3-D Locator Sets of Opaque Objects for Phase Retrieval

J.R. Fienup,*† B.J. Thelen,† M.F. Reiley,*† and R.G. Paxman*†

* Advanced Modular Power Systems Inc.
4667 Freedom Drive
Ann Arbor, MI 48108

† ERIM International
P.O. Box 134008
Ann Arbor, MI 48113-4008

ABSTRACT

Generating a 3-D (angle-angle-range) image can be accomplished by gathering 3-D far-field (angle-angle-frequency) heterodyne array data with multiple laser wavelengths and performing a 3-D Fourier transform. However, since heterodyne detection is difficult at optical frequencies, the collection system can be greatly simplified if direct (intensity) detection is performed instead. Then to reconstruct an image one would need a phase-retrieval algorithm. To assist the reconstruction algorithm, we place bounds (called locator sets) on the support (shape) of the illuminated object, derived from the support of the autocorrelation function, which can be computed from the Fourier intensity data. We have developed 3-D locator sets for getting tight bounds on the object support. These new locator sets are more powerful than those for 2-D imaging, and some of them make explicit use of the fact that the illuminated opaque object is effectively a 2-D surface embedded in 3-D space. For those cases in which it is tight enough, the locator set itself may be all that we need to give an accurate height profile of the object.

Keywords: Phase retrieval, support constraint, profile retrieval, opacity, laser imaging, 3-D imaging

1 INTRODUCTION

A novel imaging modality employs frequency-tunable laser illumination and Fourier intensity measurements to arrive at a fine-resolution, 3-D image without the need for imaging optics (lenses) or heterodyne detection.[1-2] This imaging modality, referred to as Phase Retrieval with an Opacity Constraint for LAser IMaging (PROCLAIM), is described in more detail in a companion paper in this volume.[3] It uses the fact that most objects are opaque, meaning that they reflect light only from their front surfaces, making them, as far as the imaging system is concerned, 2-D surfaces embedded in a 3-D volume. We can use this opacity as a constraint to help solve the phase-retrieval problem of reconstructing an object from the magnitude of its Fourier transform,[4] which is otherwise difficult for complex-valued (i.e., coherently illuminated) objects.[5,6] The key to solving the image reconstruction problem is to have bounds on the support of the object. The support of the object is the set of points outside of which we know that it is zero. The tighter (smaller) the support constraint is (while still including the entire object), the better the image-reconstruction algorithm works.

Email: fienup@erim.org; Telephone: (313)994-1200 ext. 2500; Fax: (313)994-5704

SPIE Vol. 3170 • 0277-786X/97/$10.00

By inverse Fourier transforming the measured Fourier intensity data, we get the 3-D complex autocorrelation of the object. Our goal is to compute the tightest support constraint possible from the autocorrelation information. The support constraint can serve, not only as a means of making the image reconstruction algorithm work well, but also as an end itself. If the support constraint is tight enough, it can give us the desired 3-D object profile information even if we do not reconstruct an image.

In Section 2 we describe the processing steps necessary to compute a 3-D image from the measured data, in Section 3 we describe new algorithms for computing support constraints and show results with simulated data.

2 PROCESSING ALGORITHMS

Figure 1 shows the data flow through the major algorithms necessary for fine-resolution reconstruction.

