Request Date: 11-JAN-2010 14:50:43
Expiry Date: 18-JAN-2010
ILL Number: 3627071
TQ: 16100192 **TGQ or OCLC#:** 3627057
ID: OCLC:CUI
Call Number: UCSB:Main Lib T385 .C558 Sciences Engineering Library
ISSN: 0271-4159
Material Type: Serial
Title: Computer Graphics World
Article Author: Liu, Mei-chi M
Article Title: A Shadow Algorithm for Hyperspace
Part Pub. Date: 1984-07
Pages: 51-59
Publisher: PennWell Publishing Corp.
Requester: UCI Science Library
Address: Document Access Delivery, Fax: 949-824-3695, email:sldad@uci.edu
Del. Address: Document Access Delivery, Fax: 949-824-3695, email:sldad@uci.edu
Patron ID: 50689 **Patron Name:** Black,Donald Vaughn (Graduate [])
Patron Barcode: 21970004704598
Patron Status: GRADU
Service Level: Normal - Full Search
Service Type: Copy non returnable
Delivery Method: Electronic Mail
Max Cost: 50
Notes:

Sent	Message	Note
From :UI3	Request	Digital Copy Preferred (PDF)

Verification Source: MELVYL-UCLinks-sfx:citation

Best copy - tightly bound
Pages 53, 55 + 57 not included, as they are advertisements only.

- UCSB

vill do much t
eloping remot
ssing industry

unities will b
e and softwar
nd market pr
ng equipmen
l similar satel
that the com
kept open fo
(e.g., softwar
tour levels t
e into a map
in the next fiv
tools and tech
oduced to solv
ng and remot
introduced b
of commercia
llites. ■

esearch analyst a
monton, Alberta. H
ote sensing and im
ch capacity. Mr. Mi
ophysics and a BA

at 1:2,000,000-scale.
eographic sec-
terminous U.S.,
or Hawaii.

ll or write:
ical Survey,
2,

ata

A Shadow Algorithm for Hyperspace

Calculating Shadows in Hyperdimensional Scenes

By Mei-chi Liu, Robert P. Burton and Douglas M. Campbell

This paper describes an algorithm for calculating shadows in hyperdimensional scenes. The scenes consist of multiple convex objects with dimensions that may be as great as those of the scene itself, together with multiple light sources. Shadows are calculated and added to the scene, which is subsequently projected to lower dimensions and presented. The development of a shadow algorithm for hyperspace is part of an ongoing effort to develop computer graphics techniques for meaningfully presenting hyperdimensional models which occur whenever four or more variables exist simultaneously. The utility of such models is often enhanced by visual rather than numerical representation.

Related efforts include: the development of a hidden-line algorithm and of stereo motion picture capabilities with hidden lines removed from hyperobjects; a careful study of depth cues and their application to the presentation of hyperobjects; holograms of objects in hyperspace; research to develop a hidden-volume algorithm for hyperspace, to categorize and present hypothesized four-dimensional phenomena, and to develop a constructive solid geometry scheme for presenting multi-dimensional graphic information.

Shadows cast and received by hyperdimensional objects constitute a cue that remains unexploited. When shadows are included in a three-dimensional scene, they reveal information about the relative positions of objects. By including shadows in a hyperdimensional scene, information about the relative positions of hyperobjects can be revealed.

As different points in hyperspace may project onto the same point in two-dimensional space, existing shadow algorithms that use scanning hidden-surface techniques to determine shadows in three-dimensional space do not lend themselves to extension to hyperspaces. Therefore, the shadow algorithm described here carries out all calculations in the original n-dimensional space before the scene is projected for presentation. Once shadows are determined, any projection can be selected to present the scene.

From Lower- to Higher-Dimensional Objects

Higher-dimensional objects can be built from lower-dimensional objects. For example, a line segment is defined by two bounding points, a polygon by its bounding line segments, and a polyhedron by its bounding polygons. The surface of an n-dimensional object is defined by its bounding (n − 1)-dimensional objects.

