A Computer Technique for Displaying n-Dimensional Hyperobjects

A. Michael Noll
Bell Telephone Laboratories, Murray Hill, N. J.

A digital computer and automatic plotter have been used to generate three-dimensional stereoscopic movies of the three-dimensional parallel and perspective projections of four-dimensional hyperobjects rotating in four-dimensional space. The observed projections and their motions were a direct extension of three-dimensional experience, but no profound "feeling" or insight into the fourth spatial dimension was obtained. The technique can be generalized to n-dimensions and applied to any n-dimensional hyperobject or hypersurface.

Introduction

In his now classic book on Flatland, Edwin Abbott [1] describes the social order resulting in a world restricted to two spatial dimensions. The inhabitants of this world are completely unable to visualize a third spatial dimension and are therefore thoroughly baffled by the weird distortions of the two-dimensional projections into their world of simple three-dimensional objects. Man finds himself in a similar state of puzzlement concerning spatial dimensions higher than three, and really never even knows whether he might be witnessing the three-dimensional projection of some higher-dimensional event. When man does not comprehend he sometimes gives religious significance, and therefore not surprisingly a fifth dimension has even been proposed as "the ultimate spiritual essence" [8].

The mathematics and projective geometry of three-dimensional space can be generalized to any number of dimensions so that an n-dimensional hyperobject can be mathematically projected into an $(n-1)$-dimensional space. Such projection could be applied repetitively until finally a three-dimensional object representing the successive projections of an n-dimensional hyperobject is obtained. If desired, the hyperobject might move in n-dimensional space so that its three-dimensional projection would not be stationary. A relatively simple form of motion is rotation of the hyperobject in n-dimensional space.

This paper is a review of the mathematics for two types of projection of n-dimensional hyperobjects and for n-dimensional rotation. Any n-dimensional hyperobject could then be manipulated mathematically by a digital computer. The final three-dimensional projection of the rotating hyperobject could be drawn automatically on a computer-controlled visual display device as a stereoscopic movie.

As an example of this technique, a computer technique for generating three-dimensional movies of the perspective projections into three-dimensional space of four-dimensional hyperobjects is described. Now, like the inhabitants of Flatland, we too are puzzled by the strange distortions of the projection into three-dimensional space of a rotating but rigid four-dimensional hyperobject. Using intuition to extend to four dimensions our knowledge of the type of distortions resulting from three-dimensional perspective projection, it is possible to explain the distortions but still impossible to visualize the rigid four-dimensional hyperobject.

Rotation

Throughout this paper, bold-face lower-case letters will represent column vectors while matrices will be represented by bold-face capital letters. The column vector $\mathbf{x} = (x_1, x_2, \cdots, x_n)^t$ represents any point in n-dimensional space where the superscript t indicates transposition.

In n-dimensional space the simplest rotation is in a two-dimensional plane. If rotation is in the plane of x_a and x_b (the a-b plane), then the rotation matrix $\mathbf{R}_{ab}(\alpha)$ has the elements

$$\begin{aligned} r_{ii} &= 1 \quad \text{except } r_{aa} = r_{bb} = \cos\alpha, \\ r_{ij} &= 0 \quad \text{except } r_{ab} = -r_{ba} = -\sin\alpha. \end{aligned} \tag{1}$$

For example, the rotation matrix for a rotation through an angle α in the 2-4 plane in five-dimensional space is

$$\mathbf{R}_{24}(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2}$$

The n-dimensional rotation specified by eq. (1) is called a two-dimensional plane rotation since only those coordinates of a vector in the two-dimensional plane determined by the axes x_a and x_b are changed. Thus, in three-dimensional space, rotation about the x_3 axis would be called rotation in the x_1-x_2 plane. In spaces of higher than three dimensions, rotation about an axis is meaningless in terms of eq. (1) since a multitude of nonparallel two-dimensional planes are all perpendicular to the same axis. For example, in four-dimensional space the x_1-x_2, x_1-x_3, and x_2-x_3 planes are all perpendicular to the x_4-axis.