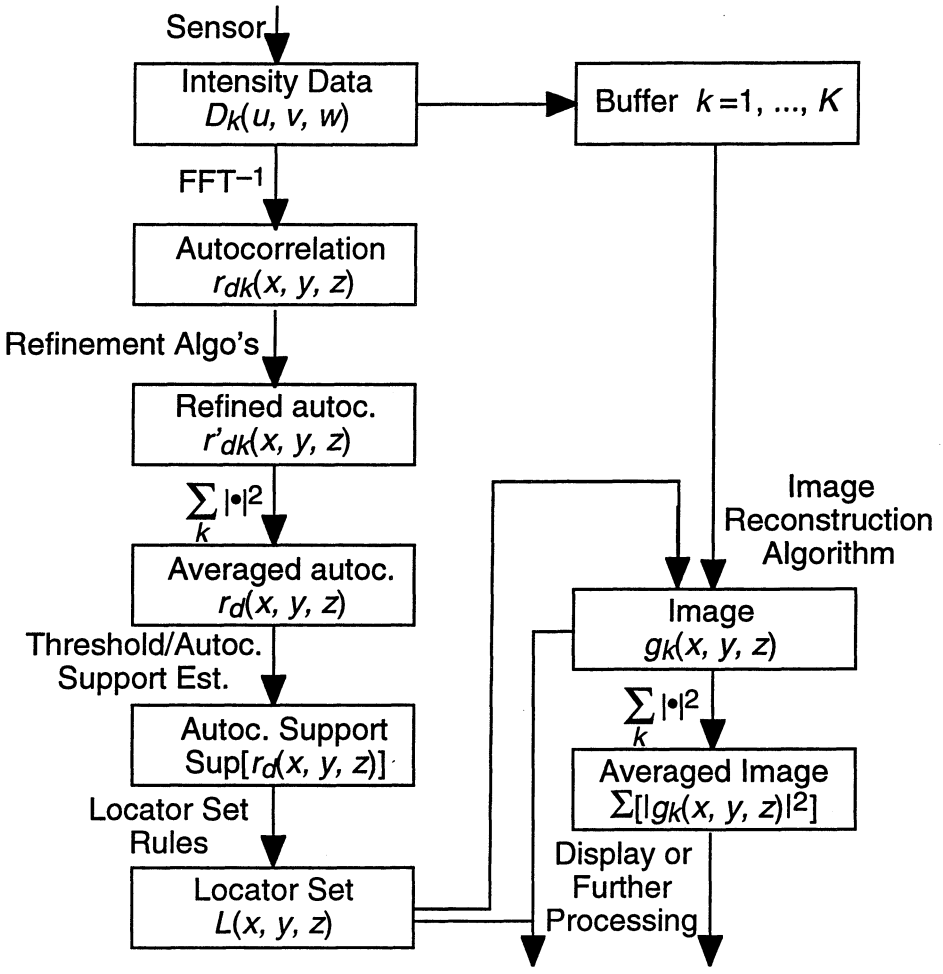

Figure 1. Data Processing Steps for PROCLAIM.

The data, $D_k(u, v, w)$, for the k^{th} speckle realization, are noisy 3-D Fourier intensity measurements of the coherent optical fields. The coordinates (u, v) are in angle-angle (Fourier-plane CCD detector pixels) space and w is laser frequency. Inverse 3-D Fourier transformation of the Fourier intensity gives the 3-D complex autocorrelation of the image, $r_{dk}(x, y, z)$. This has a large impulse response term at the origin which we subtract off. We accomplish this by subtracting the mean of $D_k(u, v, w)$ before the Fourier transformation. The complex autocorrelations suffer from two effects that make it difficult to estimate their support and require refinement algorithms: sidelobes outside the support of $r_{dk}(x, y, z)$ and speckle nulls within the support of $r_{dk}(x, y, z)$. We attack the sidelobes by using Spatially Variant Apodization,[7] which eliminates most sidelobes without degrading resolution. We must do this on the complex autocorrelation. We attack the speckle drop-outs in a variety of ways. A speckle reduction algorithm can reduce them.[8] If we have multiple speckle realizations, we average their intensities to further reduce the effects of speckle. We then threshold the result to arrive at an estimate of the autocorrelation support. We can fill in the remaining speckle nulls with a closing (dilation followed by erosion) operation.[6]

From the autocorrelation support we use intersection rules[9,10] to compute locator sets. This is described further in Section 3.

In some problems, such as target classification, the locator set may be tight enough to successfully perform the task. Alternatively, we can use the locator set, which we compute from multiple speckle realizations if they are available, as a support constraint, along with an individual Fourier intensity data set to reconstruct a complex-valued, speckled image, for each speckle realization, using a phase retrieval algorithm. The intensities of the reconstructed images from all the speckle realizations can then be averaged to arrive at a speckle-reduced 3-D image.

3 LOCATOR SETS

Given an object, we can easily compute its autocorrelation function; and given the support of an object, we can easily compute the support of its autocorrelation. The inverse problem -- given an autocorrelation support, determine the object support -- is difficult, and often not unique. Typically our best hope is to compute a locator set, which is a support within which any object support (or its twin) that could give rise to the autocorrelation support must fit. (What we previously called "single-sided locator sets"[10] we now call "locator sets.") We previously developed rules for computing locator sets in 2-D, but we have recently proven that most of these rules work in 3-D as well. In addition, we have developed a new rule that takes advantage of the opacity for 3-D objects. Finally, we developed a method for making a locator set tighter by identifying which of its points are necessary to the given autocorrelation support.

