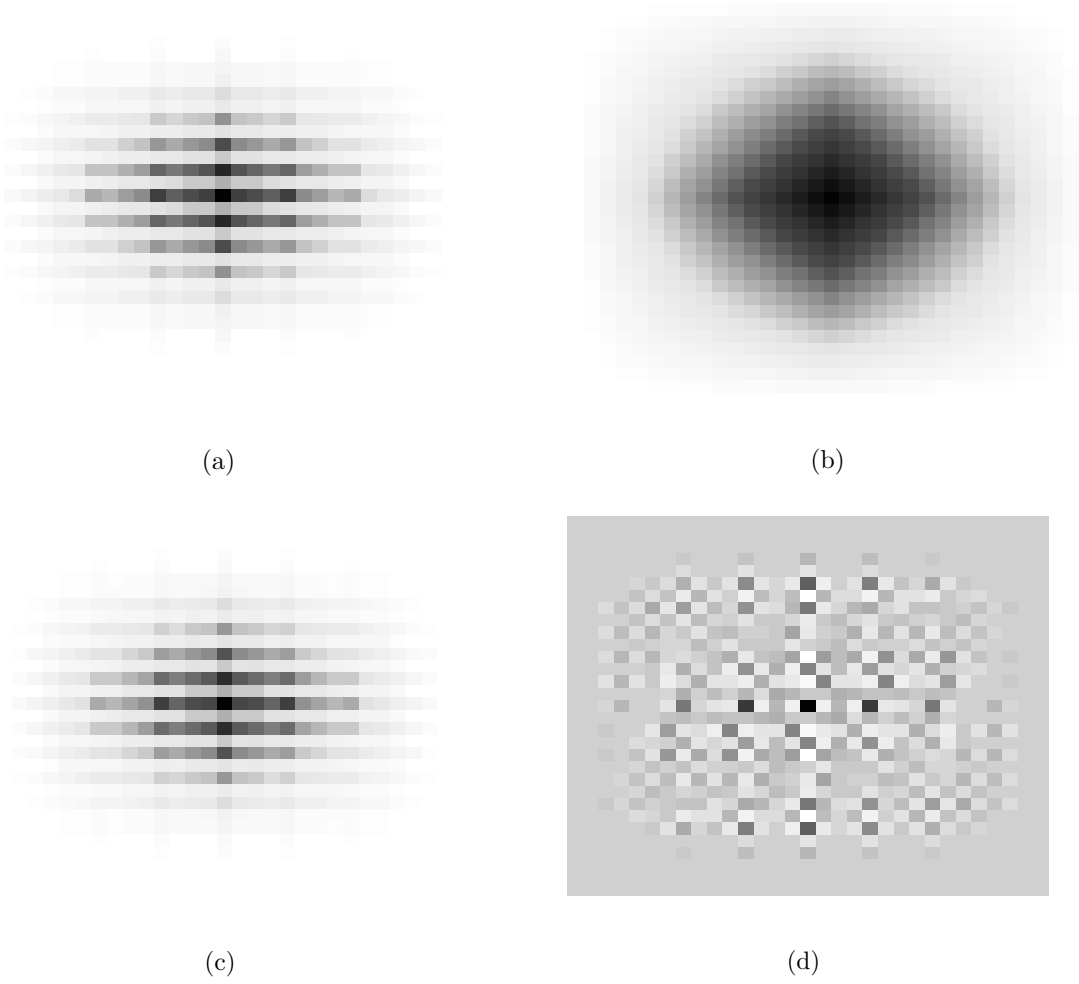
Fig. 17. Line plots associated with Fig. 14 are shown. The rows and columns are selected such that all other lines are common in the overall trend with one of the rows or columns. (a) Line plots of some selected rows of the original image. (b) Line plots of some selected columns of the original image. (c) Line plots of some selected rows of the final estimate. (d) Line plots of some selected columns of the final estimate.



(a) (b)

(c)

