Shape from Equal Thickness Contours

G. Cong and B. Parvin
Information and Computing Sciences Division
Lawrence Berkeley National Laboratory
Berkeley, CA 94720

Abstract

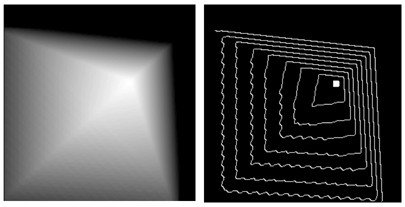
A unique imaging modality based on Equal Thickness Contours (ETC) has introduced a new opportunity for 3D shape reconstruction from multiple views. We present a computational framework for representing each view of an object in terms of its *object thickness*, and then integrating these representations into a *3D surface* by algebraic reconstruction. The object thickness is inferred by grouping curve segments that correspond to points of second derivative maxima. At each step of the process, we use some form of regularization to ensure closeness to the original features, as well as neighborhood continuity. We apply our approach to images of a sub-micron crystal structure obtained through a holographic process.

1 Introduction

The problem of shape-from-X has been a central research topic in the computer vision community. These include–but are not limited to–shape from shading [4, 11], texture [8], contour [14], color [2, 1], etc. These techniques have been applied from images obtained in controlled environments to natural outdoor scenes that may include more than one view. In this paper, we introduce a new imaging modality, and the corresponding method for shape recovery that has not yet been addressed by the computer vision community. This is based on equal thickness contour (ETC), which is obtained through a holographic process. One imaging source example is holographic electron microscopy of sub-micron crystal structures. Conventional electron microscopy presents projected images with little or no depth information. In contrast, electron holography–with coherent illumination–provides both magnitude and phase information that can be used to infer object thickness in terms of ETCs from each view of the sample. The holographic images contain interference fringes with spacings, in the best case down to less than an angstrom, where interference is between the transmitted and diffracted beams. Similar imaging technique includes satellite radar interferometry, where the phase difference between two radar returns–at two different times–from the same spatial location is used for change detection [9]. It has been demonstrated that minute geological changes (as a result of earthquake or movement in the earth's crust) can be recovered with this approach.

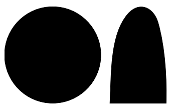
*This work is supported by the Director, Office of Energy Research, Office of Computation and Technology Research, Mathematical, Information, and Computational Sciences Division, and Office of Basic Energy Sciences, Material Sciences Division of the U. S. Department of Energy under contract No. DE-AC03-76SF00098 with the University of California. The LBNL publication number is 41678. E-mail: gcong@george.lbl.gov or parvin@george.lbl.gov

A simulation of ETCs for a synthetic object is shown in Figure 1. The main issue is that this mode of representation is inherently ambiguous since objects with completely different geometry can produce similar ETCs, as shown in Figure 2. Thus, multiple views of an object is essential for 3D shape recovery. In practice, however, these images may have low contrast, be noisy, contain artifacts, and may have shading; as a result, it is difficult to compute closed contours from these fringe patterns. Figure 3 shows three views of a real crystal structure that will be used for shape recovery. There is a small angle of rotation between different views, as reflected by the changes in the fringe patterns.



(a) (b)

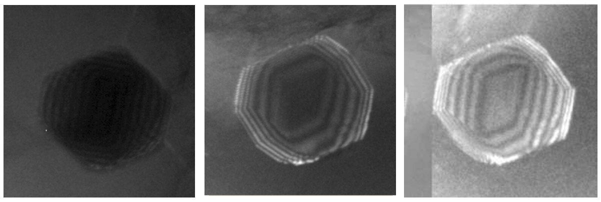
Figure 1: (a) Thickness of a synthetic pyramid and (b) the corresponding equal thickness contours.



(a) (b)

Figure 2: An object with these cross sections generates *approximately* the same ETC when viewed from the top.

