Numerical investigation of the uniqueness of phase retrieval

J. H. Seldin and J. R. Fienup

*Optical Science Laboratory, Advanced Concepts Division, Environmental Research Institute of Michigan,
P.O. Box 8618, Ann Arbor, Michigan 48107*

Both a new iterative grid-search technique and the iterative Fourier-transform algorithm are used to illuminate the relationships among the ambiguous images nearest a given object, error metric minima, and stagnation points of phase-retrieval algorithms. Analytic expressions for the subspace of ambiguous solutions to the phase-retrieval problem are derived for 2 × 2 and 3 × 2 objects. Monte Carlo digital experiments using a reduced-gradient search of these subspaces are used to estimate the probability that the worst-case nearest ambiguous image to a given object has a Fourier modulus error of less than a prescribed amount. Probability distributions for nearest ambiguities are estimated for different object-domain constraints.

1. INTRODUCTION

The phase-retrieval problem considered in this paper is the reconstruction of an object function $f(x, y)$ from the modulus $|F(u, v)|$ of its Fourier transform:

$$F(u, v) = |F(u, v)|\exp[i\psi(u, v)] = \mathcal{F}[f(x, y)]$$

$$= \int\int f(x, y)\exp[-i2\pi(ux + vy)]dxdy. \quad (1)$$

It is equivalent to the reconstruction of the Fourier phase $\psi(u, v)$ from the Fourier modulus and to the reconstruction of $f(x, y)$ or $\psi(u, v)$ from the autocorrelation function

$$r(x, y) = \mathcal{F}^{-1}|F(u, v)|^2. \quad (2)$$

This problem arises in several disciplines, including optical and radio astronomy, wave-front sensing, holography, and remote sensing.

There are the omnipresent ambiguities: that the object $f(x, y)$, any translation of the object $f(x - x_0, y - y_0)$, the twin image $f^*(-x - x_0, -y - y_0)$, and any of these multiplied by a constant of unit magnitude $\exp(i\phi_c)$ all have exactly the same Fourier modulus. These ambiguities change only the object's position or orientation, not its appearance. If they are the only ambiguities, then we refer to the object as being unique. A solution is considered to be ambiguous only if it differs from the object in ways other than these omnipresent ambiguities.

If nothing is known about the object, then reconstruction from its Fourier modulus is generally ambiguous except for special cases. Fortunately, for many applications one has additional *a priori* knowledge about or constraints on the object. In the astronomy application, for example, the object's spatial brightness distribution, $f(x, y)$, is a real, nonnegative function. For several applications, one has a support constraint, i.e., the object is known to be zero outside some finite area. Even if the support constraint is not known *a priori*, upper bounds can be placed on the support of the object since it can be no larger than half the diameter of the autocorrelation along any direction. Additional measurements or other forms of *a priori* information may be available for specific applications; in this paper we consider real-valued objects with known support, both with and without a nonnegativity constraint.

Until the late 1970's, there was much doubt that the phase-retrieval problem could be solved or that the solution would be useful, because the one-dimensional theory of analytic functions available at the time indicated that there were ordinarily a huge number of ambiguous solutions.[1-3]

The first indications that the two-dimensional (2-D) case is usually unique, despite the lack of uniqueness in one dimension, came from empirical reconstruction results[4,5]: images that were reconstructed resembled the original simulated objects used to compute the Fourier modulus data. These results gave hope that 2-D phase-retrieval problems might be solvable and unique. (Other phase-retrieval problems, such as in electron microscopy in which one has squared-modulus measurements in each of two domains[6] and in x-ray crystallography in which one has the *a priori* information that the object consists of a finite collection of atoms,[7] had been solved; but those earlier successes depended on much greater object-domain constraints than just nonnegativity and support.) Those empirical results gave impetus to attempts to extend the one-dimensional (1-D) theory to two dimensions. Although progress has been made,[8-13] the level of understanding of the 2-D problem has not yet matched that of the 1-D problem.

One of the most enlightening developments has been the work of Bruck and Sodin,[14] who modeled the object distribution as an array of delta functions on a regular grid. Then the continuous Fourier transform becomes the discrete Fourier transform (DFT),

$$F(u, v) = |F(u, v)|\exp[i\psi(u, v)] = \text{DFT}[f(x, y)]$$

$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x, y)\exp\left[-j2\pi\left(\frac{ux}{2M} + \frac{vy}{2N}\right)\right], \quad (3)$$

where the DFT is taken over a $2M \times 2N$ array but $f(x, y)$ is zero outside an $M \times N$ array in order to avoid aliasing in the computation of $r(x, y)$ and $|F(u, v)|^2$. For this discrete case the Fourier transform given in Eq. (3) can then be expressed

as a polynomial of two complex variables, $z = \exp(j\pi u/M)$ and $w = \exp(j\pi v/N)$. It is also equivalent to the z transform. Then the presence of ambiguity in the phase-retrieval problem is equivalent to the factorability of the polynomial. This explains the vast difference between the 1-D and 2-D cases, because polynomials (of degree 2 or greater) of a single complex variable are always factorable, whereas polynomials of two (or more) complex variables are rarely factorable.[14-16] Other interesting results have been obtained by exploiting this discrete model. Fiddy et al.[17] and Nieto-Vesperinas and Dainty[18] described an object support that, by virtue of Eisenstein's irreducibility theorem, guarantees uniqueness. Brames[19] showed that any discrete object having a support whose convex hull has no parallel sides is unique among objects with supports having the same convex hull; so if the convex hull of the support of such an object is known a priori, then it is unique. For these cases, there also exists a closed-form recursive reconstruction algorithm.[20,21]

Whether the objects are discrete or continuous, it is easy to make up cases that are ambiguous. If $g(x, y)$ and $h(x, y)$ are two functions of finite support with Fourier transforms $G(u, v)$ and $H(u, v)$ respectively, then the convolutions

$$f_1(x, y) = g(x, y) * h(x, y) \tag{4}$$

and

$$f_2(x, y) = g(x, y) * h^*(-x, -y) \tag{5}$$

are different objects as long as neither g nor h is conjugate centrosymmetric, they have Fourier transforms

$$F_1(u, v) = G(u, v)H(u, v) \tag{6}$$

and

$$F_2(u, v) = G(u, v)H^*(u, v) \tag{7}$$

that have the same modulus,

$$|F_1(u, v)| = |F_2(u, v)| = |G(u, v)||H(u, v)|, \tag{8}$$

and the objects f_1 and f_2 are ambiguous. This demonstrates the equivalence of phase-retrieval ambiguity to convolutions in the object domain [Eqs. (4) and (5)] and factorability in the Fourier domain [Eqs. (6) and (7)]. Furthermore, if there are K irreducible Fourier factors, then there are 2^{K-1} ambiguous solutions. By this convolutional (products or factors in the Fourier domain) method, it is possible to make up an uncountably infinite number of ambiguous cases even though the theory indicates that ambiguity is rare (of zero probability) in two dimensions. Consider that it is also true that any randomly chosen real number has probability zero of being a rational number (almost all are irrational numbers). Yet any real number, even if irrational, can be approximated arbitrarily well by a rational number. Thus the fact that the probability of any given object's being ambiguous (the Fourier transform being factorable) is zero is not necessarily comforting.

Sanz et al. have shown that the "uniqueness condition is stable in the sense that it is not sensitive to noise."[22] However, their analysis does not shed light on a more practical definition of uniqueness. If a given nonfactorable polynomial is near enough (in an integrated mean-squared difference sense) to a factorable polynomial, then the ambiguous solutions associated with the factorable polynomial will be consistent (to within the noise) with the noisy Fourier-modulus data. Under this circumstance the object may be considered to be ambiguous in a practical sense, even though it may be unique, traditionally speaking. Up to this point it was not known how close an arbitrary polynomial is, on the average, to a factorable polynomial. Furthermore, the existence of ambiguous objects close to a given object is likely to cause the existence of local minima in which iterative reconstruction algorithms will become trapped. Current theory has not adequately addressed these questions, even for the discrete model. These questions can be answered, though, by numerical means, as will be seen below.

One way to test for practical uniqueness is the use of the iterative Fourier-transform algorithm.[4,23-25] If multiple solutions exist, then the algorithm tends to find all of them if many reconstructions are performed, each starting from a different array of random numbers as the initial estimate.[26] In most instances investigated, when the algorithm is applied to the Fourier modulus of an object of interest, if it does not stagnate[25] it reconstructs essentially the correct object,[27] giving strong evidence of uniqueness for those types of object. Furthermore, when noise is added to the Fourier-modulus data, the result is usually a noisy image of the object rather than a completely different reconstruction,[28,29] contrary to some predictions.[30] While this approach has provided some assurance that the phase-retrieval problem is usually unique in the practical sense even in the presence of noise, it has not yielded any quantitative results on the probability of uniqueness for any given level of noise.

An important consideration in the probability of uniqueness is the set of constraints placed on the object. In all cases we assume that the object has finite support (it is zero outside some finite region). The support of the object plays a crucial role. If the object has a delta function known to satisfy the holography condition,[31] then it is unique. As mentioned above, discrete objects having certain supports are guaranteed to be unique.[17,19] In addition, objects having separated parts are more likely to be unique.[32] Although it is less well understood, nonnegativity also plays an important role in uniqueness.

In this paper we establish a methodology for determining the probability of phase-retrieval uniqueness in the practical sense. We have developed a method, suitable for small images, for answering the questions: Given an arbitrary object and its Fourier polynomial, how close is the nearest factorable polynomial, and does it have an ambiguous solution that is significantly different from the given object? In this paper we explore this question for the case of objects defined within 2×2 and 3×2 supports. A derivation of object-domain conditions for factorability provides a means for finding nearest factorable polynomials through a constrained-minimization search over the space of 2×2 or 3×2 ambiguous images. These searches are implemented with different object-domain constraints in a Monte Carlo simulation to estimate the probability that the nearest factorable polynomial, with an ambiguous solution that is significantly different from a given object, is within some distance of the given polynomial. Before describing these main results, we first define the pertinent error metrics and discuss some preliminary results of a grid-search method for finding local minima in phase retrieval, and relationships among minima, ambiguities, and phase-retrieval stagnation.

2. OBJECT-TO-FREQUENCY-DOMAIN MAPPINGS AND ERROR METRICS

A useful means for visualizing the ambiguity problem is through a mapping between the space of objects (images) and the space of Fourier moduli as illustrated in Fig. 1. In Fig. 1 each domain is a finite-dimensional space in which any one point represents a 2-D function. In this diagram $|F(u, v)|$ represents Fourier-modulus data for a unique object and $|G_a(u, v)|$ modulus data for an ambiguous object, since both g_a and g_{ac} map into it. We refer to g_a and g_{ac} as ambiguous counterparts of each other, gotten by conjugating one or more of the Fourier-domain polynomial factors. For the case depicted in Fig. 1, as indicated by the distances between the points, two widely different images, f and g_{ac}, may have similar, but not identical, Fourier moduli. Thus, although f is unique, one might unknowingly reconstruct g_{ac} by a phase-retrieval algorithm given a noisy measurement of $|F|$.

The following error metrics provide a means for quantifying differences in both domains. These metrics are the focus of the numerical approach presented in this paper. (Other related error metrics are also useful.) Given two real-valued functions $g(x, y)$ and $f(x, y)$ defined on an $M \times N$ support and zero padded to a $2M \times 2N$ array, we define the Fourier-modulus error, the error (distance) between $|F(u, v)|$ and $|G(u, v)|$, as

$$\epsilon(g, f) \equiv \left[\frac{\sum_{u,v} [\alpha_f |G(u, v)| - |F(u, v)|]^2}{\sum_{u,v} |F(u, v)|^2} \right]^{1/2}, \qquad (9)$$

where

$$\alpha_f = \left[\frac{\sum_{u,v} |F(u, v)|^2}{\sum_{u,v} |G(u, v)|^2} \right]^{1/2} \qquad (10)$$

is an energy normalization factor, $G(u, v) = \mathrm{DFT}\{g(x, y)\}$, and u and v summations are taken over the intervals 0, 1, ..., $2M - 1$ and 0, 1, ..., $2N - 1$, respectively.