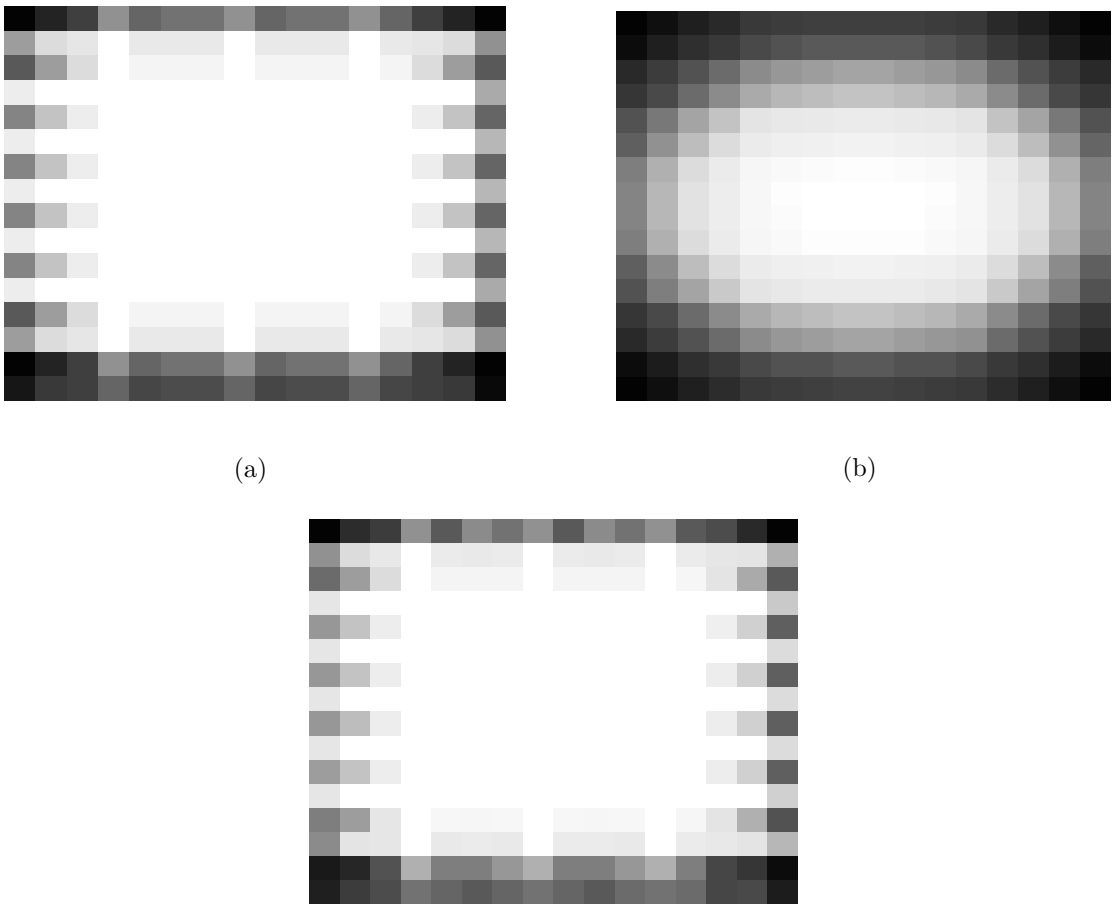
Fig. 18. (a) Original image. (b) Initial estimate. (c) Final estimate.



(a) (b)

(c) (d)

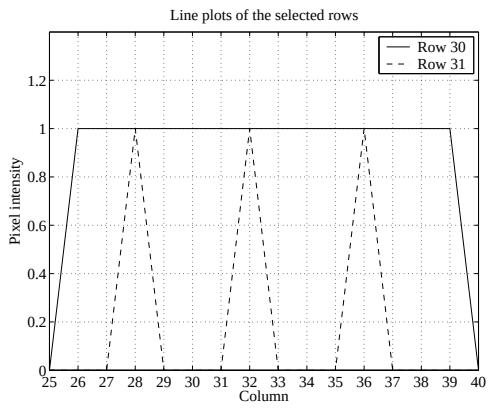
Fig. 19. Images associated with Fig. 18. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-0.93 2.10].



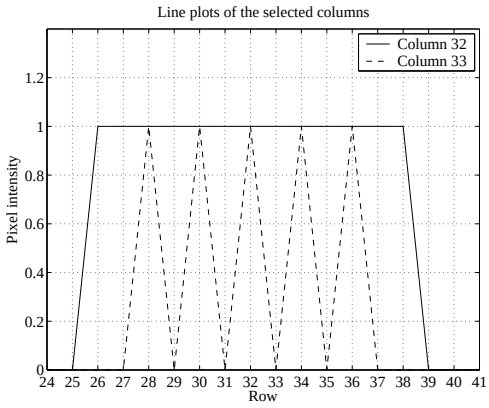
(a) (b)

(c)

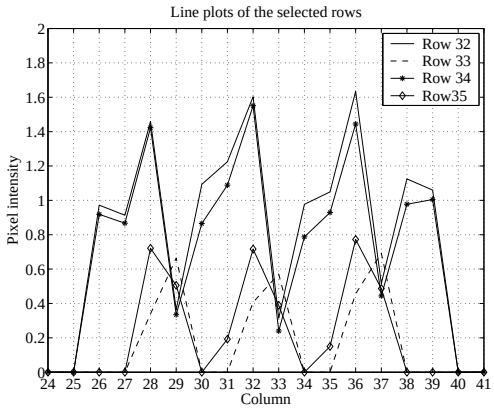
Fig. 20. Images associated with Fig. 18. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.