Given a surface portion of an n-dimensional object X and a light source L for n-space, a test is needed to determine whether the surface portion is illuminated. The surface portion is illuminated only if the normal to the surface portion forms an angle of less than 90 degrees with a vector to the light source. The advantage of defining the surface of an n-dimensional object with (n − 1)-dimensional objects is that the illumination of an n-dimensional object can be determined simply from the direction of each

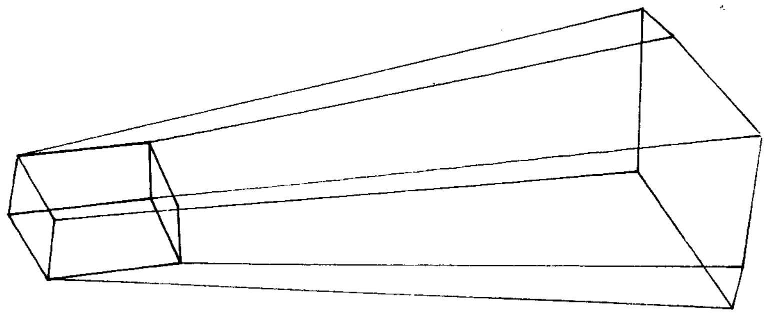

Figure 1: The shadow areas of a polyhedron.

normal of its (n – 1)-dimensional surfaces. Only the illuminated (n – 1)-dimensional surfaces need to be processed by the algorithm, thereby decreasing the computation time.

Description of the Algorithm

The intersecting shadow volume algorithm is an object-space algorithm. It accepts geometrical and topological descriptions of multiple convex objects and the positions of multiple light sources. The viewer is restricted to the near side of all objects which in turn are restricted to the near side of a background plane whose dimension is one less than the dimension of the scene. Shadows are calculated and added to the scene which is then projected for presentation.

In the accompanying figures, parts of the scene are presented in various colors: black for objects; red for the shadow volume of an object formed by a single light source; blue for the shadow volume of a second light source; and green for the intersection of the illuminated portion of the surface of an object and the shadow volume of another object (i.e., the shadow cast upon the illuminated portion of the first object).

Two Three-Dimensional Objects X and Y and One Light Source L: The algorithm is applied in three steps to determine shadows.

☐ *Step 1:* Use normals to separate the surface polygons of a polyhedron into those illuminated by L and those that are not. The illuminated surface of a polyhedron is defined as consisting of all its illuminated surface polygons. The shadow volume of the polyhedron from the light source is generated

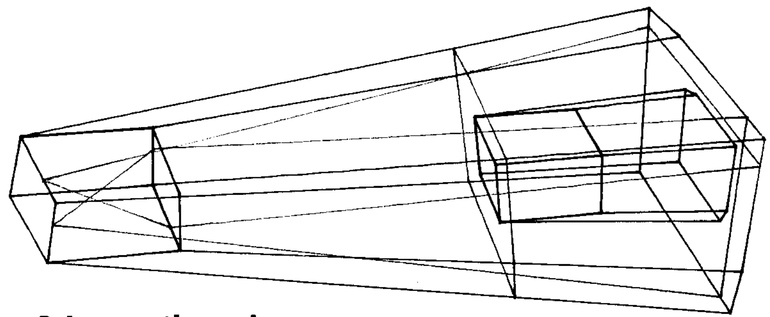

Figure 2: Intersecting polygons attached to the extended surface polygons.

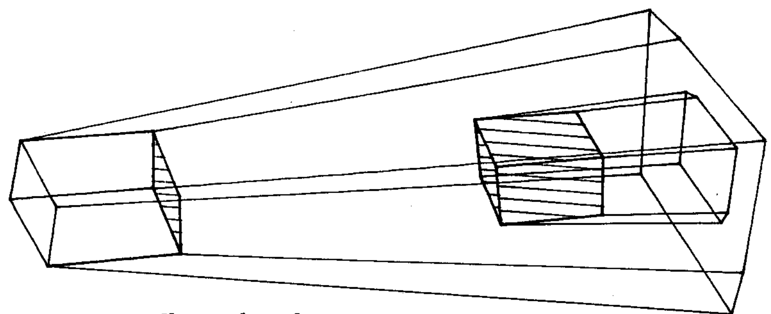

Figure 3: Three-dimensional shadows.