Any n-dimensional rotation matrix can be written as the product of $n(n-1)/2$ two-dimensional-plane n-dimensional rotation matrices [5]. Thus, e.g., in four-dimensional space

$$\mathbf{R} = \mathbf{R}_{12}(\alpha_1)\mathbf{R}_{13}(\alpha_2)\mathbf{R}_{14}(\alpha_3)\mathbf{R}_{23}(\alpha_4)\mathbf{R}_{24}(\alpha_5)\mathbf{R}_{34}(\alpha_6). \quad (3)$$

Projection

The most common form of projection in three-dimensional space is the perspective projection of an object as seen by each of our two eyes. A perspective projection is produced graphically, as shown in Figure 1, by first choosing a viewing point from which to view the object. A two-dimensional plane is then placed between the object and the viewing point. Straight lines are drawn from the object to the viewing point; their intersections with the plane are the perspective projection of the object. This procedure can be extended as follows to n-dimensional objects.

As shown in Figure 2, the point $\mathbf{p} = (X_1, X_2, \cdots, X_n)^t$ in the n-dimensional space with axes $x_1, x_2, \cdots, x_{n-1}, x_n$ is to be perspectively projected onto the $(n-1)$-dimensional hyperplane defined by $x_n = F$. For simplicity, the viewing point $\mathbf{v}$ is located a distance R along the x_n-axis. A straight line is drawn from $\mathbf{p}$ to $\mathbf{v}$, and its intersection $\mathbf{p}'$ with the $(n-1)$-dimensional hyperplane is the desired perspective projection:

$$\mathbf{p}' = \begin{pmatrix} X_1' \\ X_2' \\ \vdots \\ X_{n-1}' \\ X_n' \end{pmatrix} = \begin{pmatrix} \dfrac{(R-F)X_1}{R-X_n} \\ \dfrac{(R-F)X_2}{R-X_n} \\ \vdots \\ \dfrac{(R-F)X_{n-1}}{R-X_n} \\ F \end{pmatrix}. \quad (4)$$

The nth coordinate of $\mathbf{p}'$ is constant since $\mathbf{p}'$ lies in the $(n-1)$-dimensional hyperplane. Accordingly, $\mathbf{p}'$ can also be represented as a $(n-1)$-dimensional vector mathematically derived as the perspective projection of its counterpart in the n-dimensional space:

$$\mathbf{p}'_{n-1} = \begin{pmatrix} \dfrac{(R-F)X_1}{R-X_n} \\ \dfrac{(R-F)X_2}{R-X_n} \\ \vdots \\ \dfrac{(R-F)X_{n-1}}{R-X_n} \end{pmatrix}. \quad (5)$$

Another type of projection is derived from perspective projection by choosing the viewing point at infinity, i.e., $\mathbf{v} = (0, 0, \cdots, \infty)^t$. Since the projection lines are all parallel in the limit, this is commonly called parallel projection. By taking the limit of eq. (4) as $R \to \infty$, the parallel projection $\mathbf{q}'$ of the n-dimensional point $\mathbf{p} = (X_1, X_2, \cdots, X_n)^t$ is

$$\mathbf{q}' = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_{n-1} \\ F \end{pmatrix}. \quad (6)$$

Thus, the parallel projection is identical with the original point in the first $(n-1)$ dimensions.

Hyperobjects

A hyperobject in n-dimensional space can be represented as straight line segments connecting an ordered set of points (m in number). Similarly, an n-dimensional hypersurface can be depicted visually as a finite set of points randomly scattered over its surface. In either case, the n-dimensional hyperobject or hypersurface can be specified as a set of n-dimensional vectors which, if desired, might be combined together as the columns of a matrix. Thus, an