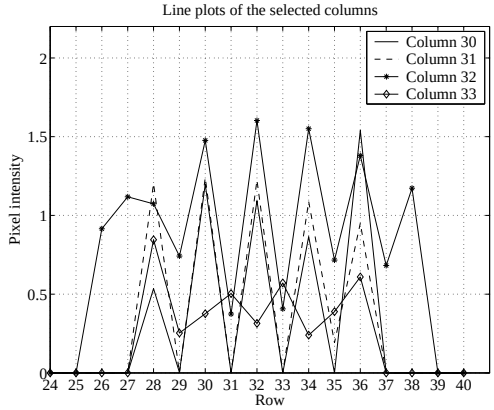
(a)



(b)



(c)



(d)

Fig. 21. Line plots associated with Fig. 18 are shown. The rows and columns are selected such that all other lines are common, in the overall trend, with one of the rows or columns. (a) Line plots of some selected rows of the original image. (b) Line plots of some selected columns of the original image. (c) Line plots of some selected rows of the final estimate. (d) Line plots of some selected columns of the final estimate.

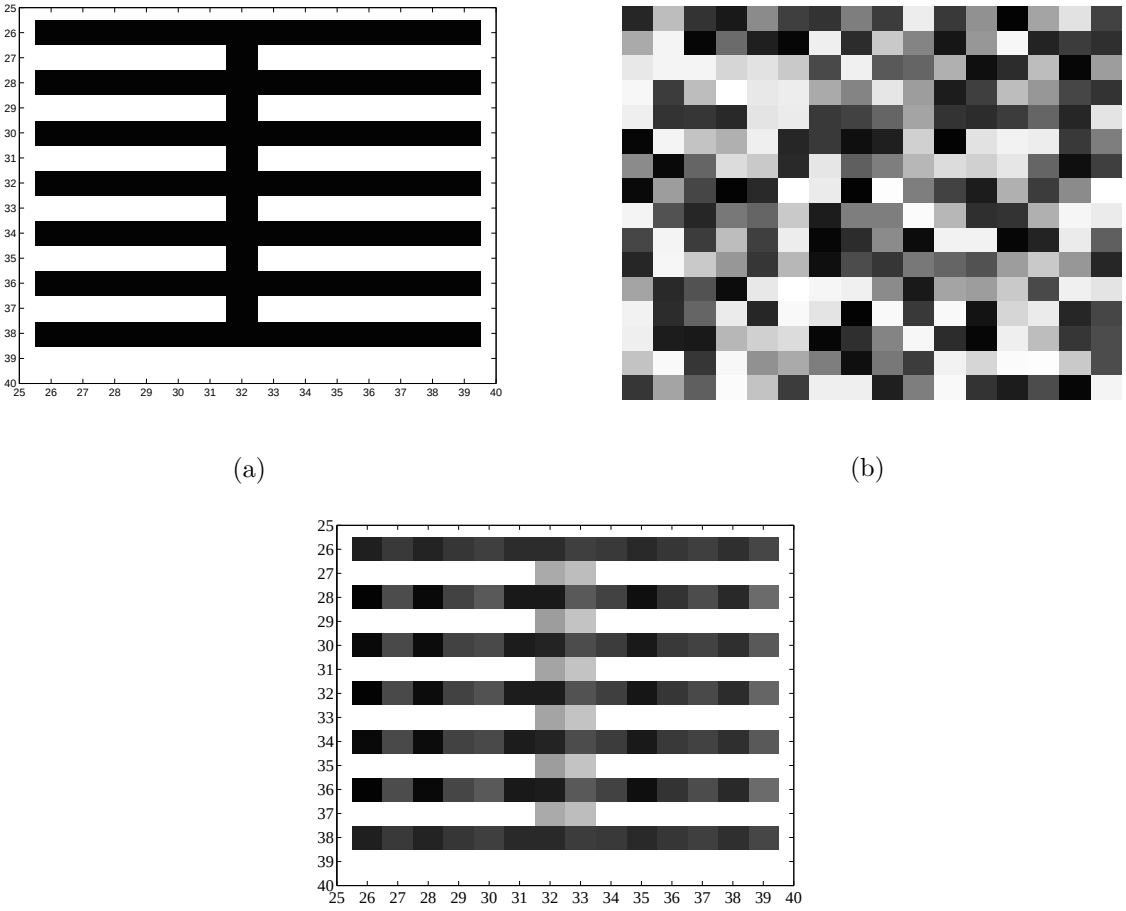
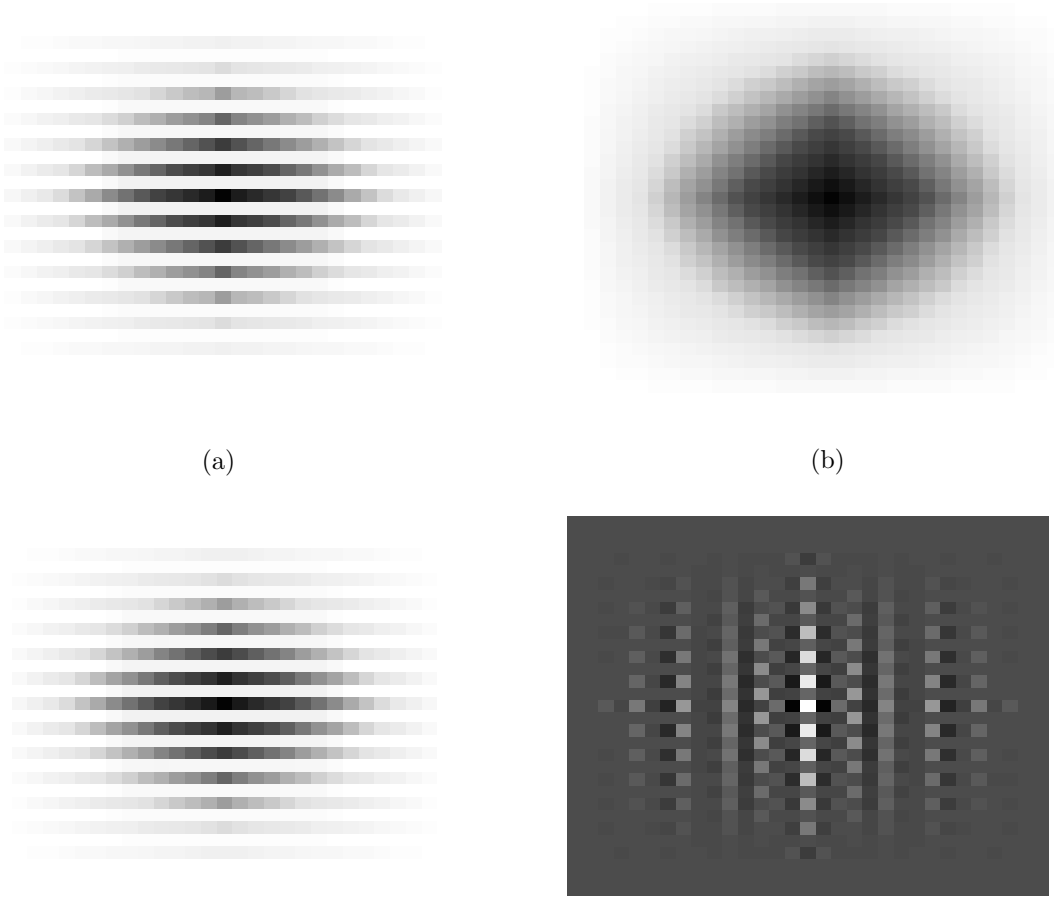