A similar metric defines the object-domain error between $f(x, y)$ and $g(x, y)$:

$$\delta(g, f) \equiv \left[\frac{\sum_{x,y} [\alpha_0 g(x, y) - f(x, y)]^2}{\sum_{x,y} f^2(x, y)} \right]^{1/2}, \qquad (11)$$

where

$$\alpha_0 = \alpha_f \, \mathrm{sign}\left[\sum_{x,y} f(x, y) g(x, y) \right] \qquad (12)$$

and x and y summations are taken over 0, 1, ..., $M - 1$ and 0, 1, ..., $N - 1$, respectively. The parameter α_0 takes into account any differences in scaling and polarity between g and f. Translations are ignored here because the support constraint automatically rules them out. Because $g(x, y)$ and its twin, $g(M - 1 - x, N - 1 - y)$, share the same Fourier modulus, we compute $\delta(g, f)$ for both $g(x, y)$ and its twin and use the smaller of the two values of δ. Of particular interest from the point of view of phase retrieval are images that have a small Fourier-modulus error ϵ, but a large object-domain error δ, since these images may be ambiguous in the practical sense.

3. GRID SEARCHES

Our first approach to understanding the relationship between ϵ and δ, for a collection of images g relative to a given object f, was by a grid search. What we mean by a grid search is illustrated as follows for the case of 3×2 ($M = 3, N = 2$) objects. Given a 3×2 object f, we calculate ϵ and δ for all 3×2 images $g = g_{\mathrm{ref}} + g_{\mathrm{inc}}$, where g_{ref} is another 3×2 real-valued image and

$$g_{\mathrm{inc}} = \begin{bmatrix} s_1 & s_2 & s_3 \\ s_4 & s_5 & s_6 \end{bmatrix}, \qquad (13)$$

where, given a real-valued increment Δs, each s_i can assume values in the set $\{k\Delta s; k = -L, -L + 1, \ldots, 0, 1, \ldots, L\}$. If we think of both f and g as points in a six-dimensional (6-D) space, then we are calculating ϵ and δ for all g's sampled on a symmetric 6-D grid of step size Δs centered about the point g_{ref}, with the grid width equal to $2L + 1$ steps in each of the six dimensions.

This search can become quite extensive as the grid width increases. Since the number of different g_{inc}'s (grid points) is $(2L + 1)^6$, even a five-step search ($L = 2$) requires 15,625 calculations of ϵ and δ. If the search uses the zero image for g_{ref}, we can cut down on redundant calculations of ϵ by eliminating twin images and images with polarity [sign of $F(0, 0)$] opposite f. Note that the saving is in the calculation of ϵ, which is computationally more expensive than the calculation of δ.

Grid-Search Example

The use of a successively finer grid search to find minima in ϵ (which could constitute a phase-retrieval algorithm) and shed light on the properties of ϵ and δ is illustrated in the following example. An integer-valued image f was chosen:

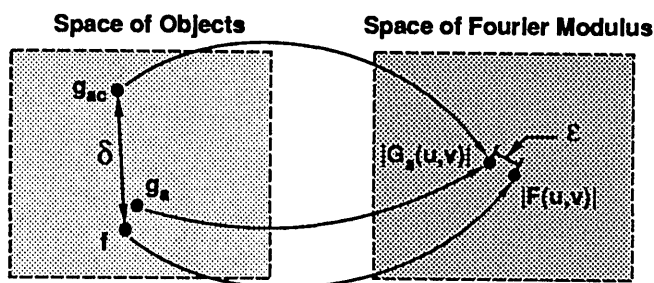$$f = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & -2 \end{bmatrix}. \qquad (14)$$

Fig. 1. Object-space to Fourier-modulus-space mappings of a unique object f and a pair of ambiguous images (g_a, g_{ac}), with error metrics δ and ϵ.
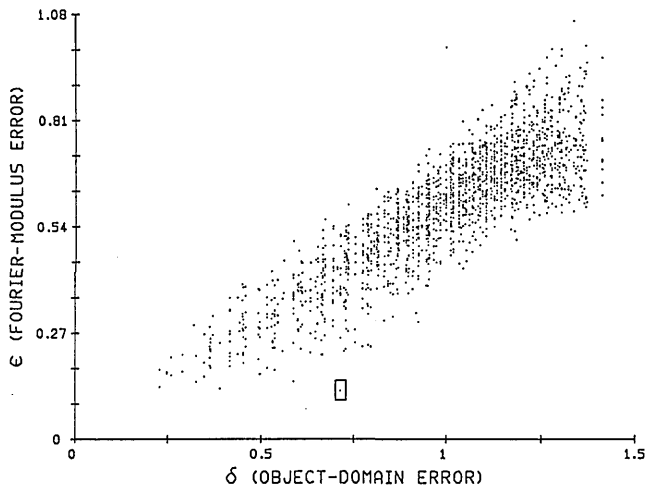
Fig. 2. Fourier-modulus error ϵ versus object-domain error δ for a five-step grid search with step size $\Delta s = 1$. The minimum value of ϵ (excluding $g = f$) is boxed.

A five-step search ($L = 2$) was implemented with g_{ref} equal to the zero function and with $\Delta s = 1$; i.e., the pixel values of g are taken from the set $\{-2, -1, 0, 1, 2\}$. Since the search is centered about the zero function, the twin and polarity search-reduction techniques mentioned above were implemented. The results are displayed in Fig. 2 in the form of a scatter plot of ϵ versus δ. Several features of the scatter plot are noted:

(1) ϵ is less than or equal to δ. The proof of this fact is given in Appendix A.

(2) The vertical striping reflects the discrete nature of the search, i.e., the elements of g take on only integer values.

(3) ϵ and δ can both be greater than unity, despite the normalization that takes place in the denominators of Eqs. (9) and (11).

(4) The scatter plot exhibits a banded type of structure, i.e., the points tend to cluster in a region where both δ and ϵ are large. This is not surprising, since we expect most images that are quite different in the object domain to be quite different in the Fourier-modulus domain as well.

The single point of greatest interest, an outlier with large δ and relatively small ϵ, is outlined by a box in Fig. 2. It corresponds to the image

$$g_0 = \begin{bmatrix} 1 & 1 & -2 \\ 1 & -1 & -2 \end{bmatrix}, \quad (15)$$

with $\delta(g_0, f) = 0.714$ and $\epsilon(g_0, f) = 0.124$. It is the point within the grid search with the lowest value of ϵ aside from $g = f$. Since it represents the point on the grid search closest to being a serious ambiguity, we explored it further by performing another five-step search, with $g_{\mathrm{ref}} = g_0$ of Eq. (15) and a step size of $\Delta s = 1/3$. Because g_{ref} is not the zero function, no data reduction was implemented, and ϵ and δ were calculated for the 15,625 different grid points. Figure 3 shows the scatter plot for this second search for $\epsilon < 0.125$. It is apparent that our initial search with unit steps was quite coarse and that, compared with g_0, there are images with significantly smaller values of ϵ and comparably large values

of δ. The minimum value of ϵ for this grid search corresponds to the image

$$g_1 = \begin{bmatrix} \dfrac{2}{3} & \dfrac{2}{3} & -2 \\ 1 & \dfrac{-2}{3} & \dfrac{-7}{3} \end{bmatrix}, \quad (16)$$

with $\delta(g_1, f) = 0.704$ and $\epsilon(g_1, f) = 0.0648$.

We performed a third five-step search, with $g_{\mathrm{ref}} = g_1$ and $\Delta s = 1/9$. The image corresponding to the minimum ϵ for this search is

$$g_2 = \begin{bmatrix} \dfrac{2}{3} & \dfrac{7}{9} & \dfrac{-17}{9} \\ \dfrac{10}{9} & \dfrac{-2}{3} & \dfrac{-22}{9} \end{bmatrix}, \quad (17)$$

with $\delta(g_2, f) = 0.666$ and $\epsilon(g_2, f) = 0.0569$.

Iterative Grid Searches

The iterative searching above is an approach for finding minima of ϵ. It is summarized more generally by the following steps for the case of $M \times N = 3 \times 2$.

(1) Initialize. Choose g_{ref}, the number of search steps $(2L + 1)$, the step size (Δs), and a step-size reduction factor (r).

(2) Perform a 6-D $(2L + 1)$-step search with $g = g_{\mathrm{ref}} + g_{\mathrm{inc}}$, where

$$g_{\mathrm{inc}} = \begin{bmatrix} s_1 & s_2 & s_3 \\ s_4 & s_5 & s_6 \end{bmatrix} \quad (18)$$

and each s_j, $j = 1, 2, \ldots, 6$, is from the set $\{k\Delta s; k = -L, -L + 1, \ldots, 0, 1, \ldots, L\}$.

(3) Set g_{ref} equal to the image, g, which has the minimum value of ϵ found in the search of the previous step.

(4) Set Δs equal to $\Delta s/r$.

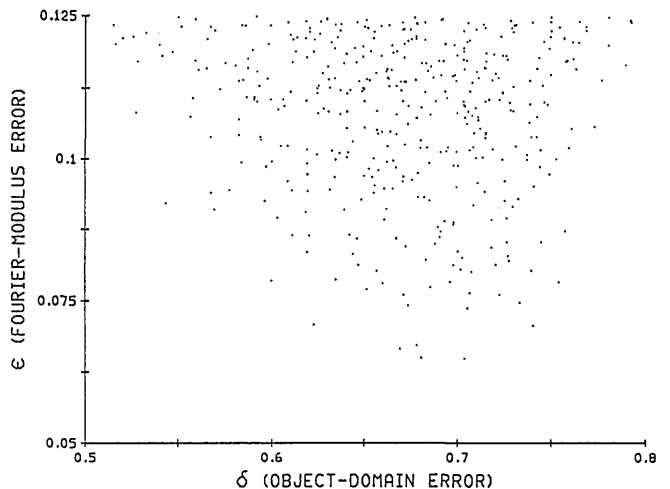(5) Stop if the stopping criterion is met; otherwise go to step 2.

Fig. 3. Fourier-modulus error ϵ versus object-domain error δ for a five-step grid search with $\Delta s = 1/3$ about the minimum of the grid search of Fig. 2. All points satisfying $\epsilon < 0.125$ are shown here.

The stopping criterion is based on the percentage change in the minimum value of ϵ from iteration to iteration, or set for a maximum number of iterations, whichever is satisfied first. For a large value of L, the search time is prohibitive, but the sampling is finer. Also, the initial step-size and step-reduction factor must be chosen carefully, since the step size at the kth iteration is $\Delta s/(r^{k-1})$. If r is chosen too large, the grid may shrink too quickly to progress to a minimum. If Δs is too small, the minimum might not be found because it lies outside the initial grid. The most reliable search uses a slowly shrinking grid with a large number of grid points (large L) that samples the space over a large region. The more finely we sample the space, the more computationally burdensome the algorithm becomes, yet a coarser grid would leave doubt about the reliability of our minimum.

This iterative search could constitute a phase-retrieval algorithm. However, it would be a computationally inefficient algorithm, requiring many thousands of DFT's to converge to a solution for the case of larger objects. Here we are using it only to find a local minimum (the global minimum is at $g = f$ for which $\epsilon = \delta = 0$).

The iterative grid search was tested for f given by Eq. (14) and with the following three sets of parameters: (1) $L = 1$, $\Delta s = 1/2, r = 2$; (2) $L = 2$, $\Delta s = 1/3, r = 3$; and (3) $L = 3$, $\Delta s = 1/4, r = 4$. Each iterative search started with $g_{ref} = g_0$ given by Eq. (15), corresponding to the minimum ϵ found in the first search described above. Each of these searches found a scalar multiple of the same image, g_{min}, given by

$$g_{min} = \begin{bmatrix} 0.623 & 0.749 & -1.871 \\ 1.149 & -0.659 & -2.530 \end{bmatrix}, \qquad (19)$$

with $\delta(g_{min}, f) = 0.667$ and $\epsilon(g_{min}, f) = 0.0558$. This probably represents a deep local minimum for the phase-retrieval problem and could represent a practical ambiguity if the noise in the Fourier modulus data were to exceed $\epsilon(g_{min}, f)$.