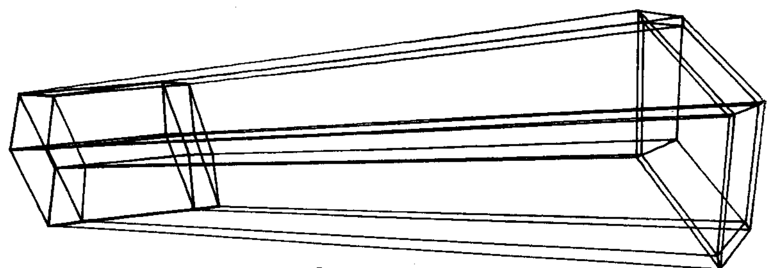

Figure 4: The shadow areas of hyperpolyhedra.

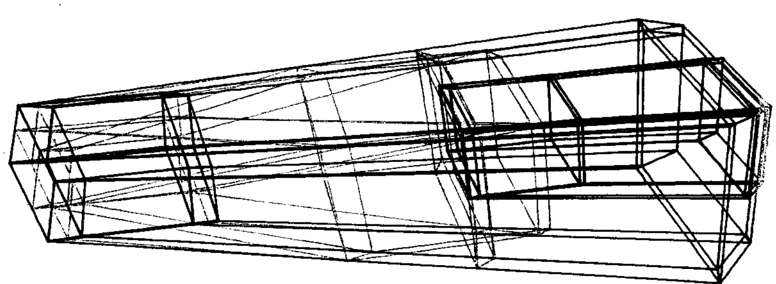

Figure 5: Intersecting polyhedra attached to the extended surface polyhedra.

by projecting the illuminated surface from the polyhedron toward the background plane. The shadow volume is itself a three-dimensional polyhedron bounded on the ends by the illuminated surface and the projection of the illuminated surface onto the background plane *(Figure 1)*. This is computationally equivalent to projecting each illuminated surface polygon onto the background plane.

☐ *Step 2:* Intersect the shadow volume of polyhedron Y with each illuminated surface polygon of polyhedron X. If the intersection is nonempty, the intersection is a shadow polygon, line, or point and is added to the list of shadows in the scene. Repeat, interchanging the rôles of polyhedra X and Y *(Figures 2* and *3)*.

☐ *Step 3:* Project and present the illuminated surfaces and shadows (polygons, lines, and points).

Two Four-Dimensional Objects X and Y and One Light Source L: The algorithm is extended to determine shadows in four-dimensional space in three steps.

☐ *Step 1:* Use normals to separate the surface polyhedra of the hyperpolyhedron into those illuminated by L and those that are not. The illuminated surface of a hyperpolyhedron is defined as consisting of all its illuminated surface polyhedra. The shadow hypervolume of the hyperpolyhedron from the light source is generated by projecting the illuminated surface from the hyperpolyhedron toward the background hyperplane. The shadow hypervolume is itself a four-dimensional hyperpolyhedron bounded on the ends by the illuminated surface and the projection of the illuminated surface onto the back-

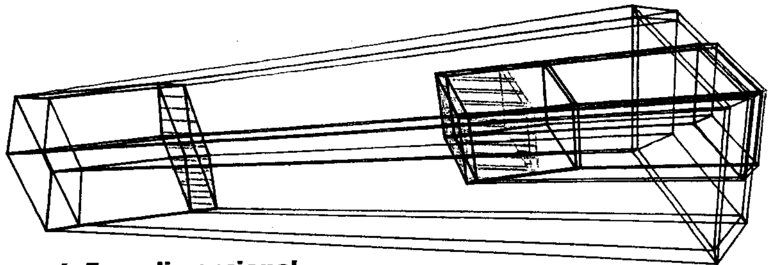

Figure 6: Four-dimensional shadows.