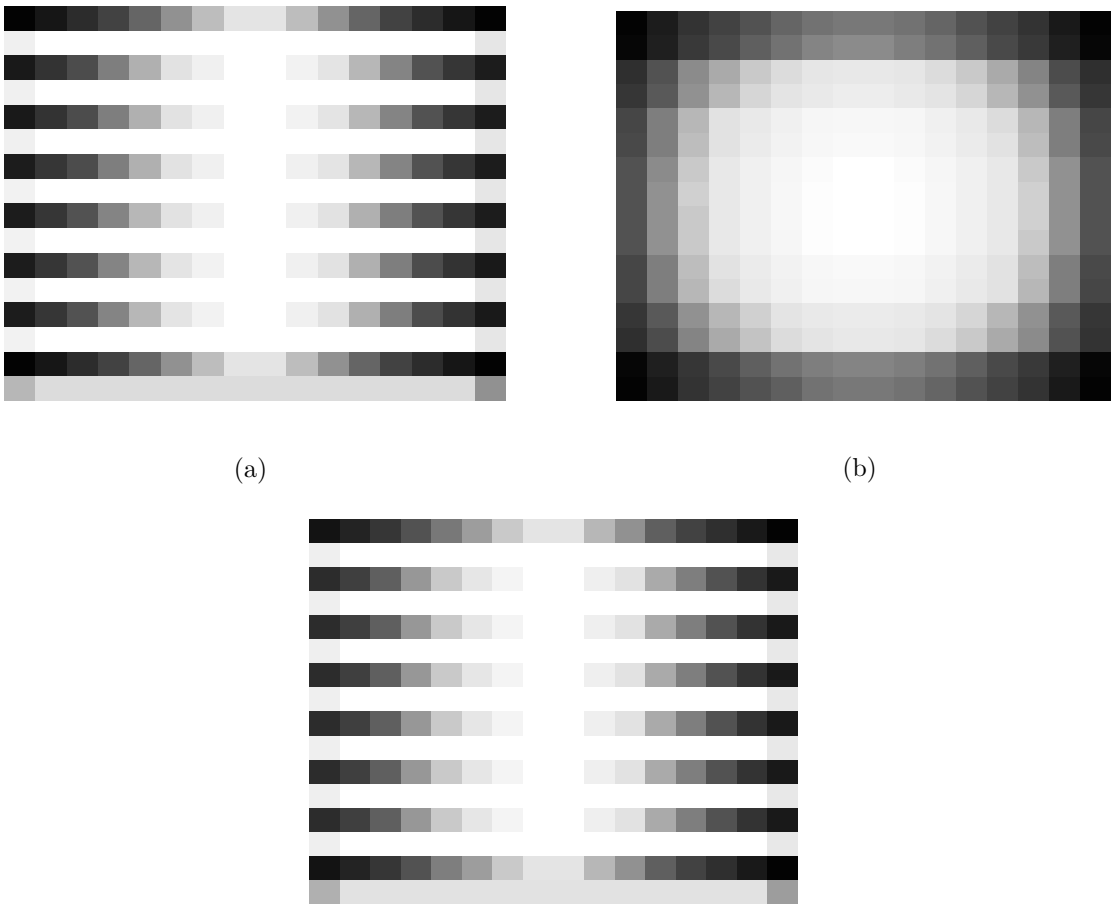
Fig. 22. (a) Original image. (b) Initial estimate. (c) Final estimate.