4. MINIMA AND PHASE RETRIEVAL

The minimum in ϵ, represented by g_{min} found in the iterative grid searches described above, represents two potential problems for phase retrieval. First, a relatively small error in the modulus data (5.58%) could cause the data to be consistent with g_{min}, which, if reconstructed, would have a very large object-domain error (66.7%). Second, even when it is performing phase retrieval with error-free modulus data, the algorithm could get trapped and stagnate at this local minimum. In particular, the error-reduction (ER) version of the iterative transform algorithm is equivalent to a steepest-descent gradient search method on a cost function closely related to ϵ.[24] Thus, if the local minimum found in our iterative searches were a true local minimum, the ER algorithm could stagnate at this image, unable to find a direction in which to descend. To visualize how ϵ and δ vary around g_{min}, we plot ϵ and δ along the line joining f [Eq. (14)] and g_{min} [Eq. (19)]. Figure 4 shows $\epsilon(g, f)$ and $\delta(g, f)$ versus t for

$$g = f + t(g_{min} - f). \qquad (20)$$

While Fig. 4 represents only a 1-D slice through a 6-D space, it gives the appearance of a minimum in ϵ at $t = 1$ ($g = g_{min}$).
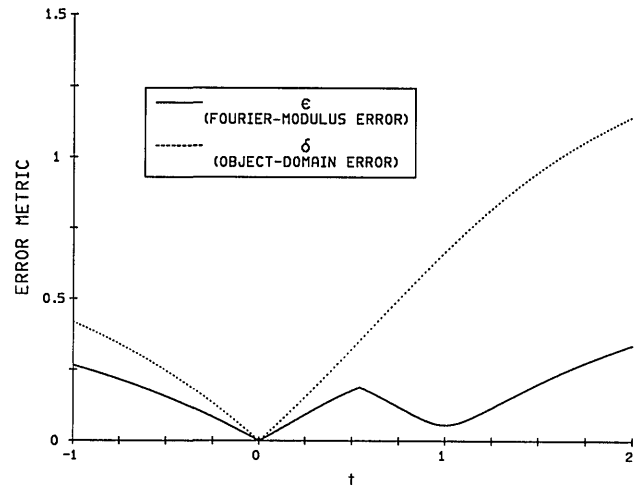

Fig. 4. $\epsilon(g, f)$ and $\delta(g, f)$ versus t for $g = f + t(g_{min} - f)$, the line joining f and g_{min}.

When ER is performed on $|F|$ with g_{min} as the initial guess, stagnation occurs immediately, giving further evidence of the presence of a local minimum.

As another test of ER's tendency to stagnate at a minimum in ϵ, we use g's corresponding to different values of t in Eq. (20) as initial guesses. These values are selected on both sides of the peak in the ϵ curve in Fig. 4. We might expect values of t chosen on the right-hand side of the peak to correspond to initial guesses that stagnate at g_{min} and guesses chosen to the left of the peak to converge to the correct solution, f. Several values of t were selected on both sides of the peak, and the predicted result was verified for all initial guesses.

The hybrid input–output (HIO) version of the iterative Fourier-transform algorithm[24] is one way of climbing out of local minima. Simulated annealing[33] is another. Cycles of HIO iterations followed by ER iterations[24] were used with a variety of starting points: g_0, g_1, g_2, and g_{min}. In each case the HIO/ER combination converged to the correct solution, f, although ER by itself stagnated in each of these same cases. As we will see below, HIO is not always sufficient to overcome stagnation.

5. MINIMA AND AMBIGUOUS IMAGES

A clue to the understanding of the stagnation point described above is its relationship to ambiguous images. Consider again the object f given by Eq. (14). Using methods that are described below, one can verify that the 3×2 ambiguous image whose Fourier modulus is closest to the Fourier modulus of the object f is

$$g_a = \begin{bmatrix} 0.594 & 1.624 & -1.211 \\ 2.330 & 1.415 & -1.730 \end{bmatrix}, \qquad (21)$$

with $\delta(g_a, f) = 0.217$ and $\epsilon(g_a, f) = 0.0859$. The ambiguous counterpart to g_a [gotten by conjugating one of the factors of $G_a(u, v)$] is

$$g_{ac} = \begin{bmatrix} -0.363 & -0.618 & 1.987 \\ -1.422 & 0.600 & 2.837 \end{bmatrix}, \qquad (22)$$

with $\delta(g_{ac}, f) = 0.677$ and $\epsilon(g_{ac}, f) = 0.0859$. A comparison of g_{min} [Eq. (19)] with $-g_{ac}$ (which, for our purposes, is equivalent to g_{ac}) reveals a similarity between this pair of images. The error metrics reveal their similarity in both domains: $\delta(-g_{ac}, g_{min}) = \delta(g_{ac}, g_{min}) = 0.113$ and $\epsilon(g_{ac}, g_{min}) = 0.0663$.

Because $-g_{ac}$ and g_{min} are quit similar, we might expect the ER algorithm with an initial guess of $-g_{ac}$ to stagnate at g_{min}. This is indeed the case after approximately 50 iterations. This result, coupled with the similarity between f [Eq. (14)] and its nearest ambiguity, g_a, might lead us to conclude that ER would find the correct solution if it were started with an initial guess of g_a. This is not the case, however, and the algorithm stagnates after fewer than 20 iterations at

$$g_{stag} = \begin{bmatrix} 0.694 & 1.778 & -1.010 \\ 2.235 & 1.355 & -1.856 \end{bmatrix}, \tag{23}$$

with $\delta(g_{stag}, f) = 0.152$ and $\epsilon(g_{stag}, f) = 0.0631$. This stagnation point is close to g_a, with $\delta(g_a, g_{stag}) = 0.0828$ and $\epsilon(g_a, g_{stag}) = 0.0577$. Because g_{stag} is not in the range of the iterative grid searches that found g_{min}, it was not found earlier. A plot of ϵ and δ along the line joining f and g_{stag} is shown in Fig. 5. Despite the difference in vertical scaling, the minimum in Fig. 5 does not appear to be as deep as that in Fig. 4, so one would suspect there might be a good chance of perturbing g_{stag} enough to get the algorithm out of stagnation. As with g_{min}, it was verified that the HIO is able to move out of stagnation at g_{stag} and to the solution.

Figure 6 depicts the possible relationships in both domains between f, its nearest ambiguous image and counterpart, and the two stagnation points. From the previous results we form the following conjecture: For a given object f and its Fourier modulus $|F|$, stagnation points of the iterative transform algorithm (particularly ER) tend to be near ambiguous images that have Fourier moduli close to $|F|$. This conjecture is supported more strongly by the following example.

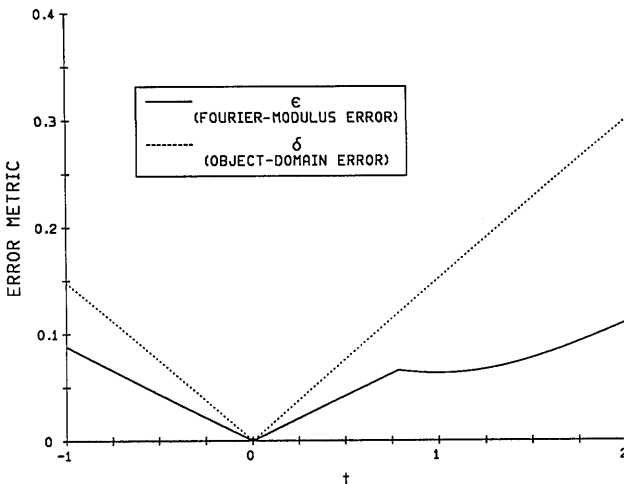Consider the following image f and its nearest ambiguity, g_a, with ambiguous counterpart g_{ac}:

Fig. 5. $\epsilon(g, f)$ and $\delta(g, f)$ versus t for $g = f + t(g_{stag} - f)$, the line joining f and g_{stag}.
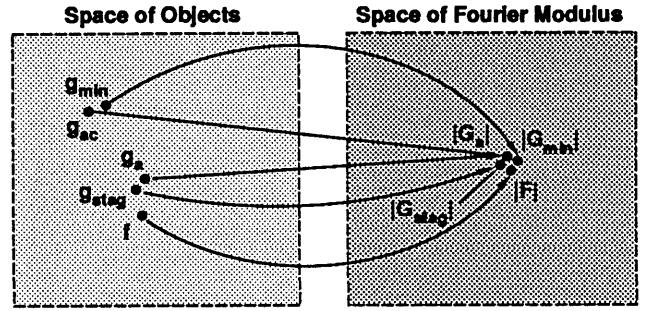

Fig. 6. Object-space to Fourier-modulus-space mappings of an object f, two stagnated images g_{min} and g_{stag}, and the nearest ambiguous image to f with respect to the Fourier-modulus error (g_a, g_{ac}).

$$f = \begin{bmatrix} 0.476 & 3.244 & 1.379 \\ 1.659 & 2.939 & 1.102 \end{bmatrix}, \tag{24}$$

$$g_a = \begin{bmatrix} 0.867 & 3.521 & 1.278 \\ 1.679 & 2.651 & 0.796 \end{bmatrix}, \tag{25}$$

$$g_{ac} = \begin{bmatrix} 0.350 & 2.146 & 3.171 \\ 0.677 & 2.475 & 1.974 \end{bmatrix}, \tag{26}$$

with $\delta(g_a, f) = 0.128$, $\delta(g_{ac}, f) = 0.502$, and $\epsilon(g_a, f) = \epsilon(g_{ac}, f) = 0.00861$. This is a case of a close ambiguity; i.e., the object, f, would be ambiguous in the practical sense unless the data, $|F|$, were low in noise. The ER algorithm was run close to 900 times on $|F|$ using a nonnegativity constraint, each time with a different random initial start. The algorithm converged to the correct solution f of Eq. (24) only 10% of the time. The algorithm stagnated near g_a approximately 9% of the time and at several images close to g_{ac} the rest of the time (81%). When a combination of HIO and ER was used with the same set of random starts, convergence to the solution f was improved to a 26% rate. 74% of the time the algorithm stagnated at one of two different minima, g_{s1} and g_{s2}, each close to the image g_{ac} in Eq. (26):

$$g_{s1} = \begin{bmatrix} 0.353 & 2.143 & 3.172 \\ 0.684 & 2.470 & 1.976 \end{bmatrix} \tag{27}$$

35% of the time, with $\delta(g_{s1}, g_{ac}) = 0.00195$ and $\epsilon(g_{s1}, g_{ac}) = 0.00144$, and

$$g_{s2} = \begin{bmatrix} 0.266 & 1.876 & 2.971 \\ 0.746 & 2.711 & 2.222 \end{bmatrix} \tag{28}$$

39% of the time, with $\delta(g_{s2}, g_{ac}) = 0.0978$ and $\epsilon(g_{s2}, g_{ac}) = 0.0115$. The images g_{s1} and g_{s2} are analogous to g_{min} in Fig. 6. While convergence to g_{s1} is bad in the sense that g_{s1} is different from the solution f [$\delta(g_{s1}, f) = 0.502$], it is still consistent with the given data [$\epsilon(g_{s1}, f) = 0.00848$] and could be considered a solution (albeit the wrong one). The stagnation at g_{s2} is even more troublesome since it is not only similarly consistent with the given data [$\epsilon(g_{s2}, f) = 0.00869$] and far from f [$\delta(g_{s2}, f) = 0.511$] but also is not so close to g_{ac} [$\delta(g_{s2}, g_{ac}) = 0.0978$].

A complete understanding of phase-retrieval stagnation points and their relationship to ambiguous images is not yet available. However, from the limited number of experiments of the type described above, we can say that stagnation points are often related to ambiguous images.

6. NEAREST AMBIGUITIES

In this section we investigate the space of ambiguous images in order to gain some insight into just how close the nearest ambiguous image is to a typical image. This may in turn have implications about how nearest ambiguities relate to stagnation points encountered in iterative phase retrieval. It also will tell us the probability of an ambiguity in the practical sense, as a function of the noise in the Fourier-modulus data.

Object-Domain Conditions for Ambiguity

As described above, ambiguous images are characterized in the Fourier domain by factorable Fourier transforms and in the object domain by being expressible as the convolution of two or more smaller images. We choose the object-domain relationship to characterize the space of ambiguous images. We begin by deriving the ambiguity condition for the smallest possible 2-D ambiguous image (2 × 2 support) and then similarly derive it for a 3 × 2 support.