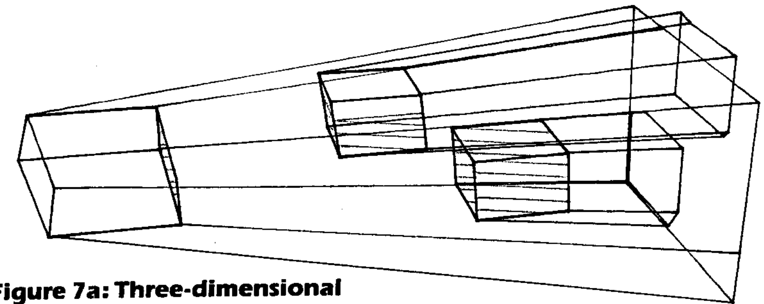

Figure 7a: Three-dimensional shadows with multiple objects.

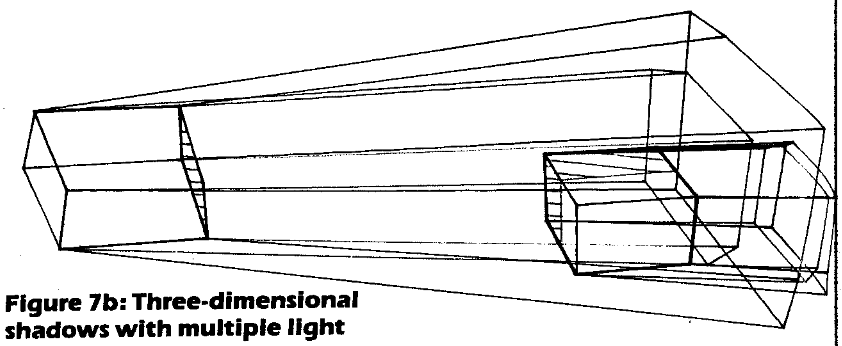

Figure 7b: Three-dimensional shadows with multiple light sources.

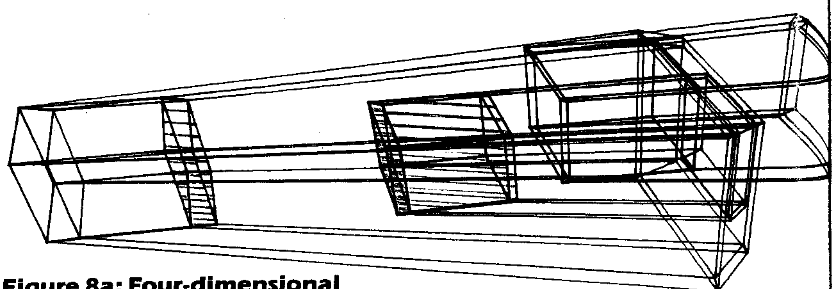

Figure 8a: Four-dimensional shadows with multiple objects.

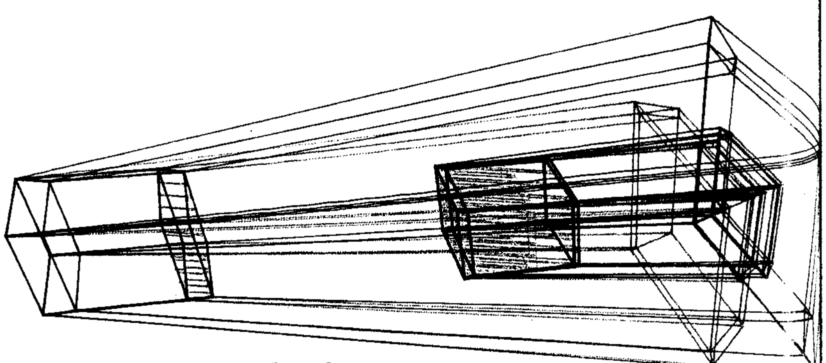

Figure 8b: Four-dimensional shadows with multiple light sources.

ground hyperplane *(Figure 4)*. This is computationally equivalent to projecting each illuminated surface polyhedron onto the background hyperplane.