(a) (b)

(c) (d)

Fig. 23. Images associated with Fig. 22. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-0.70 0.39].



(a) (b)

(c)

Fig. 24. Images associated with Fig. 22. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.

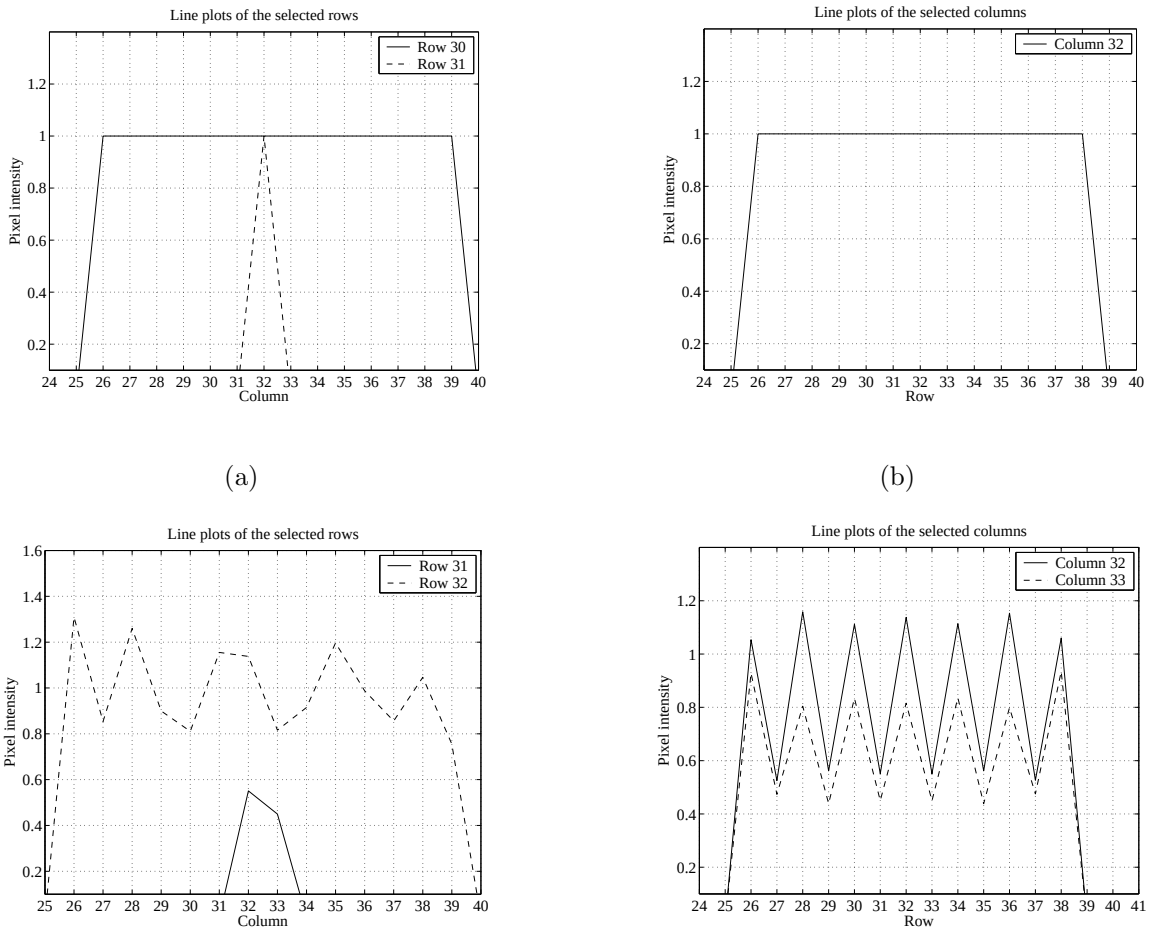


Fig. 25. Line plots associated with Fig. 22 are shown. The rows and columns are selected such that all other lines are common in the overall trend with one of the rows or columns. (a) Line plots of some selective rows of the original image. (b) Line plots of some selective columns of the original image. (c) Line plots of some selective rows of the final estimate. (d) Line plots of some selective columns of the final estimate.