2 × 2 Ambiguity Conditions

Consider the case of a real-valued image on a 2 × 2 support. It is ambiguous if it can be expressed as the convolution of two 1-D sequences:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} e \\ f \end{bmatrix} * [g \ \ h]$$
$$= \begin{bmatrix} eg & eh \\ fg & fh \end{bmatrix}, \tag{29}$$

where $e, f, g,$ and h are all nonzero (for simplicity only the nonzero rows and columns of the arrays are shown). This gives the following equations for $a, b, c,$ and d:

$$a = eg, \tag{30a}$$
$$b = eh, \tag{30b}$$
$$c = fg, \tag{30c}$$
$$d = fh. \tag{30d}$$

Multiplying Eq. (30a) with Eq. (30d) and Eq. (30b) with Eq. (30c), we arrive at the following 2 × 2 convolution condition:

$$ad = bc. \tag{31}$$

In this case a single ambiguous counterpart to an image satisfying Eq. (31) is generated by convolving one of the 1-D sequences by the flip (rotation by 180°) of the other (equivalent to conjugating the corresponding Fourier factor). However, if $e = f$ and/or $g = h$ (i.e., one of the 1-D sequences is symmetric), then flipping the factor has no effect, and the image is still unique. Furthermore, if $e = -f$ and/or $g = -h$, then a flip of either convolution factor becomes the negative of the original factor. Since we do not consider two images that differ by a scalar multiple (−1 in this case) as ambiguous counterparts, we must also rule out this special case of negative symmetric factors. Therefore the image is unique if $|e| = |f|$ or if $|g| = |h|$. From Eqs. (30) we see that, if $|a| = |c|$ or $|b| = |d|$, then $|e| = |f|$, and if $|a| = |b|$ or $|c| = |d|$, then $|g| = |h|$. When these special cases are combined with Eq. (31), the ambiguity condition for the case of 2 × 2 support becomes

$$ad = bc, \tag{32a}$$
$$|b| \neq |a| \neq |c|. \tag{32b}$$

Note that the inequalities of relation (32b) combined with Eq. (32a) imply that $|b| \neq |d| \neq |c|$.

Equation (32a) describes a three-dimensional surface in the four-dimensional space of real-valued 2 × 2 images. While it is accepted that there is zero probability that an arbitrarily selected object will land on this surface, i.e., the phase-retrieval problem is almost always (with probability 1) unique, in this paper we are concerned with how close the Fourier modulus of a given object is likely to be to the Fourier moduli of images lying upon this surface.

3 × 2 Ambiguity Conditions

The same approach is used to formulate object-domain ambiguity conditions for 3 × 2 images. A 3 × 2 image results from convolving either (a) a 3 × 1 sequence with a 1 × 2 sequence or (b) a 2 × 1 sequence with a 2 × 2 image. Since it is known that any 1-D sequence can always be written as the convolution of smaller sequences, we can write the 3 × 1 sequence of case (a) as the convolution of two 2 × 1 sequences. We can then combine one of these factors with the 1 × 2 factor to give case (b). Thus we need only consider case (b), and our 3 × 2 image is ambiguous if

$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} = [g \ \ h] * \begin{bmatrix} i & j \\ k & l \end{bmatrix}$$
$$= \begin{bmatrix} gi & hi + gj & hj \\ gk & hk + gl & hl \end{bmatrix}, \tag{33}$$

where g and h are nonzero and none of the pairs (i and j) or (i and k) or (j and l) or (k and l) is zero. This gives six nonlinear equations for $a, b, c, d, e,$ and f in terms of $g, h, i, j, k,$ and l. As is shown in Appendix B, these equations can be solved to give the following ambiguity condition:

$$(af - cd)^2 - (ae - bd)(bf - ce) = 0. \tag{34}$$

Equation (34) describes a five-dimensional surface in the 6-D space of real-valued 3 × 2 images. In comparison, for the 2 × 2 case the ambiguity surface describes a three-dimensional surface embedded within a four-dimensional space. Appendix B also shows that Eq. (34) can be solved to give, for example, b in terms of the remaining five values:

$$b = \frac{1}{2}\left[e\left(\frac{c}{f} + \frac{a}{d}\right) \pm (e^2 - 4df)^{1/2}\left(\frac{c}{f} - \frac{a}{d}\right)\right]. \tag{35}$$

An ambiguous, real-valued 3 × 2 image arising from the convolution of a 2 × 1 sequence with a nonfactorable 2 × 2 image can be shown to have an ambiguous counterpart that must also be real valued. However, if the 2 × 2 convolution factor of Eq. (33) can itself be factored, then we have the case of a 3 × 2 image resulting from the convolution of a 3 × 1 sequence with a 1 × 2 sequence. An ambiguous, real-valued image formed in this way will have rows that are scalar multiples of one another; i.e., $a = Kd, b = Ke,$ and $c = Kf$ for some scalar K. This condition makes each difference term in Eq. (34) equal to zero. It is straightforward to show that if $b^2 < 4ac$, then this real-valued ambiguity will have a complex-valued ambiguous counterpart. If the image is constrained to be real valued, then this complex-valued image does not constitute an ambiguity within the space of real-valued images. Furthermore, because this special case is a small subset of the entire ambiguity surface, we expect it to have a relatively minor effect on the likelihood of stagnation due to nearby ambiguities.

Fig. 7. Flow chart for determining the ambiguity of the 3 × 2 real-valued image of Eq. (33). Multiple conditions in a box must all be satisfied for "YES," except where "or" is specified.

The ability to factor an image into the convolution of two or more images is necessary, but not sufficient, for determining ambiguity. When we discussed the ambiguity condition for 2 × 2 images, we considered the special cases of what we called symmetric and negative-symmetric convolution factors. These special cases, as well as the effect of zero-valued pixels, also must be considered for 3 × 2 images. To save space, rather than discussing these exceptions in detail we summarize them in the ambiguity flow chart in Fig. 7.

Nearest Ambiguity by Means of Constrained Minimization

The mathematical description of ambiguities for 2 × 2 and 3 × 2 images can be used to investigate the nearness of a given object to an ambiguous image. We formulate the task of finding the ambiguous image nearest a given object as a multidimensional constrained-minimization problem. By nearest ambiguity we mean the image on the ambiguity surface for which some objective function involving the ambiguous image and the given object is minimized. For the objective function we choose

$$E(g, f) = \sum_{u,v} [|G(u, v)| - |F(u, v)|]^2, \qquad (36)$$

which is just $\epsilon^2(g, f)$ of Eq. (9) with $\alpha_f = 1$ and without the normalization.

Each $M \times N$ image, having $L = M \times N$ pixel values, can be thought of as a single point in an L-dimensional vector space. To emphasize this fact we can denote an image g by the L-dimensional vector $\bar{x}$, where $\bar{x} = (a\ b\ c\ d)^t$ for the 2 × 2 case and $\bar{x} = (a\ b\ c\ d\ e\ f)^t$ for the 3 × 2 case (the ordering of the pixels in the vector $\bar{x}$ is arbitrary). Therefore, for a given image f, we desire to find $\bar{x}$ (or g) on the ambiguity surface that minimizes $E(\bar{x}) \equiv E(g, f)$. (Note that if we did not constrain $\bar{x}$ to be on the ambiguity surface, then we would just be solving the phase-retrieval problem!) If we define the ambiguity surface by $h(\bar{x}) = 0$, then the problem of finding the nearest ambiguity to f can be stated as follows:

Given an object f, find the $\bar{x}$ that minimizes the objective function $E(\bar{x})$ subject to the ambiguity condition $h(\bar{x}) = 0$.

The two image supports for which we have derived ambiguity conditions [Eqs. (32) and (34)] give rise to the following $h(\bar{x})$:

2×2 Images $\{L = 4\}$

$$h(\bar{x}) = ad - bc = 0, \qquad (37)$$

3×2 Images $\{L = 6\}$

$$h(\bar{x}) = (af - cd)^2 - (ae - bd)(bf - ce) = 0. \qquad (38)$$

Iterative Constrained Minimization

Using the mathematical framework developed above, we now implement a generalized reduced-gradient (sometimes referred to as a gradient-projection) method[34] to find the nearest ambiguity to a given image. This method is explained in detail in Appendix C and is summarized below.

In an unconstrained gradient-search method, we search for a minimum to the objective function $E(\bar{x})$ in the direction of $-\nabla E(\bar{x})$, the negative gradient of that function. In a constrained search we still would like to follow the negative gradient, but we are constrained to move along a particular surface within the space, described by the equation $h(\bar{x}) = 0$. We alter the search direction by projecting $-\nabla E(\bar{x})$ onto a tangent plane of $h(\bar{x})$, and we then move along the plane in the direction of the projection, $\bar{p}$, as depicted in Fig. 8. Then, from a point along $\bar{p}$, which is generally not on the constraint surface, we find a nearby (not necessarily the closest) point on the constraint surface. The method used here to return to the constraint surface is detailed in Appendix C. The search for the solution is iterative, and we define our estimate of the solution after the kth iteration as $\bar{x}_k$. At the solution, $\bar{x}_s$, $-\nabla E(\bar{x})$ is perpendicular to the tangent plane to the constraint surface, and the projection onto the tangent plane is zero.

It is difficult to determine whether the minimum found is indeed the global minimum or just a local minimum. In a numerical simulation such as this, one can gain confidence in claiming a minimum as global only through repeated searching with different initial guesses. Our practical criteria for claiming that a minimum, $\bar{x}_s$, is global is that $E(\bar{x}_s)$ is the smallest among all minima found and that it is found more than twice as many times as the total number of minima

Fig. 8. Gradient-projection constrained-minimization algorithm. The search direction is determined by projecting the negative gradient of the objective function onto the tangent plane to the constraint surface.

found, which must be more than four. If the above criteria are not satisfied after 40 different minima are found, then the one that minimizes $E(\bar{x})$ is chosen (and we simply realize that it may not be the global minimum). It should be noted that at points on the surface where $\nabla h(\bar{x}) = 0$ the tangent plane is not defined. If such a singular point is encountered the search may terminate without satisfying a convergence criterion, but the estimate at the singular point may still minimize the objective function over all other estimates (see Appendix C).

Although the constrained-minimization algorithm minimizes an objective function defined in Fourier-modulus space, the search itself takes place on surfaces in object space. The minima found on the surface of Eq. (37) will always correspond to images with two convolution factors, and that usually will be the case for the minima found on the surface of Eq. (38) as well. Thus the nearest ambiguity in Fourier-modulus space to an object f corresponds in object space to any of four images (not counting scalar multiples of these images): the ambiguity, its ambiguous counterpart, and the twin image of each. So, once we have an estimate of the global minimum with respect to Fourier-domain error [Eq. (36)], denoted by g_1, we calculate the object-domain error δ for g_1 and its twin image, retaining the smaller of the two values. We then find the ambiguous counterpart to g_1, denoted by g_{1c}, by convolving one of the factors of g_1 with the twin of the other. After finding the smaller δ for g_{1c} and its twin, we keep as the worst-case nearest ambiguity the larger of this δ and the one retained for g_1 and its twin. Referring back to Fig. 1, the smaller value of δ corresponds to the nearest ambiguity in the object domain, g_a, and the larger retained value of δ corresponds to its ambiguous counterpart, g_{ac}, the worst-case nearest ambiguity. Although g_a and g_{ac} are both nearest ambiguities to f with respect to Fourier-domain error, we differentiate them by defining the worst-case nearest ambiguity as the one with the larger value of the object-domain error, δ, with respect to f. The worse-case nearest ambiguity corresponds to the point in object space farthest from the true image that either is likely to cause local minima to trap phase-retrieval algorithms or could be confused with the true image if the squared error in the data exceeds $E(\bar{x})$.

Monte Carlo Simulations

To investigate the prevalence of ambiguities we implemented the constrained-minimization nearest-ambiguity search in a Monte Carlo simulation in which nearest ambiguous images were found for a large number of random objects $f(x, y)$. Each pixel of the object was an independent, real-valued random number uniformly distributed on the interval [−2, 2] or [0, 4] for nonnegative objects. The results of the Monte Carlo simulations are presented in the form of scatter plots of ϵ versus δ for the worst-case nearest ambiguity. For each random object f, the value of ϵ for the nearest ambiguity is plotted versus the worst-case δ. The interpretation of these scatter plots should not be confused with that of the grid-search scatter plots shown above. Recall that all the (δ, ϵ) pairs in a grid-search scatter plot are calculated by using a single object f and have nothing to do with ambiguities, while each (δ, ϵ) point in Monte Carlo scatter plot represents metrics for the worst-case nearest ambiguity to a different random object f. We computed these plots for five separate

cases: (1) 2×2 objects without a nonnegativity constraint on f, (2) 2×2 objects with a nonnegativity constraint, (3) 3×2 objects without a nonnegativity constraint, (4) 3×2 objects with a nonnegativity constraint, and (5) L-shaped (with $b = c = 0$) 3×2 objects with a nonnegativity constraint. The five cases above represent different constraints on f. The only constraint on the worst-case nearest ambiguity, g_{ac}, is that it lie upon the ambiguity surface corresponding to the support of f.