3.1 3-D Extensions of Locator-Set Rules

The concept of support is important to our rules for determining locator sets. For the object being imaged, the support is the set of points in 3-D that are occupied by the object or mathematically speaking, it is the set of points at which the complex object reflectivity function is non-zero. Our standard notation for the set corresponding to the object support is S. The support of the object autocorrelation is defined similarly, i.e., the autocorrelation support is the set of points in 3-D where the autocorrelation function is non-zero, and our standard notation corresponding to this set is A. In practice, what is actually detected is a noisy measurement of the intensity of the Fourier transform of

the object, which is equivalent to a noisy measurement of the autocorrelation of the complex object reflectivity function. As discussed earlier, an important component of the phase retrieval algorithm is a set, which we refer to as a locator set, which bounds the true (but unknown) object support S. We have developed rules for determining such locator sets from accurate measurements of the autocorrelation support, based on the simplifying assumption that

$$A = S - S \equiv \{x - y : x, y \in S\} \quad . \tag{1}$$

The relationship in (1) is exactly true in the absence of noise and speckle nulls, thus making this a natural starting point for our investigations. In addition, we have found in practice that the resultant rules are relatively robust to noise and speckle nulls with the proper pre-processing of the data. The rules presented in this subsection are straightforward generalizations of previous results for 2-D[10] and are based on looking for specific patterns in what we refer to as "extreme points" of the autocorrelation support A. For the set A in 3-D space and a 3-D vector u, the set of extreme points in A relative to the u-direction is defined as

$$E(A, u) \equiv \{w \in A : \langle u, w \rangle \geq \langle u, w' \rangle \text{ for all } w' \in A\} \quad , \tag{2}$$

where $\langle u, w \rangle$ denotes the inner product in 3-D. Our main result says that if a particular type of asymmetry is present in the extreme set $E(A, u)$ for any u, then one can generate a locator set by intersecting the set A with all translates of A by points in $E(A, u)$. The exact statement of this result is quite technical, so we prefer to state a corollary result that we have found to be very useful in practice:

> If the extreme set $E(A, u)$ consists of exactly two points, $\{a_1, a_2\}$, then a locator set for A is given by
>
> $$L = A \cap (A + a_1) \cap (A + a_2) \quad , \tag{3}$$

where for a point a_1, $(A + a_1) \equiv \{a + a_1 : a \in A\}$. In words, a locator set may be found by intersecting A with two copies of itself, each one translated to one of the extreme points. An example of a 3-D locator set generated by such a rule is given in Figure 2. The object, shown in (a), is a discretely sampled cone. The z direction is vertical and the apex of the cone is at the (1,1,1) point in the lower right portion of the figure [the rendition, which has different scales for each of the axes, makes this difficult to visualize; this was necessary to allow all the points to be clearly seen in part (d)]. The autocorrelation support set A is shown in (b) in a 3-D perspective shaded rendition and in (c) as a top-down projection view, along with the two points which comprise $E(A, u)$ for a u-direction. This u is in a direction that is tilted toward the top and into the page. A locator set generated by taking this triple intersection is shown by the gray and black points in (d). Here the gray points in the locator set correspond to the original object, and the black points correspond to extra locator-set points that are not in the original object. Although this locator set is not perfectly tight, as evidenced by the many black extra points, it clearly indicates the shape of the cone object.

3.2 New Locator Set Exploiting Opacity

The locator-set rules in the previous subsection do not explicitly exploit our *a priori* knowledge that the object is opaque. Opacity of the object implies that the object support is only one point thick along