Our method for shape recovery consists of five steps, some of which build on existing techniques developed in the computer vision community. The protocol for shape recovery is shown in Figure 4. Dominant features in these images are roof edges corresponding to crease lines; however, it is well-known that it is difficult to extract these features



(a) (b) (c)

Figure 3: Three views of a cubeoctahedral object with a diameter of 100 nano-meters.

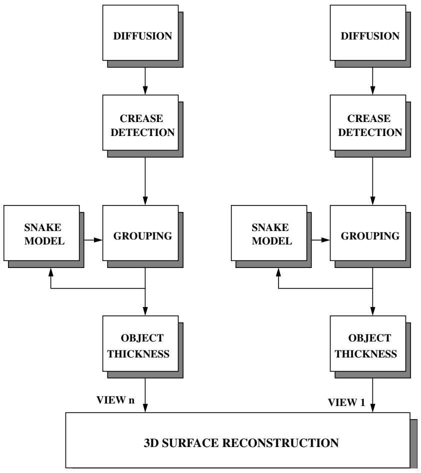


Figure 4: Protocol for recovering 3D shape from holographic images

directly in the presence of scale-change and noise. Here, we have adopted a stepwise refinement of images to extract a desirable representation. The first step of the process enhances peaks and valleys of the original data with adaptive smoothing. Next, crease points are extracted and grouped on the basis of collinearity and convexity. It is assumed that grouping does not produce closed contours, and an interactive snake is provided to establish the boundary conditions, based on innermost and outermost fringes[1]. The grouping strategy is very conservative to allow closure of small gaps, and at times it successfully provides closed contours. The contour representation (based on lines of second derivative maxima) provides the basis for obtaining object thickness at each point in the image. We have developed a regularized approach to computing object thickness, subject to continuity and smoothness constraints. The final step of our protocol is to integrate object thicknesses obtained from each viewing orientation into a 3D surface representation. The approach is based on a modification and extension of the algebraic reconstruction technique [3, 7], which is essentially a tomographic process.

In the rest of this paper, each step of this computational process is described with results from real data, followed by concluding remarks in Section 6.

2 Crease Detection

The original images are processed with a variation of adaptive smoothing [12] to enhance the roof edges corresponding to points of local maximum principal curvature. Here, each point in the image is made smoother as a function of its inverse local Laplacian. Hence, when the Laplacian is large, no smoothing is performed.

We have experimented with localization of creases for computation of local curvature maxima based on the first and second fundamental form, and with maxima computed from the second directional derivative. Although the results appear to be similar subjectively, we have used the second method due to its simplicity. The main rationale is that computation of curvature involves different orders of derivatives and a division operation.

The second derivative along any direction $\vec{n}$ can be computed as:

$$f_{nn} = f_{xx} cos^2\theta + 2 f_{xy} sin\theta cos\theta + f_{yy} sin^2\theta \tag{1}$$

where θ is the angle between x axis and $\vec{n}$. Specifically, the second derivative along the $\frac{\Pi}{4}$ direction, $\vec{p}$, is given by:

$$f_{pp} = \frac{1}{2} f_{xx} + f_{xy} + \frac{1}{2} f_{yy} \tag{2}$$

Thus,

$$f_{xy} = f_{pp} - \frac{1}{2}(f_{xx} + f_{yy}) \tag{3}$$

From (1), it is straightforward to see that f_{nn} is maximum at:

$$f_{nn} = \frac{f_{xx} + f_{yy} + \sqrt{2(f_{xx}^2 + f_{yy}^2) + 4 f_{pp}^2 - 4 f_{pp}(f_{xx} + f_{yy})}}{2} \tag{4}$$

along the direction:

$$\theta = \frac{1}{2}\arctan\left(\frac{2 f_{xy}}{f_{xx} - f_{yy}}\right) = \frac{1}{2}\arctan\left(\frac{2 f_{pp} - (f_{xx} + f_{yy})}{f_{xx} - f_{yy}}\right) \tag{5}$$

The main benefit of this technique for finding the creases is that only the second derivatives along three fixed directions are needed.