A typical scatter plot of ~4000 points required ~110 h for the 2×2 objects and ~1500 h for the 3×2 objects on an IBM AT personal computer.

The scatter plots of ϵ versus δ for the 2×2 support cases (1) and (2) are shown in Fig. 9. The points that would cause trouble are those that have small Fourier-modulus error (FME), ϵ, and significantly larger object-domain error (ODE), δ. These troublesome points are likely to induce phase-retrieval algorithm stagnation and/or are ambiguous from a practical point of view when the Fourier-modulus data are sufficiently noisy. One definition of a trouble re-
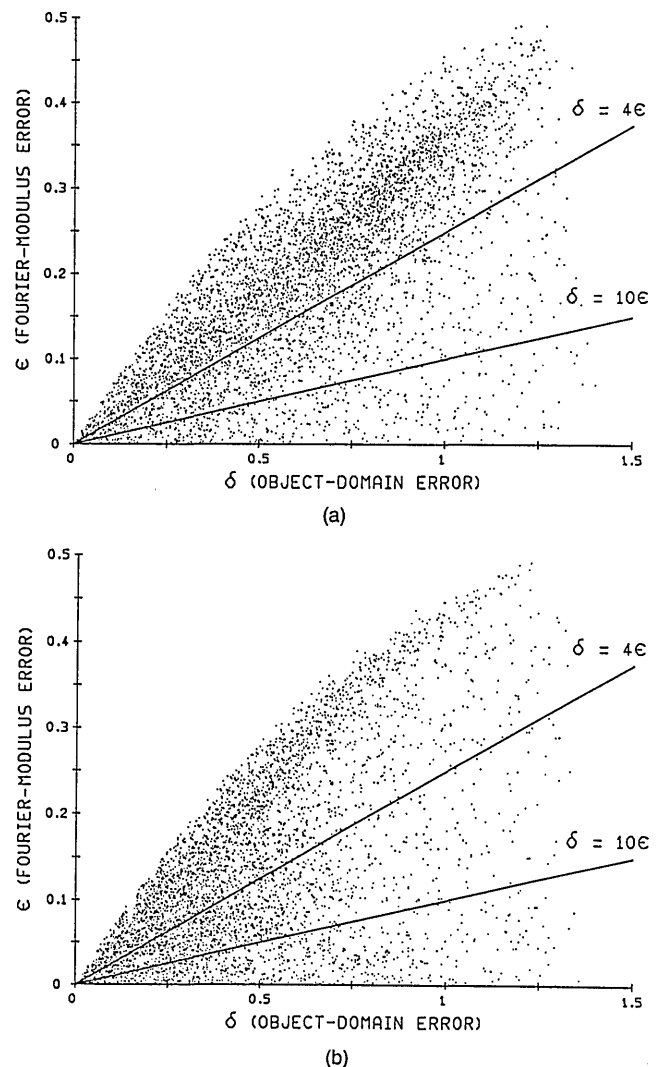

Fig. 9. Fourier-modulus error ϵ versus object-domain error δ for worst-case nearest ambiguities to 2×2 objects. (a) No nonnegativity constraint, 4752 objects; (b) nonnegativity constraint, 4486 objects.
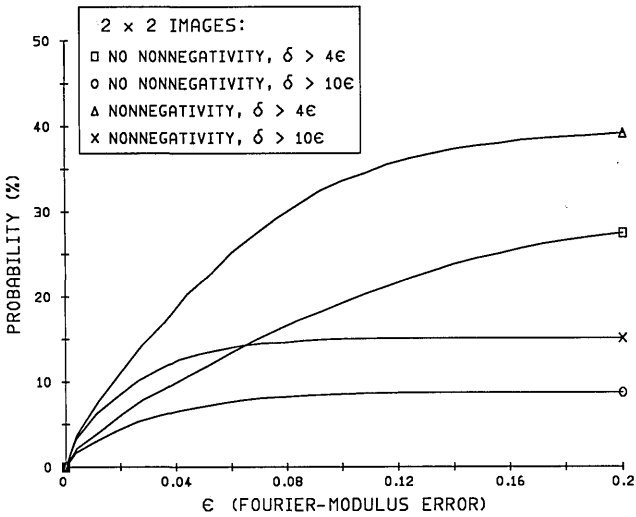
Fig. 10.   Monte Carlo estimates of the probability that the worst-case nearest ambiguity to 2 × 2 objects with and without a nonnegativity constraint has a Fourier-modulus error less than ϵ and an object-domain error greater than $K\epsilon$ ($K = 4$ and $K = 10$).

gion is all the points below the line $\delta = K\epsilon$, shown in Fig. 9 for $K = 4$ and $K = 10$. That is, we do not consider the practical ambiguity problem to be significant unless the error, δ, in the ambiguous reconstruction or stagnation point exceeds 4 times (or 10 times) the error in the Fourier-modulus data. Only then would we consider the ambiguity to be significant. (Although it was easy to show in Appendix A that $\delta > \epsilon$ for any pair of images, an analogous relationship for an image and its worst-case nearest ambiguity has not been developed.) Figure 9(a) (no nonnegativity constraint on f) exhibits a banded structure with a higher density of points above the $\delta = 4\epsilon$ line, which effectively reduces the probability of nearest ambiguities in the trouble region. Figure 9(b) (nonnegativity constraint on f) reveals a higher density of points in the trouble region, particularly for $\delta < 0.5$. Thus the nonnegativity constraint on f actually increases the probability that a random object's Fourier modulus is close to that of an ambiguous image for the 2 × 2 case.

One way to estimate the probability of significant ambiguity is to integrate these scatter plots in the trouble region below the line $\delta = K\epsilon$. If we bin the points below this line with respect to ϵ, we can obtain an estimate of the probability-density function of the probability that the worst-case nearest ambiguity has FME ϵ and $\delta > K\epsilon$. Integrating this estimated probability-density function from 0 to ϵ yields an estimate of the probability that the worst-case nearest ambiguity to an arbitrary object has less than ϵ FME and ODE $\delta > K\epsilon$. These cumulative probability distributions define what we mean by the probability of significant ambiguity. These distributions for cases (1) and (2) are shown in Fig. 10 for $K = 4$ and $K = 10$. Figure 10 verifies our previous observation that the nonnegativity constraint actually improves the chance of significant ambiguity. For example, these estimated distributions tell us that, given an arbitrary, real-valued 2 × 2 object, the probability of finding a worst-case nearest ambiguity with FME $\epsilon < 0.04$ and ODE $\delta > 0.16$ is 10% for f without nonnegativity and 18% for f with nonnegativity.

The same analysis for the 3 × 2 object support [cases (3)

and (4)] reveals the opposite trend. Figure 11 shows the ϵ versus δ scatter plots for the nearest 3 × 2 ambiguities with and without a nonnegativity constraint on f. With no non-negativity constraint, the scatter plot of Fig. 11(a) is uniform in appearance, indicating a greater likelihood of nearby ambiguities in the trouble regions. With the nonnegativity constraint, Fig. 11(b) shows a high concentration of points in the large ϵ, large δ region of the plot, away from the trouble region. It is the nonnegativity constraint that creates the favorable banding effect for the 3 × 2 case. Integrating these plots below the $K = 4$ and $K = 10$ lines yields the probability distributions of Fig. 12. In comparison with the example given for the 2 × 2 nonnegative case, the probability of finding a worst-case nearest ambiguity with FME $\epsilon < 0.04$ and ODE $\delta > 0.16$ is increased to 17% without nonnegativity but reduced by approximately one half to 9% with the non-negativity constraint on f.

One possible reason that nonnegativity reduces the probability of significant ambiguity for the 3 × 2 case is as follows. From Eq. (35) we see that there are no real-valued ambigu-
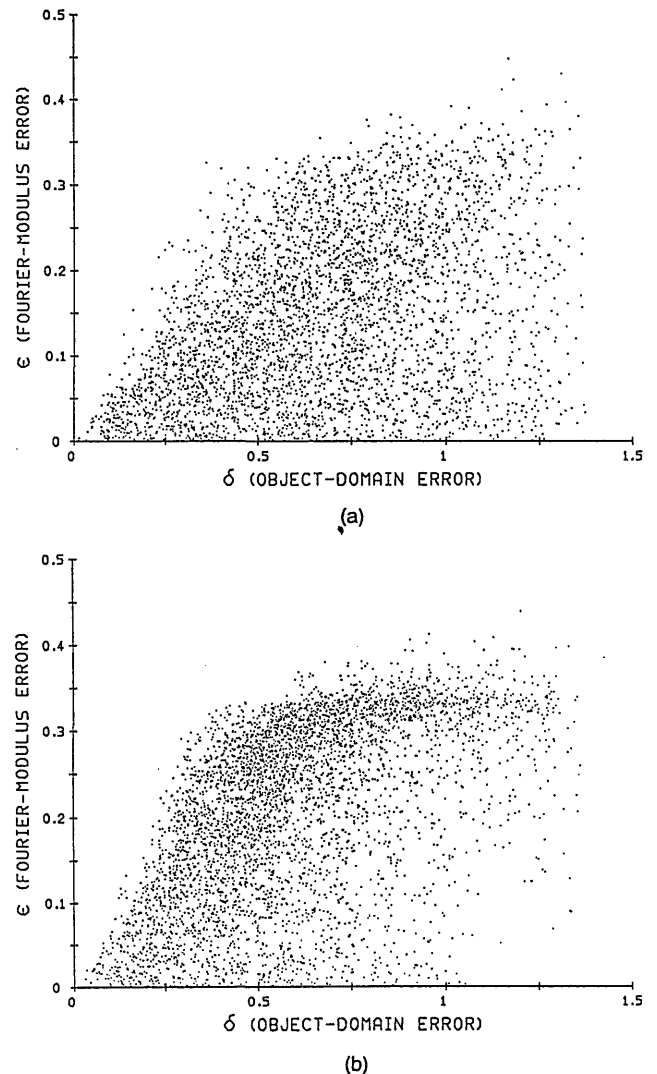

(a)

(b)

Fig. 11.   Fourier-modulus error ϵ versus object-domain error δ for worst-case nearest ambiguities to 3 × 2 objects. (a) No nonnegativity constraint, 4112 objects; (b) nonnegativity constraint, 4601 objects.
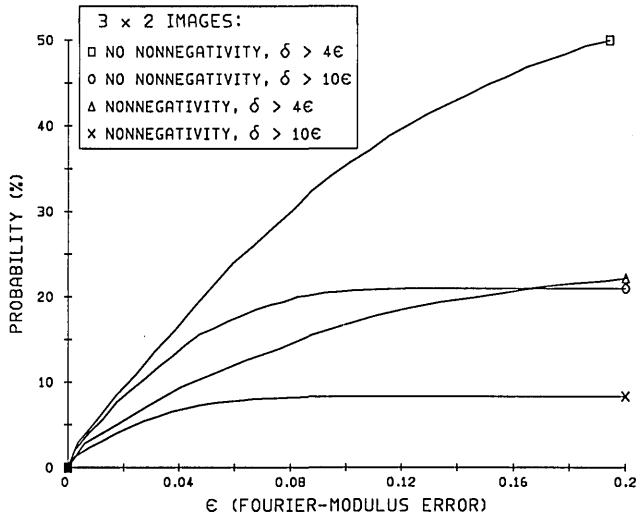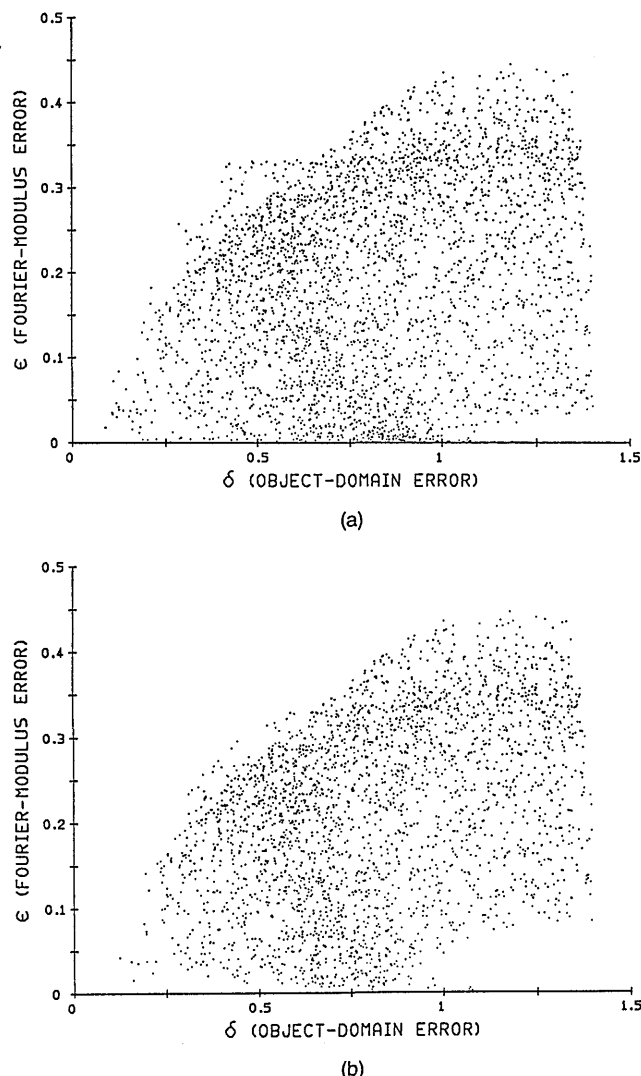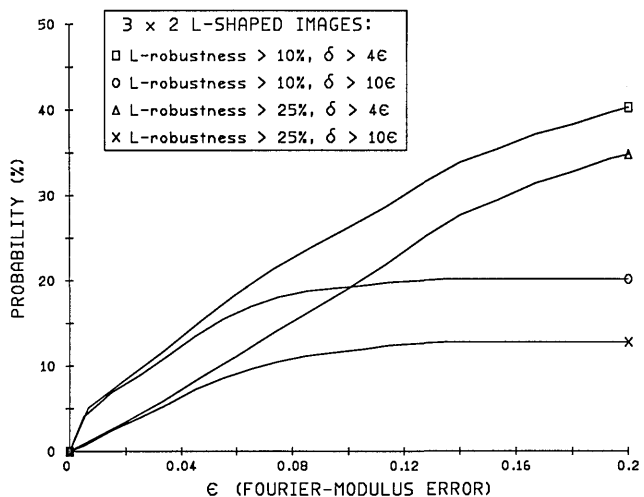
Fig. 12.   Monte Carlo estimates of the probability that the worst-case nearest ambiguity to 3×2 objects with and without a nonnegativity constraint has a Fourier-modulus error less than ϵ and an object-domain error greater than $K\epsilon$ ($K = 4$ and $K = 10$).