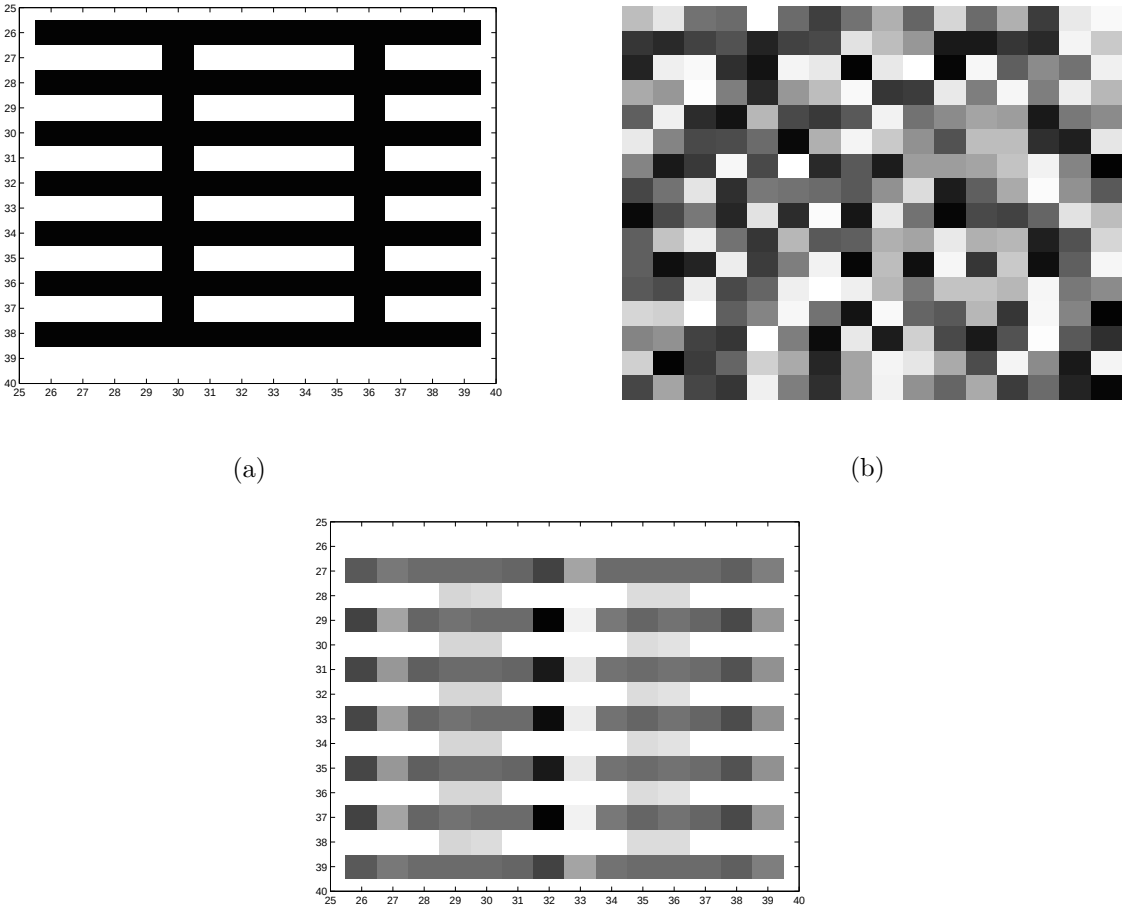
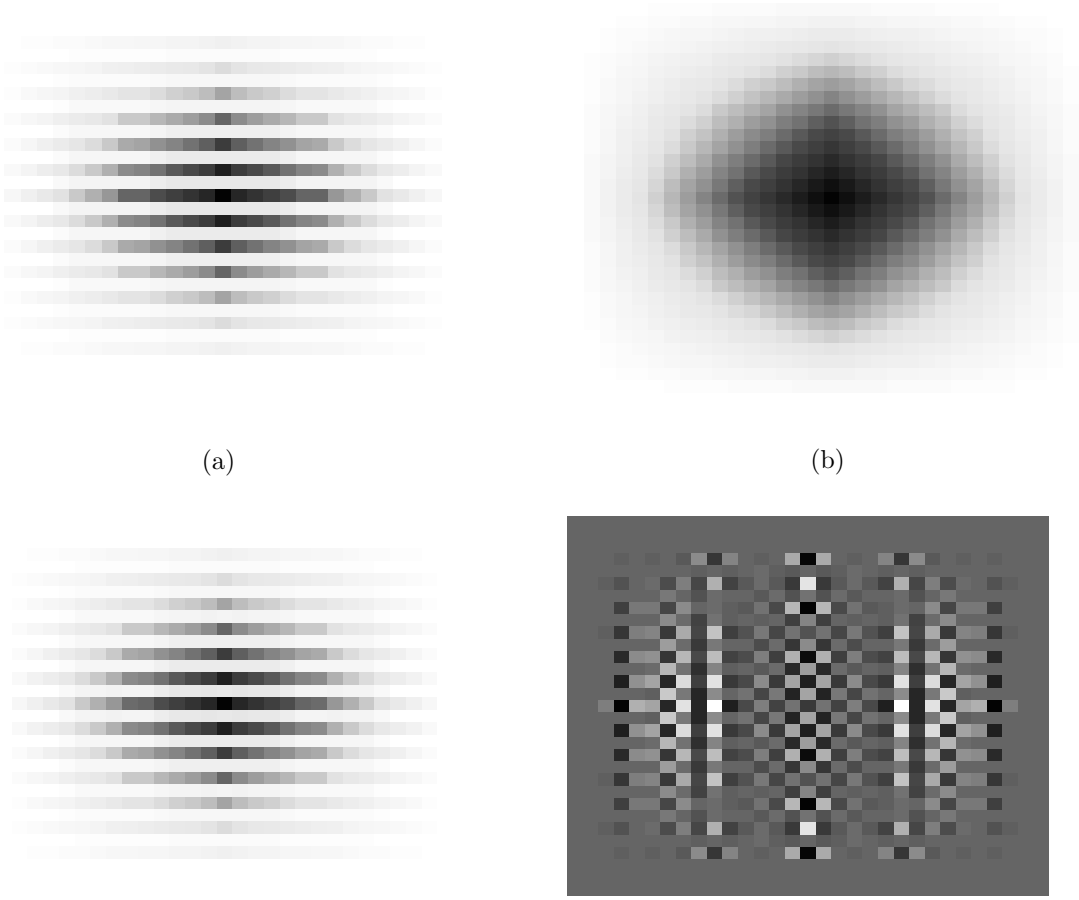