3 Grouping

At the completion of the previous stage, the crease points are linked to form curve segments that are then represented with a series of consecutive line segments. The next step of the computational process is to close the gaps as much as possible to improve the accuracy of shape recovery.

Grouping has been an active area of research in the computer vision community [6, 13], and we have based our grouping strategy on the work of many other researchers. In general, grouping can be based on collinearity, convexity, and symmetry. Symmetry is a strong constraint, not implemented here in order to evaluate the current constraints for shape recovery (the symmetry constraint will be between adjacent contours). There are several steps in the grouping process. These include neighborhood formation, choice of an appropriate cost function, and the search protocol. The neighborhood formation is based on triangulation to ensure that there will be no intersection among various hypotheses. The local cost function between two adjacent line segments is chosen to be:

[1] The emphasis is on having a robust system, so that the science project can proceed uninterrupted.

$$local\ cost = \frac{L_i + L_j}{G_{ij}} cos\theta \ \ Where\ \ \theta < \frac{\pi}{6} \cap G_{ij} < Thresh \quad (6)$$

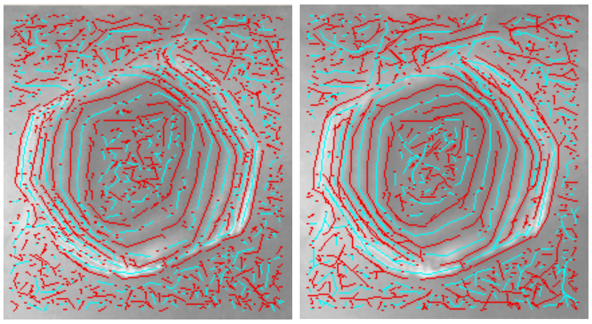
L_i, L_j, G_{ij}, and θ are length of line segment i, length of line segment j, length of gap, and the turning angle between two segments, respectively. The cost function –with hard small thresholds– is designed to be very conservative and to close gaps based on collinearity. The convexity constraint is a binary test applied during the search process. Actual search procedure is based on a special type of dynamic programming (DP) for efficiency. DP is essentially a sequential decision-making process. Let P be a set of states, D be a set of possible decisions, $F: P \times D \mapsto \mathcal{F}$ be a cost function, and $\psi: P \times D \mapsto P$ be a function that maps the current state and a decision into the next state. In a single step, the maximum possible value, starting from state p_i, is given by:

$$H_1(p_i) = \max_{d \in D} F(p_i, d) \quad (7)$$

By the same token, choosing a decision d that maximizes the value of a sequence for n states starting from p_i is found by:

$$H_n(p_i) = \max_{d \in D} [F(p_i, d) + H_{n-1}(\psi(p_i, d))] \quad (8)$$

The above recurrence relation, together with the coupling relation of Equation (6), specifies contour formation so that certain constraints are satisfied. In this formulation, the decision d corresponds to any of the candidate line segments that correspond to L_j in Equation (6). An example of the grouping with all the original noise in the image is shown in Figure 5. The grouping for this example takes less than a second; but the bulk of processing is in establishing the neighborhood relation among nearby line segments. We have augmented our system with a snake model (using second derivative features as external forces) to ensure that the outer and inner fringes remain closed at this point of the computational process. These two contours provide the boundary conditions for the next step of the process.



(a) (b)

Figure 5: Grouping results: (a) original; (b) closed gaps.

4 Reconstruction of object thickness

The goal of reconstructing object thickness is to calculate a 2D image in which each point corresponds to the object thickness in a certain direction. First, we will introduce a simple interpolation method between the boundary conditions that were derived earlier. This will serve as the initial condition for refinement based on local contour information.