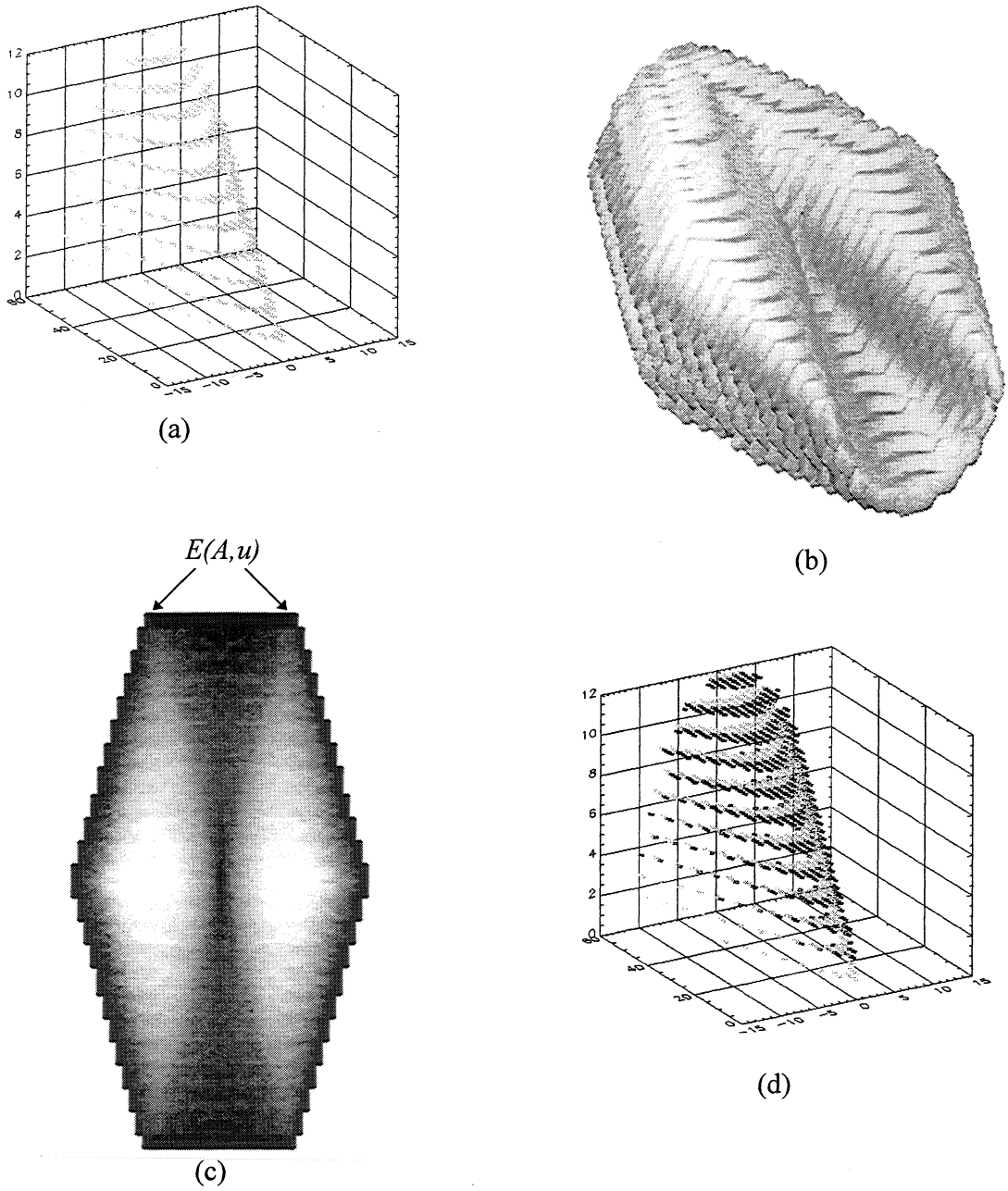

Figure 2. Example of Locator Set Generated from Autocorrelation Support. (a) Original discrete cone object; (b) view of 3-D autocorrelation support A of the cone object (the scalloped nature arises from the interaction of the 3-D rendering with the discrete grid of points on which the data is defined); (c) top-down view of the 3-D autocorrelation support A of the cone object with a shading proportional to depth, i.e., A is thin in dark regions and thick in light regions; (d) locator set obtained by intersecting autocorrelation support A with translates of A by the two points in $E(A,u)$, as shown in (c). The black points represent extra points not in the original object.

the line-of-sight direction, which we denote by the vector z. We call such an opaque object support, in which each point has a unique angle-angle projection relative to any other point in it, "z-thin." This *a priori* knowledge seems, intuitively, to be a potentially powerful constraint in helping identify improved locator sets. In this subsection, we present new rules which take advantage of this *a priori* knowledge about opacity. Our locator set rules based on object opacity all presuppose that one has some knowledge about the projection of the object support S to the 2-D space corresponding to the angle-angle projection, i.e., the 2-D space that is perpendicular to the line-of-sight. We denote this 2-D projection of the object support by S'. For the cone object shown in Figure 2, this 2-D projection S' is a triangle; a projection in a slightly different direction would give a sort of triangle with two straight edges and one curved edge. Our basic result is that object opacity along with the knowledge of object support projection, S', allows us to derive locator sets by intersecting A with all translates of A corresponding to points in $E' \subset E(A, u')$ provided that:

(i) u' is perpendicular to the line-of-sight vector z

(ii) the angle-angle projection of the 0 point and E' are all contained in some translate of S'

(iii) E' is equal to the set of all points in $E(A, u')$ which have a unique angle-angle projection (relative to $E(A, u')$).