ous images for which $e^2 - 4df < 0$. Since $-4df$ is negative for positive d and f, but is positive if one of them is negative, $e^2 - 4df$ is more often negative for nonnegative images. Thus nonnegative objects are less likely to have nearest ambiguities that are nearby (in the object domain) than are objects without a nonnegativity constraint. Since objects that are similar in the object domain will tend to be similar in the Fourier-modulus domain, the nearest ambiguities to nonnegative objects are less likely to be nearby with respect to Fourier modulus as well.

An important point that should be stressed is that the nonnegativity constraint discussed in this section is on the object f and not on the nearest ambiguity. Because of this fact, the nearest ambiguous image to a nonnegative object might not be nonnegative itself; it could contain one or two negative-valued pixels. Thus a nonnegativity constraint in a phase-retrieval algorithm may help to move the image away from a stagnation point near the ambiguity, and the probability of ambiguity in the practical sense would be reduced compared with the results shown here.

At this point it is useful to recall the conjecture made in Section 5, i.e., that stagnation points of the iterative Fourier-transform algorithm tend to be near ambiguous images that have Fourier moduli close to the given Fourier modulus, $|F|$. The example given in Section 5 used an object f and its nearest ambiguity [Eqs. (24)–(26)] taken from the Monte Carlo experiment with 3×2 nonnegative objects. Recall that, for the object f of Eq. (24), after numerous trials we found two stagnation points, g_{s1} and g_{s2}, of both the HIO and ER versions of the iterative Fourier-transform algorithm. The closeness in both domains of these stagnation points to the worst-case nearest ambiguity, g_{ac} [Eq. (26)], was shown. A few more simulations of this type were performed for different nonnegative 3×2 objects. Objects were selected based on the locations in Fig. 11(b) of their worst-case nearest-ambiguity error metrics. All objects selected had a worst-case nearest ambiguity with $0.45 < \delta < 0.55$. Three objects with (significant) worst-case nearest ambiguities with $\epsilon < 0.05$ [as was the case for f of Eq. (24)] were selected,

and, compared with the 26% success rate for f with HIO, the true solution was found 48%, 49%, and 59% of the time, respectively, by using HIO on these three objects. As with f, when the true solution was not found, the algorithm stagnated near the worst-case nearest ambiguity (g_{ac}) to each of the three objects. Two objects with a worst-case nearest ambiguity with $\epsilon \simeq 0.10$ converged to the true solution 78% and 100% of the time, and another object with a worst-case nearest ambiguity with $\epsilon = 0.30$ converged to the solution 100% of the time. Thus stagnation tends to decrease as the nearest ambiguities move farther away with respect to ϵ (equivalently, as the significance of ambiguity decreases). As mentioned above, the limited number of experiments of this type has not yet provided us with a complete understanding of phase-retrieval stagnation points and their relationship to worst-case nearest ambiguous images. Nevertheless, the correlation of the object's worst-case nearest ambiguity having large δ and small ϵ ($\epsilon < 0.05$ for our experiments) with the presence of stagnation points has been convincingly established.

The final case investigated is nonnegative, 3×2 objects with $b = c = 0$, which we call L-shaped objects. The L-shaped support itself mandates uniqueness; i.e., it is not possible to convolve two nontrivial functions to obtain an image with this support. After running the Monte Carlo simulation for these objects, we discovered a class of L-shaped ambiguities that gives rise to misleading results. Consider the object

$$f = \begin{bmatrix} 1.48155 & 0 & 0 \\ 2.01553 & 3.97050 & 0.16831 \end{bmatrix}, \qquad (39)$$

with nearest 3×2 ambiguous image

$$g_a = \begin{bmatrix} 1.48170 & 6.29E\text{-}4 & -2.78E\text{-}3 \\ 2.01419 & 3.97109 & 0.16907 \end{bmatrix}$$

$$= [1 \quad 0.04354] * \begin{bmatrix} 1.48170 & -0.06388 \\ 2.01419 & 3.88340 \end{bmatrix}, \qquad (40)$$

with $\delta(g_a, f) = 7.015E\text{-}4$ and $\epsilon(g_a, f) = 4.167E\text{-}4$. The ambiguous counterpart to g_a, obtained by flipping the first convolution factor in Eq. (40), is

$$g_{ac} = [0.045354 \quad 1] * \begin{bmatrix} 1.48170 & -0.06388 \\ 2.01419 & 3.88340 \end{bmatrix}$$

$$= \begin{bmatrix} 0.06451 & 1.47892 & -0.06388 \\ 0.08769 & 2.18326 & 3.88340 \end{bmatrix}. \qquad (41)$$

The object-domain error between f and g_{ac} as defined by Eq. (11) is $\delta(g_{ac}, f) = 1.0629$. However, comparison of f and g_{ac} reveals that the image g_{ac} is similar to the image f shifted by one pixel to the right. This is because the first convolution factor of Eq. (41) is nearly a delta function, and the second factor is very similar to the image f without its right-hand column. The first convolution factor causes a tapering of the image, making one column much smaller in value than the other nonzero pixels. Flipping one of the convolution factors simply shifts the significant pixels and moves the tapered column to the other side of the image. Because the object-domain error metric δ does not take such shifts into account, the value of $\delta(g_{ac}, f)$ calculated for this case is much too large, resulting in a misleading point on the scatter plot. (If the calculations were to be redone, then this problem could be accounted for by cross correlating g_{ac} with f and

shifting g_{ac} according to the cross-correlation peak to minimize δ.)

A similar problem may occur if the shorter leg of the L-shaped support is tapered, leading to nearest ambiguities that are close to 1-D sequences. To reduce the misleading effects of tapered images on our analysis, we consider only those images that satisfy a bound on the robustness of the L shape. An L-shaped image $\left[\begin{smallmatrix} a & 0 & 0 \\ d & e & f \end{smallmatrix}\right]$ has L robustness $R\%$, defined by

$$\frac{R}{100} = \min\{a, f\}/[(a^2 + d^2 + e^2 + f^2)/4]^{1/2}. \qquad (42)$$

Images with large R are robustly L shaped, whereas images with small R (strongly tapered) are only weakly L shaped.

It should be noted that the same taper problem can also cause misleading ODE calculations of worst-case nearest ambiguities for the 2 × 2 and 3 × 2 images in cases (1)–(4). In these cases, whole rows or columns would have to be significantly smaller than the rms pixel intensity of the image. Since the images are random, it is much less likely for

Fig. 14. Monte Carlo estimates of the probability that the worst-case nearest ambiguity to 3 × 2, nonnegative, L-shaped objects with L robustness greater than $R\%$ ($R = 10$ and $R = 25$) has a Fourier-modulus error less than ϵ and an object-domain error greater than $K\epsilon$ ($K = 4$ and $K = 10$).

this to occur in cases (1)–(4) for which two or more pixels must be small simultaneously than for the L-shaped case (5) for which only a single pixel must be small.

The worst-case nearest-ambiguity scatter plots for nonnegative, L-shaped images with L robustness greater than 10% and 25% are shown in Fig. 13. As the L-robustness requirement is increased, many points clustered about the δ axis disappear. (Had we been able to calculate δ with image shifts taken into account, we would have found these points moving horizontally into the small δ, small ϵ region of the plot.) Despite the nonnegativity of f, these scatter plots are less banded than for general 3 × 2 nonnegative objects—case (4) in Fig. 11(b). This is verified by the estimated distributions for both taper percentages (Fig. 14). For the case of L robustness greater than 25%, the distributions of Fig. 14 achieve a lower probability than does case (4) for values of ϵ less than 0.07, reflected by the small number of points near the origin of the plots in Fig. 13. Therefore, for the low-noise case, the L-shaped support constraint not only prevents ambiguity in the absolute sense but it also makes ambiguity less likely in the practical sense.

7. SUMMARY AND CONCLUSIONS

An ambiguous image is one whose Fourier modulus is identical to the Fourier modulus of a second image that is other than a scaled version, a translation, or a twin of the image. Arbitrary objects are almost never (i.e., with probability zero) ambiguous. Nevertheless, the existence of an ambiguous image close to a given object has two harmful effects: it causes stagnation points for phase-retrieval algorithms and, for the case of noisy Fourier-modulus data, it may cause the solution to be ambiguous in the practical sense. Because of the nonlinearity of the phase-retrieval problem, these issues are difficult to characterize analytically. We investigated the prevalence of ambiguous images for the phase-retrieval problem, using numerical approaches. This is practical because we considered the case of small objects defined on 2 × 2 and 3 × 2 supports.

(a)

(b)

Fig. 13. Fourier-modulus error ϵ versus object-domain error δ for worst-case nearest ambiguities to 3 × 2, nonnegative, L-shaped objects. (a) L robustness > 10%, 3190 objects; (b) L robustness > 25%, 2714 objects.

Using both a new iterative grid-search algorithm and the iterative Fourier-transform algorithm, multiple phase-retrieval experiments were performed, and stagnation points were found that correspond to local minima in the Fourier-domain error metric. These stagnation points were shown to be close to ambiguous images whose Fourier moduli are close to the modulus of the Fourier transform of the object. The implication is that the existence of the ambiguous images causes the local minima to occur. However, the precise relationship between the local minima and the ambiguous images is not yet understood, and nearest ambiguities may not be the sole cause of stagnation.

The prevalence of ambiguities close (with respect to Fourier modulus) to a given object was explored by a Monte Carlo experiment in which nearest ambiguities were found. First, object-domain analytic expressions for the set of ambiguous images were derived for both the 2 × 2 and 3 × 2 supports [Eqs. (37) and (38)]. For the 2 × 2 case, the set of ambiguous images forms a three-dimensional surface embedded in the four-dimensional space of 2 × 2 real-valued images. For the 3 × 2 case, the set of ambiguous images forms a five-dimensional surface embedded in the 6-D space of 3 × 2 real-valued images. Next, a reduced-gradient search technique was used to search along the surfaces of ambiguous images to find the ambiguous image nearest a given object with respect to Fourier modulus. Of the nearest-ambiguity pair of images, one is usually close to the object f, while its ambiguous counterpart is usually a worse case: it is much farther from the given object, yet it has a Fourier modulus identical to the ambiguous image that is close to f. Histograms of Fourier-modulus-domain versus object-domain errors were accumulated in Monte Carlo experiments involving numerous random objects and their worst-case nearest ambiguities. Integration of the histograms, over the points for which the object-domain error is large relative to the Fourier-modulus error, yielded estimates of the probability that a significant ambiguity would occur within a given Fourier-modulus error tolerance. It was found that nonnegativity of the object decreased the probability of significant ambiguity for the 3 × 2 case (as anticipated) but increased the probability of significant ambiguity for the 2 × 2 case. However, since the ambiguous images were allowed to have negative values even when the objects were restricted to be nonnegative, it is likely that the imposition of a nonnegativity constraint in a phase-retrieval algorithm would help to avoid some of those ambiguities. L-shaped images, whose support guarantees uniqueness in the absolute sense, were also investigated. It was found that, for low-noise data, the L-shaped support of the object also makes ambiguity less likely in the practical sense.