4.1 A simple interpolation method

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ represent two 2D curves with C_2 inside of C_1. The problem is how to construct a surface $f(x, y)$, such that:

$$f(\mathcal{C}_1) = 0, \quad f(\mathcal{C}_2) = H \quad (9)$$

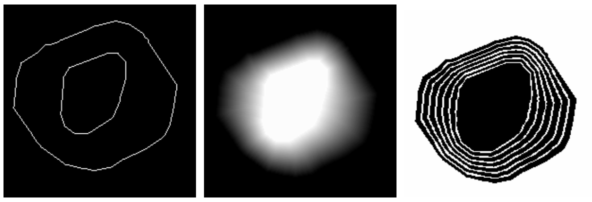
where H is a constant. It is well known that there are infinitely many surfaces that satisfy these constraints, and the literature in shape-from-contour has addressed this issue extensively [14]. However, we are only interested in a simple interpolation that will serve for initialization. Let $d_1(x, y)$ and $d_2(x, y)$ be the distance transformation (computed from chamfer images) of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively:

$$d_1(p = (x, y)) = Min_s\{\mathcal{D}(p, \mathcal{C}_1(s))\} \quad (10)$$
$$d_2(p = (x, y)) = Min_s\{\mathcal{D}(p, \mathcal{C}_2(s))\}$$

where $\mathcal{D}(p, \mathcal{C}(s))$ is the distance from point p to a point $\mathcal{C}(s)$ on the curve, then $f(x, y)$ can be constructed by:

$$f(x, y) = \frac{d_1(x, y) \cdot H + d_2(x, y) \cdot 0}{d_1(x, y) + d_2(x, y)} = \frac{d_1(x, y)}{d_1(x, y) + d_2(x, y)} H \quad (11)$$

An example is shown in Figure 6.



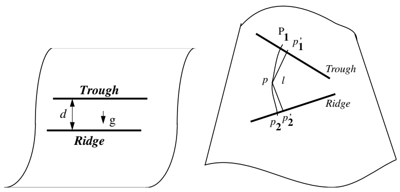
(a) (b) (c)

Figure 6: Simple interpolation: (a) The initial curves; (b) Interpolated surface, intensity indicates surface height; (c) Level sets of the interpolated surface.

4.2 Thickness gradient

In practice, it is very difficult to extract closed contours, which complicates the reconstruction process. However, local contour segments provide the means for partial correction and updating in a local surface neighborhood. Adjacent contour segments are either parallel or intersect at a small angle. Consider trough T and ridge R contour segments that are adjacent to one another, and assume that from T to R the object thickness decreases by a constant, say 1, on the surface. If T and R are parallel with d distance apart (see Figure 7(a)) and the thickness decreases linearly, then the magnitude of gradient for points between T and R is given by: $|| \bigtriangledown f|| = \frac{1}{d}$. On the other hand, if the two segments are not parallel–see Figure 7(b)– then for each point p, there should be a curve passing through p that intersects T and R at p_1 and p_2, respectively. The tangent of this curve is in the direction of thickness gradient, which could also be obtained by tracking the gradient direction from p_1 to p_2. Since R and T are level set segments, it is easy to show that the normal to these segments at any point and the gradient of object thickness are in the same direction. When R and T are close to each other and the intersection angle between them is small, we can approximate a curve l, passing through p, by two line segments $pp_1' \perp T, pp_2' \perp R$, where $p_1'pp_2'$ is an approximation to p_1pp_2. Hence, $l \approx |p_1'p| + |p_2'p|$. These line segments are estimated from distance transforms (chamfer images[10]), $d_t(x,y)$ and $d_r(x,y)$ that are computed from trough and ridge segments. Thus, $d_t = |p_1'p|$ and $d_r = |p_2'p|$, and the magnitude of the gradient is given by:

$$|| \bigtriangledown f(x,y)|| = \frac{1}{d_r(x,y) + d_t(x,y)} \tag{12}$$



(a) (b)

Figure 7: Measure of gradient estimates from local contour segments: (a) T and R are parallel; (b) T and$R$ are not parallel.