☐ *Step 2:* Intersect the shadow hypervolume of a hyperpolyhedron Y with each illuminated surface polyhedron of hyperpolyhedron X. If the intersection is nonempty, the intersection is a shadow entity of dimension less than four and is added to the list of shadows in the scene. Repeat, interchanging the roles of hyperpolyhedra X and Y *(Figures 5* and *6)*.

☐ *Step 3:* Project and present the illuminated surfaces and shadows (polyhedra, polygons, lines, and points).

Two n-Dimensional Objects X and Y and One Light Source L: The algorithm is applied to determine shadows in n-dimensional space in three steps.

☐ *Step 1:* Use normals to separate the $(n-1)$-dimensional surface elements of the n-dimensional convex object into those illuminated by L and those that are not. The illuminated $(n-1)$- dimensional surface of the object is the union of its illuminated $(n-1)$-dimensional surface elements. The shadow hypervolume of the object from the light source is generated by projecting the illuminated surface of the object toward the background hyperplane. The shadow hypervolume is itself an n-dimensional hyperpolyhedron bounded on the ends by the illuminated surface and the projection of the illuminated surface onto the background hyperplane.

☐ *Step 2:* Intersect the n-dimensional shadow hypervolume of the object Y with each illuminated surface element of the object X. If the intersection is nonempty, the intersection is a shadow entity and is added to the list of shadows in the scene. Since the shadow hypervolume is of dimension n and the illuminated surface element is of dimension $n-1$, the dimension of a shadow entity is at most $n-1$. Repeat, interchanging the roles of the n-dimensional objects X and Y.

☐ *Step 3:* Project and present the illuminated surfaces and shadows (which are at most $(n-1)$-dimensional).

Multiple Objects and Light Sources: If there are m n-dimensional objects $X_1, X_2, \ldots, X_m$ in the scene, with $V_1, V_2, \ldots, V_m$ illuminated surface elements respectively, then Step 2 of the algorithm is repeated for each illuminated surface hyperpolyhedron for each of the $m-1$ n-dimensional shadow volumes. Assuming the average number of visible surface polyhedra to be V, the computational time of the algorithm is increased by a factor of $(m-1)V$.

If there are k light sources $L_1, \ldots, L_k$, then Step 2 of the algorithm must be repeated k times. This increases the computation time of the algorithm by a factor of k.

Assuming k light sources and n objects with the same average number of illuminated surfaces, the total computation time—including transformations, hidden-surface elimination, calculation of shadow volumes, intersections and shadows cast upon objects—is $kn(n-1)t_av$, where t_av is the average computation time to calculate shadows cast upon one object from another.

Running on a VAX 11/750, the computation time to calculate typical three- and four-dimensional shadows is listed in *Table 1*.

Conclusion

The shadow algorithm presented here has been successfully implemented to determine shadows in three- and four-dimensional spaces. Multiple objects *(Figures 7a* and *8a)* and multiple light sources *(Fig-*

Determining Object Surface Illumination

An n-dimensional convex object Y is defined by a finite set of linear equations $E_1,\ldots,E_m$ as follows. A point $X + (x_1,\ldots x_n)$ of n-space is strictly in Y only if $E_i(x)*0$ for $i = 1,\ldots,m$, (i.e., if the point x is strictly "inside" every hyperplane which defines Y). A point is inside, or on, Y only if $E_i(x)\dagger 0$ for $i = 1,\ldots,m$.

The convex object Y defined by $E_1,\ldots,E_m$ has m $(n-1)$-dimensional boundary surfaces, each of which lies in one of the m hyperplanes $E_i(x) = 0$. If E_i is the linear equation that defines the i-th $(n-1)$-dimensional surface element, then the element is illuminated by a light source at the point X only if $E_i(x)>0$ (i.e., if X is "outside" the boundary hyperplane used to define object Y).