Future work should include the application of this approach to objects with larger supports. This is important since it is difficult to extrapolate from these results for 2 × 2 and 3 × 2 supports to the case of most interest: supports with many pixels in each dimension. The probability of significant ambiguity for the 3 × 2 case was of similar magnitude to that of the 2 × 2 case. This is probably because the ambiguity surfaces in both cases were of dimension one less than the dimension of the space of objects. When larger objects are considered, however, this changes. For example, for 3 × 3 objects

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix},$$

factoring into a (3 × 2) convolved with a (1 × 2), the ambiguity condition is given by the simultaneous equations

$$(ah - bg)^2 - (ae - bd)(dh - eg) = 0 \qquad (43)$$

and

$$(ah - bg)(af - cd) - (ae - bd)(ai - cg) = 0. \qquad (44)$$

These describe two eight-dimensional surfaces embedded in a nine-dimensional space of 3 × 3 real-valued objects, the intersection of which would ordinarily be expected to be a seven-dimensional surface embedded in the nine-dimensional space. The ambiguity condition for the factoring of a 3 × 3 object into a (2 × 2) convolved with another (2 × 2) is also given by a pair of simultaneous equations describing two eight-dimensional surfaces embedded in a nine-dimensional space, the intersection of which would ordinarily be a seven-dimensional surface in the nine-dimensional space. Thus for these larger images the dimensionality of the surface of ambiguous images is smaller relative to the space of all objects than for the 2 × 2 or 3 × 2 case; consequently one would expect the probability of significant ambiguity to be less for these larger images. The importance of the shape of the support constraint (convex versus nonconvex versus separated parts, etc.) may also reveal itself more forcefully for larger supports. Finally, a better understanding of the precise relationship between local minima and nearest ambiguous images could lead to methods for avoiding phase-retrieval algorithm stagnation at local minima.

APPENDIX A: PROOF THAT $\epsilon \le \delta$

By definition, $|\alpha_0| = \alpha_f$, or $\alpha_0 = \pm\alpha_f$. The proof that $\epsilon(g, f) \le \delta(g, f)$ can be given by using Parseval's theorem with the definition of $\delta(g, f)$:

$$\delta(g, f) = \left[\sum_{x,y} [\alpha_0 g(x, y) - f(x, y)]^2 \Big/ \sum_{x,y} f^2(x, y) \right]^{1/2}$$

$$= \left[\sum_{u,v} |\pm\alpha_f G(u, v) - F(u, v)|^2 \Big/ \sum_{u,v} |F(u, v)|^2 \right]^{1/2}.$$

$$(A1)$$

By the triangle inequality, given two vectors, v_1 and v_2, $|v_1 - v_2|^2 \ge [|v_1| - |v_2|]^2$. Therefore

$$|\pm\alpha_f G(u, v) - F(u, v)|^2 \ge [\alpha_f |G(u, v)| - |F(u, v)|]^2. \quad (A2)$$

Inserting inequality (A2) into Eq. (A1), we have

$$\delta(g, f) \ge \left[\sum_{u,v} [\alpha_f |G(u, v)| - |F(u, v)|]^2 \Big/ \sum_{u,v} |F(u, v)|^2 \right]^{1/2}$$

$$\equiv \epsilon(g, f). \qquad (A3)$$

APPENDIX B: DERIVATION OF 3 × 2 AMBIGUITY CONDITION

Equation (33) gives us the following six equations:

$$a = gi, \tag{B1a}$$

$$b = hi + gj, \tag{B1b}$$

$$c = hj, \tag{B1c}$$

$$d = gk, \tag{B1d}$$

$$e = hk + gl, \tag{B1e}$$

$$f = hl. \tag{B1f}$$

Multiplying Eqs. (B1a) and (B1f) gives

$$af = ghil, \tag{B2}$$

and multiplying Eqs. (B1c) and (B1d) gives

$$cd = ghjk. \tag{B3}$$

Combining these yields

$$(af - cd)^2 = g^2 h^2 (il - jk)^2. \tag{B4}$$

From Eqs. (B1b), (B1c), (B1e), and (B1f) we have

$$(bf - ce) = h^2(il - jk), \tag{B5}$$

and from Eqs (B1a), (B1b), (B1d), and (B1e) we have

$$(ae - bd) = g^2(il - jk). \tag{B6}$$

Taking the product of Eqs. (B5) and (B6) yields

$$(ae - bd)(bf - ce) = g^2 h^2 (il - jk)^2. \tag{B7}$$

From Eqs. (B4) and (B7) we arrive at the result

$$(af - cd)^2 - (ae - bd)(bf - ce) = 0. \tag{B8}$$

This equation is the condition that must be met in order for the 3 × 2 image of Eq. (33) to be ambiguous.

From Eq. (B8) we can solve for any of the six variables in terms of the other five. For example, by expanding and collecting powers of b, we arrive at

$$b^2(df) - b(aef + cde) + ace^2 + (af - cd)^2 = 0. \tag{B9}$$

The solution of Eq. (B9), which is quadratic in b, is given by

$$b = \frac{[e(af + cd) \pm (e^2 - 4df)^{1/2}(af - cd)]}{2df}$$

$$= \frac{1}{2}\left[e\left(\frac{c}{f} + \frac{a}{d}\right) \pm (e^2 - 4df)^{1/2}\left(\frac{c}{f} - \frac{a}{d}\right)\right]. \tag{B10}$$

APPENDIX C: GENERALIZED REDUCED-GRADIENT METHOD

The generalized reduced-gradient method is a gradient-projection technique used to apply a set of constraints to a minimization problem. The application discussed here uses a single nonlinear homogeneous constraint, $h(\bar{x}) = 0$, and the discussion is presented with this assumption. We begin by defining the tangent plane to a surface:

Given a point $\bar{x}^*$ satisfying $h(\bar{x}^*) = 0$, the tangent plane T at that point is $T = \{\bar{y}: \nabla h(\bar{x}^*) \cdot \bar{y} = 0\}$, where ∇ denotes the gradient with respect to $\bar{x}$ and $\cdot$ denotes the dot product.

Simply stated, all vectors $\bar{y}$ in the tangent plane T are perpendicular to the gradient of $h(\bar{x})$ at $\bar{x}^*$.

In an unconstrained gradient-search method, we would search for a minimum to the objective function $E(\bar{x})$ in the direction of the negative gradient of that function, $-\nabla E(\bar{x})$. In a constrained search, however, the solution is constrained to a particular surface within the space, and we must alter the direction of the search to remain on the surface. We do this by projecting $-\nabla E(\bar{x})$ onto a tangent plane of $h(\bar{x})$ and moving along the plane in the direction of the projection, $\bar{p}$. Because points lying along $\bar{p}$ in general will not lie upon the constraint surface, the goal is to move along $\bar{p}$ and then to return to the surface $h(\bar{x}) = 0$ such that there is a sufficient decrease in the objective function. More will be said below about how to return to the surface from the projection onto the tangent plane.

The solution point, $\bar{x}_s$, satisfies the following first-order condition:

All $\bar{y}$ satisfying $\nabla h(\bar{x}_s) \cdot \bar{y} = 0$ (in the tangent plane at $\bar{x}_s$) must also satisfy $-\nabla E(\bar{x}_s) \cdot \bar{y} = 0$.

The above definition implies that, at the solution, $-\nabla E$ is parallel to ∇h, which in turn implies that the projection $\bar{p}$ is zero. Note that the above definition applies to any minimum and not just to the global minimum.

The search is iterative, and we define $\bar{x}_k$ as our estimate of the solution after k iterations. The goal is to find $\bar{x}_{k+1}$ such that $E(\bar{x}_k)$ significantly decreases at each iteration and to continue iterating until the first-order condition above is satisfied with a sufficient degree of confidence.

We now discuss the reduced-gradient method in more specific terms for the case of a single homogeneous constraint. Let us assume we are working in an L-dimensional space. A tangent plane to $h(\bar{x})$ can be thought of as a surface of dimension one less than the space in which it lies. In order to use projection ideas from linear algebra, we define the tangent plane as a space spanned by a set of basis vectors.

A vector that is perpendicular to the tangent plane to $h(\bar{x})$ at a point $\bar{x} = (x_1 \, x_2 \dots x_L)^t$ is

$$\nabla h(\bar{x}) = \left(\frac{\partial h}{\partial x_1} \quad \frac{\partial h}{\partial x_2} \quad \cdots \quad \frac{\partial h}{\partial x_L}\right)^t. \tag{C1}$$

A set of $L - 1$ linearly independent L-dimensional basis vectors that span the space perpendicular to $\nabla h(\bar{x})$ (i.e., the tangent plane) is (assuming that $\partial h/\partial x_1 \neq 0$)

$$\bar{b}_1 = \left[-\left(\frac{\partial h}{\partial x_1}\right)^{-1}\left(\frac{\partial h}{\partial x_2}\right) \quad 1 \quad 0 \quad 0 \dots 0\right]^t,$$

$$\bar{b}_2 = \left[-\left(\frac{\partial h}{\partial x_1}\right)^{-1}\left(\frac{\partial h}{\partial x_3}\right) \quad 0 \quad 1 \quad 0 \dots 0\right]^t,$$

$$\vdots \qquad\qquad\qquad \vdots$$

$$\bar{b}_{L-1} = \left[-\left(\frac{\partial h}{\partial x_1}\right)^{-1}\left(\frac{\partial h}{\partial x_L}\right) \quad 0 \quad 0 \quad 0 \dots 1\right]^t. \tag{C2}$$

The set of basis vectors defined in Eqs. (C2) enables us to define a projection onto the tangent plane to $h(\bar{x})$. If we let the $\bar{b}$'s be the columns of an $L \times (L - 1)$ matrix,

$$Z \equiv [\bar{b}_1 \quad \bar{b}_2 \ldots \bar{b}_{L-1}], \tag{C3}$$

then the projection of an arbitrary L-dimensional vector, $\bar{\omega}$, onto the space spanned by the columns of Z is[34]

$$\bar{p} = Z(Z^t Z)^{-1} Z^t \bar{\omega}. \tag{C4}$$

From Eq. (C4), the projection of $-\nabla E(\bar{x})$ onto the tangent plane to $h(\bar{x})$ is just

$$\bar{p} = -Z(Z^t Z)^{-1} Z^t \nabla E(\bar{x}). \tag{C5}$$

For each estimate $\bar{x}_k$ of the solution we have $h(\bar{x}_k) = 0$. The reduced-gradient method calculates $\nabla h(\bar{x}_k)$, Z_k, and $-\nabla E(\bar{x}_k)$ and uses these with Eq. (C5) to determine the new search direction:

$$\bar{p}_k = -Z_k(Z_k^t Z_k)^{-1} Z_k^t \nabla E(\bar{x}_k). \tag{C6}$$

Once $\bar{p}_k$ is determined, we must move from $\bar{x}_k$ in the direction of $\bar{p}_k$ to find the next estimate $\bar{x}_{k+1}$. However, we must have $h(\bar{x}_{k+1}) = 0$, and, in general, it is not possible to find a step size $\gamma_k \neq 0$ along $\bar{p}_k$ such that $h(\bar{x}_k + \gamma_k \bar{p}_k) = 0$. It becomes necessary to deviate from $\bar{p}_k$ to return to the surface for our next estimate. This estimate becomes

$$\bar{x}_{k+1} = \bar{x}_k + \gamma_k \bar{p}_k + \bar{q}_k, \tag{C7}$$

with

$$h(\bar{x}_{k+1}) = 0, \tag{C8}$$

where γ_k and $\bar{q}_k$ are chosen such that $E(\bar{x}_{k+1}) < E(\bar{x}_k)$. Determining the scalar step size γ_k and the direction back to the surface, $\bar{q}_k$, in Eq. (C7) that minimize $E(\bar{x}_{k+1})$ can be a complicated subproblem.