4.3 Regularized object thickness construction

The gradient feature, computed in the previous section, is noisy and scale-limited. We have developed a regularized approach for constructing the object thickness $f(x,y)$. We define a suitable energy function to reflect the closeness to data and a smoothing term:

$$\epsilon^2 = \int\int (f_x - p)^2 + (f_y - q)^2 + \lambda(\triangle f)^2 dx dy \tag{13}$$

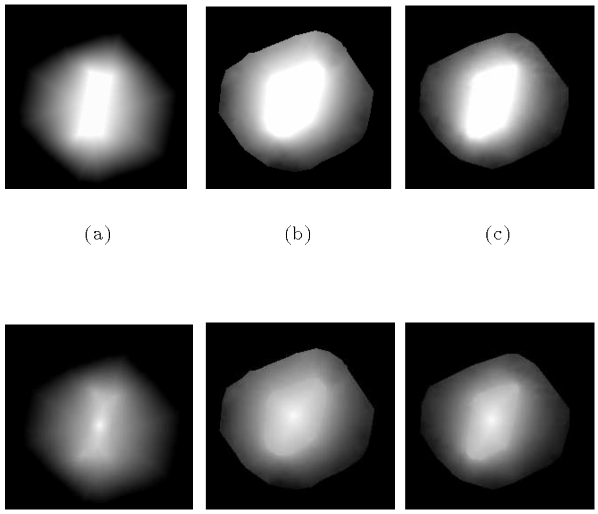
where f_x, f_y denote the x and y derivative of f; and p, q are the x and y direction gradient information computed earlier. The smoothness is enforced by the Laplacian of the reconstructed object thickness $\triangle f$. This is physically a natural constraint since the object is constructed from facets. Following the Euler equation, the surface f that minimizes (13) is:

$$(f_{xx} - p_x) + (f_{yy} - q_y) \ - \ \lambda \triangle (\triangle f) = 0 \tag{14}$$

This equation is solved iteratively [5] from an initial guess corresponding to the interpolation derived from the boundary conditions:

$$\begin{aligned} f^{k+1}(i,j) \quad &= \quad -(f^k_{xx} - p_x) - (f^k_{yy} - q_y) \\ &+ \frac{\lambda}{4}(\triangle f^k(i-1,j) + \triangle f^k(i,j-1) \\ &+ \triangle f^k(i+1,j) + \triangle f^k(i,j+1)) \\ &- \frac{\lambda}{4}(f^k(i-1,j) + f^k(i,j-1) \\ &+ f^k(i+1,j) + f^k(i,j+1)) \end{aligned} \tag{15}$$

There is a certain ambiguity in filling up the region inside the inner contour as given by the initial condition. This ambiguity originates from the choice of planar fit or further interpolation from the inner contour to the center of the mass. This is a higher-level process and can only be resolved at the multiple-view integration. Figure 8 shows the reconstructed object thickness for each view of the object shown in Figure 3.



(a) (b) (c)

(d) (e) (f)

Figure 8: Reconstructed thickness from the ETC in Figure 3. (a,b,c) planar fit; (d,e,f) interpolating inside inner contour.

5 3D shape recovery

In the preceding section, we developed an approach to recover the object thickness at each point along a certain direction. This is the thickness between front and back surfaces. In this section, we show how to integrate object thickness representation from different views into a 3D object. Our method extends the algebraic reconstruction techniques [3, 7]. However, we have a small number of views–three in our case–and the angles between these views are on the order of 7-8 degrees. The first step in 3D reconstruction is to align the object thickness representation bases in their location at center of mass. The 3D object boundary is then initialized as a polytope, which corresponds to the intersection of thickness extremes in 3D space. The 3D surface is then iteratively refined, based on object thickness information from each view. For simplicity, we present the details of our solution in 2D. Extension to 3D will be given in Appendix A. Let

- $g(i,j)$ be the 2D object image. $g(i,j), i = 0, ..., I, j = 0, ..., J$, is a discrete image that represents a binary object A and the background $\bar{A}$:

$$g(i,j) = \begin{cases} 1, & \text{if } (i,j) \in A \\ 0, & \text{if } (i,j) \in \bar{A} \end{cases} \tag{16}$$