A hyperplane E is defined by first choosing n points $p_1,\ldots,p_n$ which do not lie in $(n-2)$-dimensional space. Each point p_i can be written in terms of its n coordinates as $p_i = (p_{i1},\ldots,p_in)$, where $p_i j$ denotes the projection of p_i on the j-th axis. The $n \times n$ matrix $M = (m_{ij})$ is found next where

$$X_j - P_{1j} \quad \text{if } i = 1,\ 1 \leqslant j \leqslant n$$
$$m_{ij} = \quad P_{ij} - P_{1j} \text{ if } 2 \leqslant i \leqslant n,\ 1 \leqslant j \leqslant n$$

The linear equation E is the determinant of the matrix M. It can be given in the ordinary linear form

$$E(x) = C_i x_i - C_i P_1 i,$$

where C_i is the cofactor of the element in the first row and i-th column of the matrix M. ■

Calculation of the Point(s) of Intersection of a Line and an m-dimensional Subspace in n-space, m<n

To determine what parts of an illuminated (n − 1)-dimensional surface lie in an n-dimensional shadow volume, it is necessary to break down the illuminated surface into its line elements and then determine the intersection of the line elements of the illuminated surface and the (n − 1)-dimensional boundaries of the shadow volume. Thus, the problem of the intersection of a line and an m-dimensional space where $m<n$ needs to be solved.

A line L (a one-dimensional space) is defined by two points 1_1 and 1_2 where $(1_{i1}, 1_{i2},\ldots,1_{in})$ are the n-space coordinates of the point 1_i. A point $x=(x_1,\ldots,x_n)$ is on L only if $x_i = t(1_{1i} - 1_{2i}) + 1_{2i}$ for some real number t and for all i, $1\leq i\leq n$.

An m-dimensional space S is defined by m + 1 points $s_1,\ldots,s_m+_1$. By an appropriate change of coordinates, it is assumed, without loss of generality, that the 1-dimensional space L and the m-dimensional space S lie in a space spanned by the first (m + 1)-coordinate axis of n-space. Since S is confined to the (m + 1)-dimensional subspace in the first (m + 1) space coordinates, the space S may be given as the solution of $S(x)=0$ where

(1) $S(x) = B_j(x_j - s_{1j})$,

where B_j is the determinant of the $m \times m$ matrix

$$B_j = \begin{matrix} b_{11} & \ldots & b_1 j-1 \; b_1 j+1 & \ldots & b_1 m+1 \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ b_{i1} & \ldots & b_i j-1 \; b_i j+1 & \ldots & b_i m+1 \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ b_{m1} & \ldots & b_m j-1 \; b_m j+1 & \ldots & b_m m+1 \end{matrix}$$

and $b_{ji} = s_j+1\, i - s_{1i}$.

Since L is confined to an (m + 1)-dimensional space in the first (m + 1) space coordinates, the previous remark can be refined to state that x belongs to L only if $x_j=t(1_{1j}-1_{2j})+1_{2j}$ for some real number t for all j, $1\leq j\leq m+1$ and $x_j=1_{2j}=1_{1j}$ for all j, $m+2\leq j\leq n$.

Let $A_j=1_{1j}-1_{2j}$, $1\leq j\leq m+1$. Since $1_1\neq 1_2$, there is an index, say i, such that $A_i\neq 0$. The equation $x_j=tA_j+1_{2j}$ is multiplied by A_i, and the equation $x_i=tA_i+1_{2i}$ is multiplied by A_j and then subtracted. Then

$A_ix_i - A_ix_j = A_j1_{2i} - A_i1_{2j}\, f_j$,

which can be rewritten as

$x_j = (A_jx_i - f_j)/A_i$.

Thus, a point $x = (x_1,\ldots,x_n)$ belongs to L only if

$$\begin{aligned} x_j &= (A_jx_i - f_j)/A_i && 1\leq j\leq i\text{-}1 \\ x_i &= tA_i + 1_{2i} && j=i \\ & && (2) \\ x_j &= (A_jx_i - f_j)/A_i && i+1\leq j\leq m+1 \\ x_j &= 1_{2j} = 1_{1j} && m+1\leq j\leq n. \end{aligned}$$

Assume that L and S intersect. Let x be a point both on L and in S. Since x is on L, x satisfies (2). Since x is in S, $S(x)=0$. Using (1),

(3) $B_j(x_j - s_{1j}) = 0$.