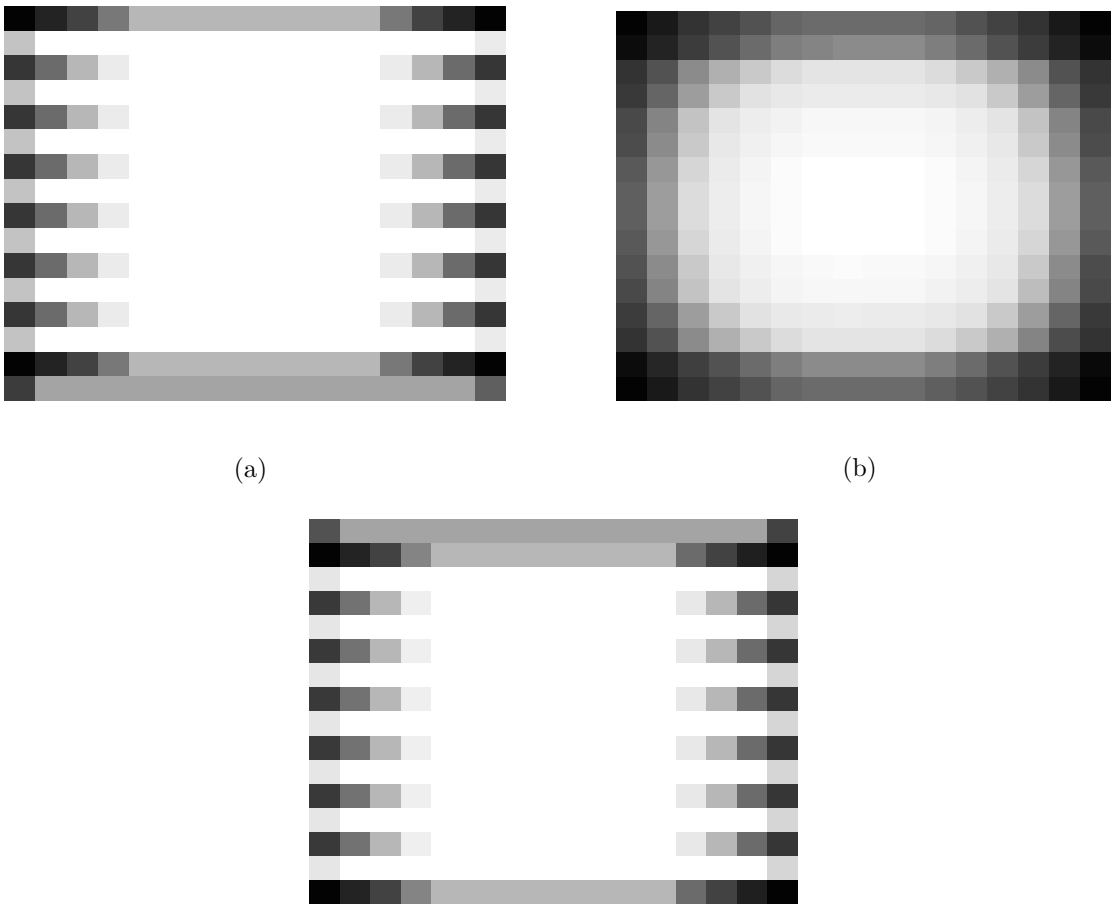
Fig. 26. (a) Original image. (b) Initial estimate. (c) Final estimate.