- $\mathcal{P}(\theta, k)$ be the computed thickness from angle θ at a point k as discussed in previous section,

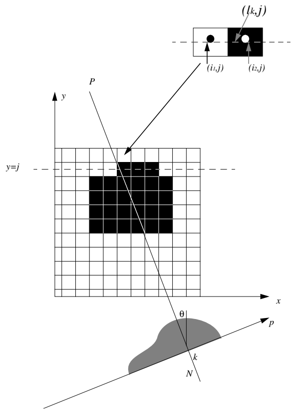
- $p(\theta, k), -\frac{\pi}{4} \le \theta \le \frac{\pi}{4}$ be the thickness computed from $g(i, j)$. This thickness (as shown in Figure 9) is given by:

$$p(\theta, k) = \sum_{j=0}^{J} \frac{g(l_k, j)}{cos(\theta)} \tag{17}$$

  where (l_k, j) is the intersection point of line PN and the horizontal line $y = j$, and $g(l_k, j)$ is just a linear interpolation of the point around (l_k, j):

$$g(l_k, j) = g(i_1, j) * (i_2 - l_k) + g(i_2, j) * (l_k - i_1) \tag{18}$$

  Thus, $p(\theta, k)$ is a linear combination of image pixels $g(i, j), i = 0, ..., I, j = 0, ..., J$.



(a)

Figure 9: Projection of a 2D binary image on a line.

- $b(A), b(\bar{A})$ be the boundary points. The boundary point of the object is defined as:

$$\begin{aligned} b(A) := \{(i, j) | g(i, j) \cap [\bar{g}(i-1, j) \cup \bar{g}(i, j-1) \\ \cup \bar{g}(i+1, j) \cup \bar{g}(i, j+1)] = 1\} \end{aligned} \tag{19}$$

  where $\cap$ and $\cup$ correspond to the logical *and* and *or* operations. $\bar{g}(i, j)$ is the inverse of $g(i, j)$.
  Similarly, the boundary point of the background is defined as:

$$\begin{aligned} b(\bar{A}) := \{(i, j) | \bar{g}(i, j) \cap [g(i-1, j) \cup g(i, j-1) \\ \cup g(i+1, j) \cup g(i, j+1)] = 1\} \end{aligned} \tag{20}$$

- $\hat{p} \in b(\bar{A})$ of point $p \in b(A)$ be the neighborhood points, where $\hat{p}$ is defined over a 3-by-3 neighborhood of p in the background and is in the normal direction of the object's occluding contour at point p.

Assume that we have N thicknesses of a 2D object A along N directions $\mathcal{P}(\theta_n, k), n = 0, ..., N$, each thickness has M_n points. To reconstruct an $I \times J$ image $g(i, j)$ for the corresponding thicknesses, we define a cost function:

$$\epsilon^2 = \sum_{n=0}^{N} \sum_{m=0}^{M_n} (p(\theta_n, m) - \mathcal{P}(\theta_n, m))^2 + \lambda \sum_{(i,j) \in b(A)} (\mathcal{K}(i, j))^2 \tag{21}$$

where $\mathcal{K}$ is the product of the curvature of the boundary conditions and $| \bigtriangledown g |^3$. Since, using the curvature would have complicated the optimization process. $\mathcal{K}$ is computed from the binary images as:

$$\mathcal{K} = g_x g_{yy} - 2 g_x g_y g_{xy} + g_y g_{xx} \tag{22}$$

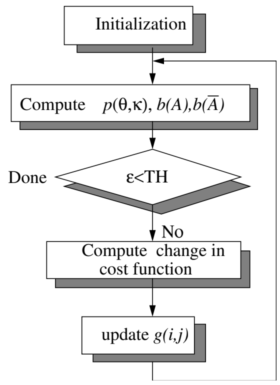
where,

$$\begin{aligned} g_x(i, j) &= g(i, j) - g(i-1, j) \\ g_y(i, j) &= g(i, j) - g(i, j-1) \\ g_{xx}(i, j) &= \frac{1}{2}(g(i-1, j) + g(i+1, j)) - g(i, j) \\ g_{yy}(i, j) &= \frac{1}{2}(g(i, j-1) + g(i, j+1)) - g(i, j) \\ g_{xy}(i, j) &= \frac{1}{2}(g(i, j) + g(i-1, j-1) \\ &\quad - g(i-1, j) - g(i, j-1)) \end{aligned} \tag{23}$$

The $\mathcal{K}$ term of the boundary point serves as a penalty term for smoothness, and the cost function is minimized with the gradient search scheme. There are two major differences with the ART method. First, we do not convert the 2D image into a 1D array vector, and second, we only calculate the derivative of ϵ with respect to points in $b(A)$ and $b(\bar{A})$. These points are updated according to the following criteria:

1. If $\frac{\partial \epsilon}{\partial g(p)} > 0$ and $\frac{\partial \epsilon}{\partial g(\bar{p})} > 0$, change $g(p)$ from 1 to 0.
2. If $\frac{\partial \epsilon}{\partial g(\bar{p})} < 0$ and $\frac{\partial \epsilon}{\partial g(\bar{p})} < 0$, change $g(\bar{p})$ from 0 to 1.

The reconstruction protocol is shown in Figure 10. The results of shape recovery for a synthetic and real object (of Figure 3) are shown in Figures 11 and 12, respectively.



(a)

Figure 10: Protocol for shape evolution

6 Conclusion

In this paper, we introduced an imaging mode, in terms of equal thickness contours, that is being investigated for scientific studies. Next, we introduced a novel approach for shape recovery from multiple views and applied our approach to interference images of a sub-micron crystal structure. Our approach consists of three primary steps. These include 1) image representation in terms of contours

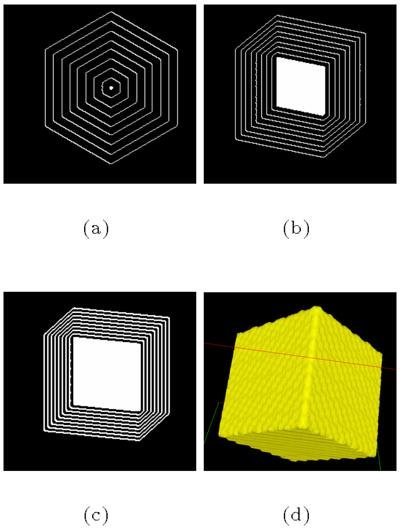


Figure 11: 3D shape reconstruction for a synthetic cube: (a,b,c) ETCs; (d) reconstructed shape.

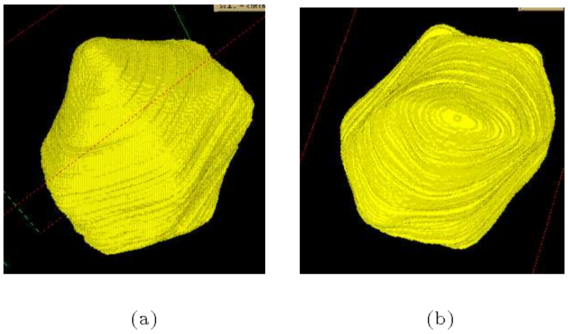


Figure 12: 3D shape recovery from three views shown in Figure 3: (a) View1; (b) View2

corresponding to creases, 2) object thickness representation from contour data, and 3) shape recovery from object thickness obtained from each view. We are currently evaluating the accuracy of our approach with respect to a model system with objective of determining the number of views required to achieve a certain accuracy. Furthermore, since each view can produce up to two possible object thicknesses (depending upon how the inner region is filled in) we are exploring various ways to select among these hypotheses either during image acquisition or shape recovery. In addition, we plan to apply our method to other interference patterns obtained from natural outdoor scenes.