Then (3) is rewritten as

(4) $B_jx_j = B_js_{1j}$

and (2) is substituted in (4):

(5) $B_ix_i + B_j(A_jx_i - f_j)/A_i = B_js_{1j}$.

Multiplying by A_i, equation (5) becomes

(6) $(\;\; B_jA_j)x_i = A_i \;\; B_js_{1j} + B_jf_j$.

If $B_jA_j=0$, then there is no restriction on x_i and the entire line L in in the subspace. If $B_jA_j\neq 0$, then the point x_i is uniquely defined by

(7) $x_i = [A_i \;\; B_js_{1j} + B_jf_j]/(\;\; B_jA_j)$.

Since x_i is uniquely defined by (7), then all the other coordinates of x are uniquely defined by (2). In this case, the point of intersection has been computed. ■

ures 7b and *8b)* have been includ in scenes. The objects used in thr dimensional scenes are cubes; t objects used in four-dimension scenes are hypercubes.

When the shadows in a four-d mensional scene overlap each oth it has been difficult to tell whe each shadow is actually located. a means of increasing the realis of the scene, depth cues to achie realism in a three-dimension scene should be considered. The might include rotation, hidden-el ment elimination, interaction a stereoscopy. It is still difficult predict that a profound feeling f hyperspace will be gained. Neve theless, the development of a shad algorithm for hyperspace repr sents one more step toward th threshold beyond which an intui tive feeling for hyperdimensiona scenes will be realized.

Some of the numerical algo rithms needed for the implemen tation of the shadow algorith accompany this article. A complet description of the numerical algo rithms is available in Mei-chi Liu' Master's Thesis (see References).

References

Abbott, E. A. (A. Square). *Flatland: A Romance of Many Dimensions*. 2nd ed. London: Seeley & Co., 1884.

Armstrong, W. P. *Hyperdimensional Graphics: The Computer Display of Objects in Hyperspace*. Master's thesis. Brigham Young University 1981.

Atherton, P., K. Weiler and D. Greenberg. "Polygon Shadow Generation." *Computer Graphics*, 12(3):275 (August 1978).

Bragdon, C. *A Primer of Higher Space: The Fourth Dimension*. Tucson, Arizona: Omen Press, 1972. (Originally published in Rochester in 1913).

Burton, R. P. and D. R. Smith. "A Hidden-line Algorithm for Hyperspace." *SIAM*

Journal on Computing, Vol. 11, no. 1 (February 1982) 71-80.

Burton, R. P. and B. F. Webster. "Some Thoughts on Higher-dimensional Realms." *BYU Studies* (Spring 1980) 281-96.

Crow, F. C. "Shadow Algorithms for Computer Graphics." *Computer Graphics*, 11(2):242 (Summer 1977).

Lindgren, C., S. Ernesto and S. M. Slaby. *Four Dimensional Descriptive Geometry*. New York: McGraw-Hill, 1968.

Liu, M. *A Four Dimensional Shadow Algorithm*. Master's Thesis. Brigham Young University 1982.

Manning, H. P. *The Fourth Dimension Simply Explained*. New York: Dover Publications, 1960 ed. (originally 1910).

Newman, W. M. and R. F. Sproull. *Principles of Interactive Graphics*. New York: McGraw-Hill, 1979.

Noll, A. M. "Computer Techniques for Displaying n-Dimensional Hyperobjects." *Communications of the ACM*, 10:469-73 1967.

Reeder, R. D. *The Presentation of Hyperdimensional Scenes: A Visual Cue Approach*. Master's thesis. Brigham Young University 1980.

Mei-chi Liu is a programmer/analyst at Eyring Research Institute. She received an M.S. in Computer Science from Brigham Young University in 1982.

Robert Burton is a Professor of Computer Science at Brigham Young University. He received a Ph.D. in Computer Science from the University of Utah in 1973.

Douglas Campbell is a Professor of Computer Science at Brigham Young University. He received a Ph.D. in Mathematics from the University of North Carolina in 1971.