In Figure 3, we show the application of this rule in simulation. The object, shown in (a), is the same discrete cone as in Figure 2. In (b), we show a projection of the discrete autocorrelation support A along with the designated points in $E(A, u')$. Here the direction u' varies slightly from the direction u in Figure 2, in that the extreme set $E(A, u')$ in Figure 2 consists of two points whereas the extreme set $E(A, u')$ in Figure 3 consists of a whole line of points corresponding to the edge. In this example, the set $E(A, u')$ also turns out to be a z-thin set so we are able to use the above rule to generate a locator set by intersecting A with all translates of A to points in $E' = E(A, u')$, with the resulting locator set shown in (c). Here, as in Figure 2, the gray points correspond to points in the original cone object, and the black points correspond to extra points not in the original object. Note the significant reduction in the "extra points" relative to the locator set given in Figure 2 (d), which was derived by intersecting only at the 2 corner points of $E(A, u')$. Though we have chosen not to show them, it turns out that we can also generate two more locator sets by intersecting over extreme points along the bottom right edge and the bottom left edge of the autocorrelation support displayed in (b). We also have another rule that allows us to take multiple locator sets and intersect them, under certain technical conditions, to generate a new locator set. These conditions are satisfied in this particular case, so the resultant intersection of all three locator sets is a new locator set, displayed in (d). Again the extra points are shown in black. Note that the number of extra points has been reduced relative to either of the previous locator sets.

3.3 Reducing Locator Sets

Once we arrive at a locator set, $L1$, for a given autocorrelation support, A, then we can employ the following procedure, which may reduce the size of $L1$, making it tighter and more useful. Let $r(n)$ be the n^{th} point in $L1$.