A 3D algebraic reconstruction

The 3D extension of the algebraic reconstruction of Section 5 is accomplished by replacing $g(i,j)$ of Figure 9 with a cube $g(i,j,l)$, and the 1D projection $p(\theta,k)$ by a 2D projection $p(\vec{\theta},k_1,k_2)$, where $\vec{\theta}=(\theta_1,\theta_2,\theta_3)$, and θ_1, θ_2, and θ_3 correspond to the angle between the projection direction with the x, y, and z axes. The projection can now be computed by:

$$p(\vec{\theta},k_1,k_2)=\sum_{l=0}^{L}\frac{g(s_{k_1},s_{k_2},l)}{cos(\theta_3)} \qquad (24)$$

where (s_{k_1},s_{k_2},l) is the point of intersection between the direction and plane $z=l$ and $-\frac{\pi}{4}\leq\theta_3\leq\frac{\pi}{4}$. $g(s_{k_1},s_{k_2},l)$ is a bilinear interpolation in 2-by-2 neighborhood, and the projection is a linear combination of the image $g(i,j,l)$. The corresponding energy function is defined as:

$$\epsilon^2=\sum_{n=0}^{N}\sum_{m_1=0}^{M_{1n}}\sum_{m_2=0}^{M_{2n}}\left(p(\vec{\theta}_n,m_1,m_2)-\mathcal{P}(\vec{\theta}_n,m_1,m_2)\right)^2 + \lambda\sum_{(i,j,l)\in b(A)}(\mathcal{K}(i,j,l))^2 \qquad (25)$$

where N is the number of projections of size M_{1n}-by-M_{2n} points, and $\mathcal{K}$ is the mean curvature of the boundary surface.

Acknowledgements: The authors thank Drs. Ulrich Dahmen and Eric Johnson of the National Center for Electron Microscopy for providing data and introducing us to this mode of imaging. We thank Mr. Wes Bethel of the Visualization Laboratory for his help and patience in using AVS. Finally, we thank Mr. Johnston –of the Information and Computing Sciences Division– for his encouragement in initiating this research project.

References

[1] P. Christensen and L. Shapiro. Three-dimensional shape from color photometric stereo. *International Journal of Computer Vision*, 13:213–227, 1994.

[2] B. Funt and M. Drew. Color space analysis of mutual illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1319–1326, 1993.

[3] R. Gordon and G. Herman. Reconstruction of picture from their projections. *Communications of the ACM*, 14:759–768, 1971.

[4] B. Horn. Height and gradient from shading. In *Proceedings of the DARPA Image Understanding Workshop*, pages 584–595, 1989.

[5] B.K.P. Horn and M.J. Brooks. The variational approach to shape from shading. *Computer Vision, Graphics, and Image Processing*, 33(2):174–208, February 1986.

[6] D. Jacobs. Robust and efficient detection of convex groups. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):23–37, 1995.

[7] R. Kashyap and M. Mittal. Picture reconstruction for projections. *IEEE Transactions on Computers*, 24:915–923, 1975.

[8] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *International Journal of Computer Vision*, 23:149–168, 1997.

[9] D. Massonnet. Satellite radar interferometry. *Scientific American*, 276(2):46–53, February 1997.

[10] P.F.M. Nacken. Chamfer metrics, the medial axis and mathematical morphology. *Journal of Mathematical Imaging and Vision*, 6:235–248, 1996.

[11] A. Pentland. Linear shape from shading. *International Journal of Computer Vision*, 4:153–162, 1990.

[12] P. Saint-Marc, Chen J., and G. Medioni. Adaptive smoothing: A general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):514–530, 1991.

[13] A. Sha'ashua and S. Ullman. Structural saliency: the detection of globally salient structures using a locally connected network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 321–327, Tampa, FL, 1988.

[14] F. Ulupinar and R. Nevatia. Perception of 3-d surface from 2-d contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:3–18, 1993.