(a) (b)

(c) (d)

Fig. 27. Images associated with Fig. 26. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-0.20 0.20].



(a) (b)

(c)

Fig. 28. Images associated with Fig. 26. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.

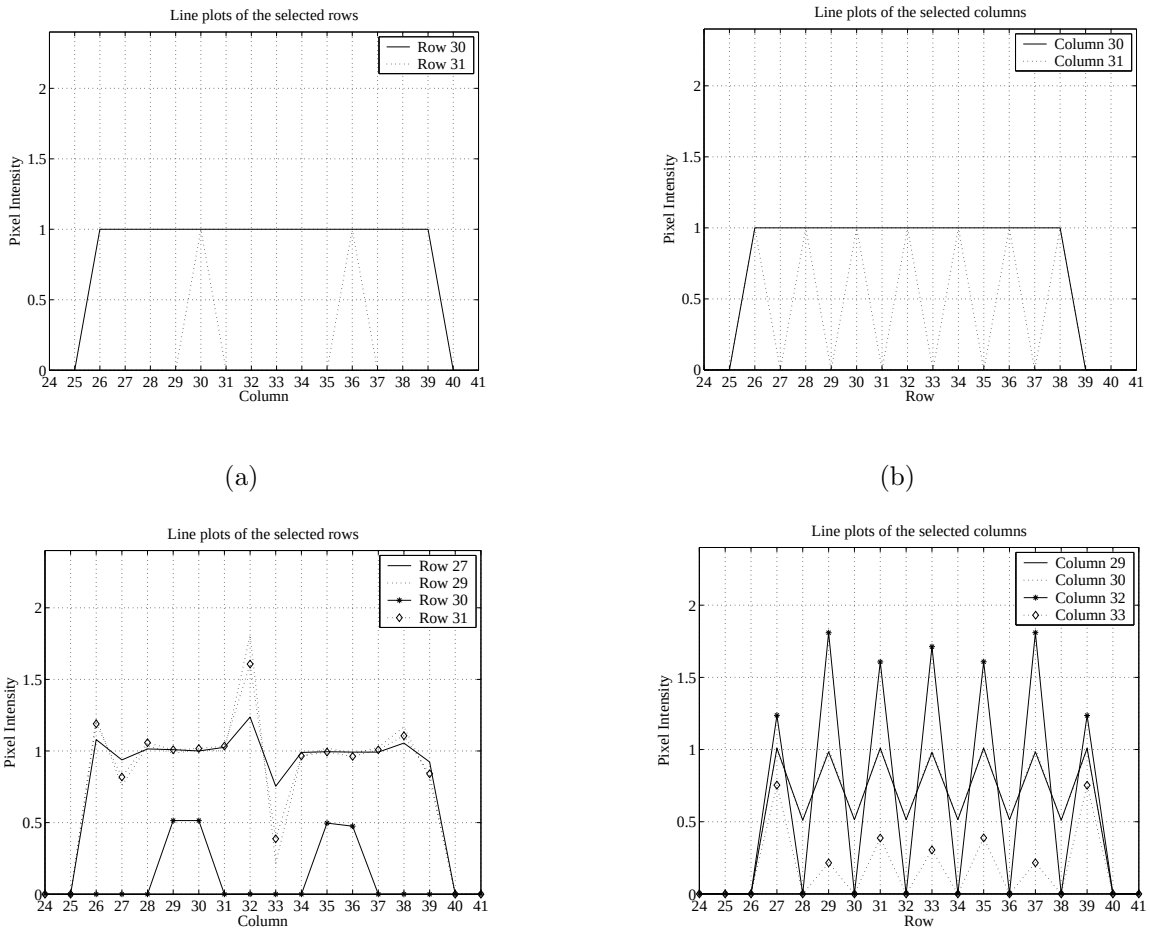


Fig. 29. Line plots associated with Fig. 26 are shown. The rows and columns are selected such that all other lines are common, in the overall trend, with one of the rows or columns. (a) Line plots of some selected rows of the original image. (b) Line plots of some selected columns of the original image. (c) Line plots of some selected rows of the final estimate. (d) Line plots of some selected columns of the final estimate.