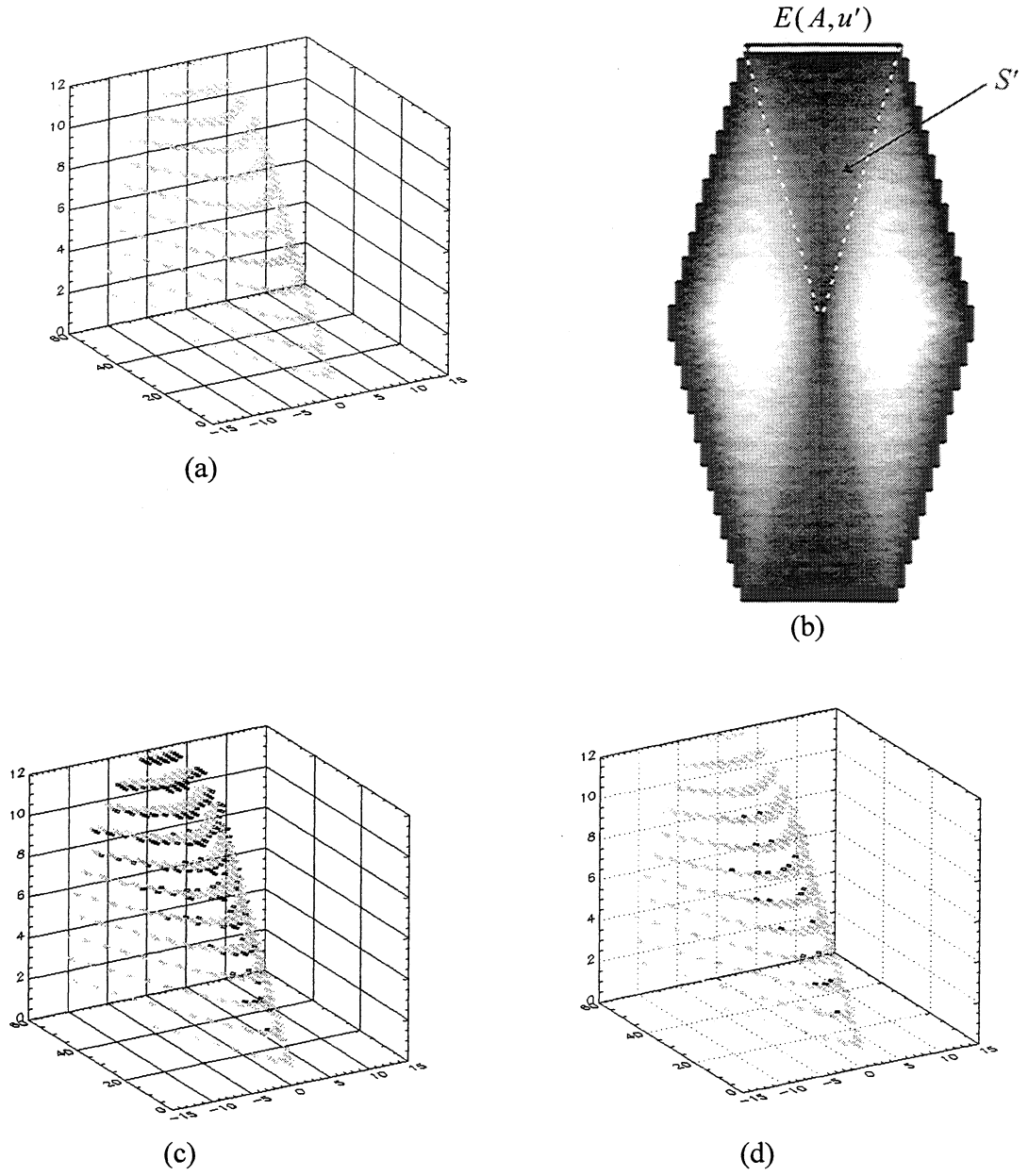

Figure 3. Results of Locator Sets Generated by the Use of the Opacity Constraint. (a) Original discrete cone object; (b) top-down projection of the autocorrelation with outline (in dashed lines) of the set S' representing the 2-D projection of the object support, along with the extreme set $E(A,u')$ which corresponds to the top edge of points in the projection (actually a curved edge in 3-D); (c) the locator set generated by intersecting the autocorrelation support with translates to all points in $E(A,u')$; (d) locator set generated by intersecting locator set in (c) with two other locator sets generated in a similar fashion as (c). Note the dramatic reduction in the "extra" black points.

1. Let $L2 = L1$ and $n = 1$.
2. If $r(n) \in L2$, then perform the following steps 3-5:
   3. Form $L_{trial} = L2 \setminus r(n)$, which is $L2$ with point $r(n)$ removed.
   4. Compute the autocorrelation support of L_{trial}: $A_{trial} = L_{trial} - L_{trial}$.
   5. If there is one or more points in A not included in A_{trial}, then L_{trial} is no longer a locator set for A, and so $r(n)$ is a "necessary point" that must be within $(S + v)$, for some translation v of the object support S; then replace $L2$ by a possibly new and smaller locator set

$$L2_{new} = L2 \cap (A + r(n)) \tag{4}$$

   (If all the points in A are also in A_{trial}, then make no changes.)

6. Increment n and repeat Steps 2 to 5 for all the points $r(n)$. This constitutes one cycle.
7. Perform additional cycles until no further changes occur in $L2$ after a complete cycle.

Figure 4 shows a 2-D example of reducing the size of a locator set in this manner. For the sparse object shown (which is not opaque), a triple intersection of its autocorrelation support yields a locator set $L1$ having several extra points that are not in a translate of the object support. After applying the reducing algorithm, the new locator set $L2$ became exactly the object support, a perfect result. For less sparse objects, the algorithm still works, but may leave some of the extra points. This algorithm works for all objects, whether opaque or not, in any number of dimensions.

4 CONCLUSIONS

We have developed new algorithms for determining locator sets and reducing the sizes of locator sets. These new locator sets are tighter (smaller) and hence provide more effective constraints for image reconstruction from Fourier-magnitude data. In some instances the locator sets may directly provide 3-D profile information as an end product.

ACKNOWLEDGMENTS

Supported by Ballistic Missile Defense Organization/ Innovative Science and Technology and managed by the Avionics Directorate of Wright Laboratory, Aeronautical Systems Center, USAF, Wright Patterson AFB OH 45433.