Rather than spending too much computation time determining the optimal γ_k and $\bar{q}_k$, we opt for a simpler approach to finding an $\bar{x}_{k+1}$ that produces a sufficient decrease in the objective function. We do this by (1) selecting a value for γ_k, then (2) using $\bar{x}_k + \gamma_k \bar{p}_k$ for all but one of the components of $\bar{x}_{k+1}$, and then (3) using Eq. (C8) to determine the last component. Equation (35) is an example of Eq. (C8) for solving for the component b. The objective function is evaluated to determine whether there is a sufficient decrease. If we are not satisfied with the new estimate, we choose another value of γ_k and repeat the procedure. Using this procedure, we can think of the objective function as a function of γ and can set γ_k to the value of γ that minimizes $E(\gamma)$. One could use any of a number of standard line search techniques to estimate γ_k, but we used a slightly different method to estimate this minimum and to find $\bar{x}_{k+1}$.

Iterative Quadratic Fit

The technique implemented to minimize $E(\gamma)$ with respect to γ can best be described as an iterative quadratic fit (IQF). It uses quadratic curve fitting to approximate the minimum of $E(\gamma)$ iteratively and thus determine γ_k. The description of the IQF below assumes the ability to fit a quadratic polynomial to three points:

(1) Initialize: $\gamma_1 = \gamma_{m2} = 0$, γ_2, γ_3.
(2) Calculate $E(\gamma_1)$, $E(\gamma_2)$, and $E(\gamma_3)$.
(3) Calculate γ_{m1}, the value of γ that corresponds to the minimum of the quadratic polynomial in γ fit to the points $[\gamma_1, E(\gamma_1)]$, $[\gamma_2, E(\gamma_2)]$, $[\gamma_3, E(\gamma_3)]$.
(4) Calculate $E(\gamma_{m1})$.

(5) If $|\gamma_{m1} - \gamma_{m2}| < \beta$, then $\gamma_k \leftarrow \gamma_{m1}$ and stop; otherwise continue with step (6).
(6) Of γ_1, γ_2 and γ_3, find the two that are closest to γ_{m1}. Call these γ_{c1} and γ_{c2}.
(7) Set: $\gamma_1 \leftarrow \gamma_{c1}, E(\gamma_1) \leftarrow E(\gamma_{c1})$,
$\gamma_2 \leftarrow \gamma_{c2}, E(\gamma_2) \leftarrow E(\gamma_{c2})$,
$\gamma_3 \leftarrow \gamma_{m1}, E(\gamma_3) \leftarrow E(\gamma_{m1})$,
$\gamma_{m2} \leftarrow \gamma_{m1}$.
(8) Go to step (3).

The initial values of γ_2 and γ_3 should be chosen based on experimentation and observation of typical $E(\gamma)$ versus γ curves. These values are not crucial to the success of the quadratic fit but should be spaced well enough to give a reasonable initial fit. The value of the termination parameter β should be based on the degree of accuracy needed and should be chosen large enough to avoid excessive iterations.

The success of the IQF depends largely on the shape of $E(\gamma)$. If $E(\gamma)$ is not fairly smooth, the IQF may not find the actual minimum; this is not a problem if a sufficient decrease in E is achieved. A more difficult problem occurs when the projection onto the tangent plane extends into a region of the 6-D space for which the equation for a return to the surface is not defined. As an example, consider using Eq. (35) to return to the surface by calculating b given the other five variables. If a range of values of γ exists for which $\gamma \bar{p}_k$ extends into the region where $e^2 - 4df < 0$, then b (which is by definition real valued) and hence $E(\gamma)$ will not be defined over this range. When we encountered a case such as this, we implemented a Fibonacci line search[34] to estimate the minimum of $E(\gamma)$ on the interval γ for which $E(\gamma)$ is defined. It should be stressed that these potential problems arise out of the method used here to return to the ambiguity surface, and other methods exist that may circumvent this but that are more computationally burdensome.

SPECIFICS TO THE NEAREST-AMBIGUITY SEARCH

Since we have discussed the constrained-minimization technique in somewhat general terms to this point, let us now mention some details and summarize the procedure.

The gradient of $E(\bar{x})$ of Eq. (36) can be computed by using the following relationship[24]:

$$\frac{\partial E}{\partial g(x, y)} = 2MN[g(x, y) - g'(x, y)], \tag{C9}$$

where

$$\text{DFT}\{g'(x, y)\} = \frac{|F(u, v)|}{|G(u, v)|} G(u, v). \tag{C10}$$

Since the ordering of the pixels of $g(x, y)$ in the vector $\bar{x}$ is defined, Eq. (C9) can be used to calculate the components of $\nabla E(\bar{x})$ using two DFT's [since $|F(u, v)|$ is given].

The various steps of the reduced-gradient constrained-minimization algorithm are as follows:

1. Initialization
   (a) Determine $|F(u, v)|$.
   (b) Make an initial guess, $\bar{x}_0$, such that $h(\bar{x}_0) = 0$.
   (c) Compute $E(\bar{x}_0)$.
   (d) $k = 0$.
2. Calculating the search direction, $\bar{p}_k$
   (a) Compute $\nabla h(\bar{x}_k)$.

(b) Form Z_k, Z_k^t.

(c) Compute $\nabla E(\bar{x}_k)$.

(d) Compute $\bar{p}_k = -Z_k(Z_k^t Z_k)^{-1} Z_k^t \nabla E(\bar{x}_k)$.

3. Iterative Quadratic Fit to find $\bar{x}_{k+1}$ from $\bar{x}_k$ and $\bar{p}_k$

4. If $[E(\bar{x}_k) - E(\bar{x}_{k+1})]/E(\bar{x}_k) < \alpha$,

then: Done; estimate of minimum is $\bar{x}_{k+1}$.

else: (a) $k \leftarrow k + 1$.

(b) Go to step 2.

The termination condition in step 4 above is based on a percentage change between successive iterations. The bound α is selected to reflect the precision of the estimate of the minimum. While it may be tempting to use the condition that $-\nabla E$ is perpendicular to the tangent plane, that is,

$$-\nabla E(\bar{x}_{k+1}) \cdot \bar{p}_{k+1} < \zeta \tag{C11}$$

for some small ζ, it is also difficult to pick the value of ζ that will consistently give us the same confidence in the precision of our estimate without choosing it so small that it causes needless iterations in many cases.

ACKNOWLEDGMENTS

REFERENCES

1. E. Wolf, "Is a complete determination of the energy spectrum of light possible from measurements of the degree of coherence," Proc. Phys. Soc. (London) 80, 1269–1272 (1962).
2. A. Walther, "The question of phase retrieval in optics," Opt. Acta 10, 41–49 (1963).
3. E. M. Hofstetter, "Construction of time-limited functions with specified autocorrelation functions," IEEE Trans. Inf. Theory IT-10, 119–126 (1964).
4. J. R. Fienup, "Reconstruction of an object from the modulus of its Fourier transform," Opt. Lett. 3, 27–29 (1978).
5. 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).
6. W. O. Saxton, Computer Techniques for Image Processing in Electron Microscopy (Academic, New York, 1978).
7. G. H. Stout and L. H. Jensen, X-Ray Structure Determination (Macmillan, London, 1968).
8. W. Lawton, "A numerical algorithm for 2-D wavefront reconstruction from intensity measurements in a single plane," in 1980 International Optical Computing Conference, W. T. Rhodes, ed., Proc. Soc. Photo-Opt. Instrum. Eng. 231, 94–98 (1980).
9. M. Nieto-Vesperinas, "Dispersion relations in two dimensions: application to the phase problem," Optik (Stuttgart) 56, 377–384 (1980).
10. I. Manolitsakis, "Two-dimensional scattered fields: a description in terms of the zeros of entire functions," J. Math. Phys. 23, 2291–2298 (1982).
11. R. Barakat and G. Newsam, "Necessary conditions for a unique solution to two-dimensional phase recovery," J. Math. Phys. 25, 3190–3193 (1984).
12. J. L. C. Sanz and T. S. Huang, "Unique reconstruction of a band-limited multidimensional signal from its phase or magnitude," J. Opt. Soc. Am. 73, 1446–1450 (1983).
13. I. S. Stefanescu, "On the phase retrieval problem in two dimensions," J. Math. Phys. 26, 2141–2160 (1985).
14. Yu. M. Bruck and L. G. Sodin, "On the ambiguity of the image reconstruction problem," Opt. Commun. 30, 304–308 (1979).
15. L. Carlitz, "The distribution of irreducible polynomials in several indeterminates," Ill. J. Math. 7, 371–375 (1963).
16. M. H. Hayes and J. H. McClellan, "Reducible polynomials in more than one variable," Proc. IEEE 70, 197–198 (1982).
17. M. A. Fiddy, B. J. Brames, and J. C. Dainty, "Enforcing irreducibility for phase retrieval in two dimensions," Opt. Lett. 8, 96–98 (1983).
18. M. Nieto-Vesperinas and J. C. Dainty, "A note on Eisenstein's irreducibility criterion for two-dimensional sampled objects," Opt. Commun. 54, 333–334 (1985).
19. B. J. Brames, "Unique phase retrieval with explicit support information," Opt. Lett. 11, 61–63 (1986).
20. J. R. Fienup, "Reconstruction of objects having latent reference points," J. Opt. Soc. Am. 73, 1421–1426 (1983).
21. T. R. Crimmins, "Phase retrieval for discrete functions with support constraints," J. Opt. Soc. Am A 4, 124–134 (1987).
22. J. L. C. Sanz, T. S. Huang, and F. Cukierman, "Stability of unique Fourier-transform phase reconstruction," J. Opt. Soc. Am. 73, 1442–1445 (1983).
23. J. R. Fienup, "Space object imaging through the turbulent atmosphere," Opt. Eng. 18, 529–534 (1979).
24. J. R. Fienup, "Phase retrieval algorithms: a comparison," Appl. Opt. 21, 2758–2769 (1982).
25. J. R. Fienup and C. C. Wackerman, "Phase-retrieval stagnation problems and solutions," J. Opt. Soc. Am. A 3, 1897–1907 (1986).
26. P. VanToorn, A. H. Greenaway, and A. M. J. Huiser, "Phaseless object reconstruction," Opt. Acta 7, 767–774 (1984).
27. J. R. Fienup, "Experimental evidence of the uniqueness of phase retrieval from intensity data," in Indirect Imaging, J. A. Roberts, ed., Proceedings of International Union of Radio Science/International Astronomical Union Symposium, August 30–September 2, 1983, Sydney, Australia (Cambridge U. Press, Cambridge, 1984), pp. 99–109.
28. G. B. Feldkamp and J. R. Fienup, "Noise properties of images reconstructed from Fourier modulus," in 1980 International Optical Computing Conference, W. T. Rhodes, ed., Proc. Soc. Photo-Opt. Instrum. Eng. 231, 84–93 (1980).
29. R. G. Paxman, J. R. Fienup, and J. T. Clinthorne, "Effect of tapered illumination and Fourier intensity errors on phase retrieval," in Digital Image Recovery and Synthesis, P. S. Idell, ed., Proc. Soc. Photo-Opt. Instrum. Eng. 828, 184–189 (1987).
30. A. M. J. Huiser and P. VanToorn, "Ambiguity of the phase-reconstruction problem," Opt. Lett. 5, 499–501 (1980).
31. E. N. Leith and J. Upatnieks, "Reconstructed wavefronts and communication theory," J. Opt. Soc. Am. 52, 1123–1130 (1962).
32. T. R. Crimmins and J. R. Fienup, "Uniqueness of phase retrieval for functions with sufficiently disconnected support," J. Opt. Soc. Am. 73, 218–221 (1983).
33. M. Nieto-Vesperinas, R. Navarro, and F. J. Fuentes, "Performance of a simulated-annealing algorithm for phase retrieval," J. Opt. Soc. Am. A 5, 30–38 (1988).
34. P. E. Gill, W. Murray, and M. H. Wright, Practical Optimization (Academic, London, 1981).
35. J. H. Seldin and J. R. Fienup, "Numerical investigation of phase retrieval uniqueness," in Digest of Topical Meeting on Signal Recovery and Synthesis III (Optical Society of America, Washington, D.C., 1989), pp. 120–123.