REFERENCES

1. R.G. Paxman, J.H. Seldin, J.R. Fienup, and J.C. Marron, "Use of an Opacity Constraint in Three-Dimensional Imaging," Proc. SPIE 2241-14, Inverse Optics III (April 1994), pp. 116-126.
2. R.G. Paxman, J.R. Fienup, J.H. Seldin and J.C. Marron, "Phase Retrieval with an Opacity Constraint," in Signal Recovery and Synthesis V, Vol. 11, 1995 OSA Technical Digest Series (Optical Society of America, Washington, DC, 1995), pp. 109-111.
3. M.F. Reiley, R.G. Paxman J.R. Fienup, J.M. Marron, K.W. Gleichman, and B.J. Thelen, "3-D Reconstruction of Opaque Objects from Fourier Intensity Data," Proc. SPIE 3170-09, Image Reconstruction and Restoration, T.J. Schulz, Chair, 1997 SPIE Symposium in San Diego, CA.
4. J.R. Fienup, "Phase Retrieval Algorithms: A Comparison," Appl. Opt. 21, 2758-2769 (1982).

5. J.R. Fienup, "Reconstruction of a Complex-Valued Object from the Modulus of Its Fourier Transform Using a Support Constraint," J. Opt. Soc. Am. A 4, 118-123 (1987).

6. J.R. Fienup and A.M. Kowalczyk, "Phase Retrieval for a Complex-Valued Object by Using a Low-Resolution Image," J. Opt. Soc. Am. A 7, 450-458 (1990).

7. H.C. Stankwitz, R.J. Dallaire, and J.R. Fienup, "Non-linear Apodization for Sidelobe Control in SAR Imagery," IEEE Trans. AES 31, 267-278 (1995).

8. T.R. Crimmins, "Geometric Filter for Speckle Reduction," Appl. Opt. 24, 1438-1443 (1985).

9. J.R. Fienup, T.R. Crimmins, and W. Holsztynski, "Reconstruction of the Support of an Object from the Support of Its Autocorrelation," J. Opt. Soc. Am. 72, 610-624 (May 1982).

10. T.R. Crimmins, J.R. Fienup and B.J. Thelen, "Improved Bounds on Object Support from Autocorrelation Support and Application to Phase Retrieval," J. Opt. Soc. Am. A 7, 3-13 (January 1990).

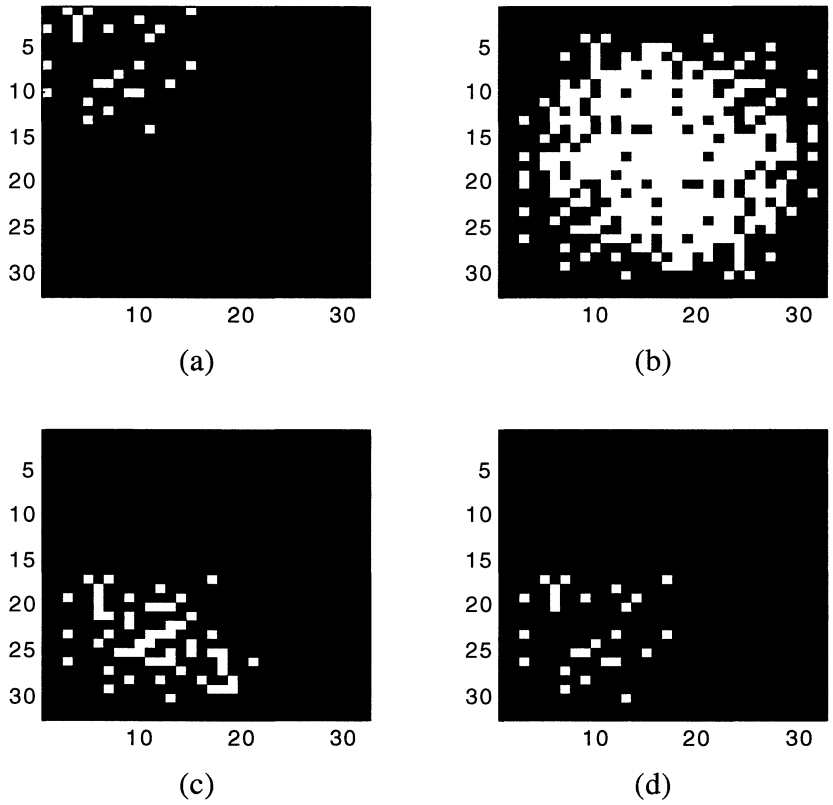

Figure 4. Demonstration of Reducing a Locator Set. (a) Object support, *S*; (b) autocorrelation support, *A*; (c) locator set, *L1*, computed from *A*; (d) reduced locator set, *L2*, computed from *L1* and *A*.