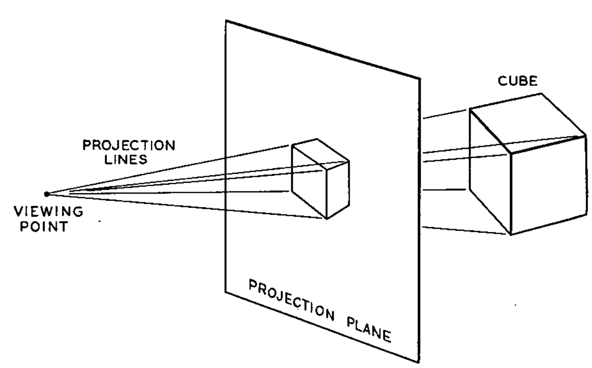

Fig. 1. Perspective projection of a three-dimensional cube onto a two-dimensional plane

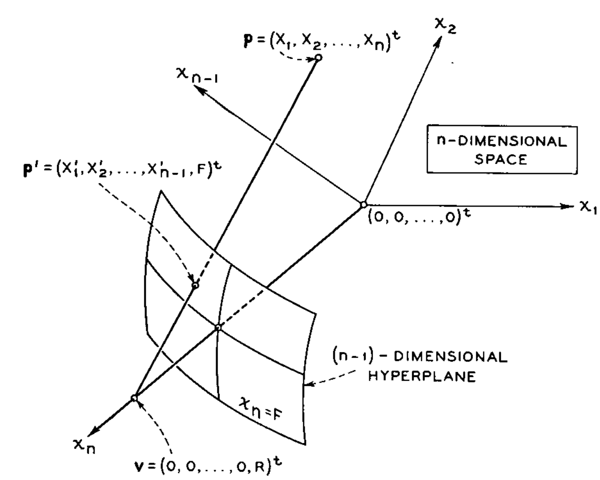

Fig. 2. Perspective projection of a point in n-dimensional space onto an $(n-1)$-dimensional hyperplane

n-dimensional hyperobject or hypersurface can be represented as an $n \times m$ matrix **H** given by

$$H = \begin{bmatrix} Y_1(1) & Y_1(2) & \cdots & Y_1(m) \\ Y_2(1) & Y_2(2) & \cdots & Y_2(m) \\ \vdots & \vdots & & \vdots \\ Y_n(1) & Y_n(2) & \cdots & Y_n(m) \end{bmatrix}. \tag{7}$$

The mathematical restriction on the Y's or the algorithm used in calculating them determines the hyperobject or hypersurface represented by the matrix **H**. For example, if

$$\sum_{i=1}^{n} (Y_i(j) - C_i)^2 = \rho^2 \tag{8}$$

for all $j = 1, \cdots, m$, then the points all lie on the surface of an n-dimensional hypersphere with center at $(C_1, C_2, \cdots, C_n)^t$.

If the line representation of a hyperobject with disjoint portions is desired, then the disjoint portions of the hyperobject must be specified so as not to be connected together with straight lines. Also, even if the hyperobject is not disjoint, certain line segments might have to be treated disjointly if the restriction is imposed that no line be drawn twice. For example, the 12 edges of a three-dimensional cube can not be drawn as a connected line without drawing some edges more than once.

Computer Technique

Since our habitation is restricted to a maximum of three spatial dimensions, we are unable to visualize a fourth much less a higher spatial dimension. We are able to perceive three-dimensional depth as a result of the slightly different images seen by our eyes. The illusion of depth can be created by viewing stereoscopically a pair of perspective two-dimensional pictures, but such perspectives are very tedious to calculate and draw. However, computer techniques are presently available for calculating and automatically plotting the left eye and right-eye images of some three-dimensional object [6]. If desired, a three-dimensional movie can be generated in this manner using the computer and automatic plotter to generate a sequence of pictures. Thus, if an n-spatial-dimensional object is mathematically projected onto three dimensions, the computer can produce the required drawings to obtain a three-dimensional depth effect. A movie can be produced by simply choosing to rotate the hyperobject in n-dimensional space. Although this procedure is generally applicable to n-dimensions, the details that follow will describe the actual implementation to four-dimensional hyperobjects and hypersurfaces.

The computer program performs the following functions. First, the object matrix **H** specifying the desired hyperobject is read into the computer from punched cards. The hyperobject is then rotated in four-dimensional space to a new orientation with object matrix

$$\mathbf{Y} = \mathbf{R}_{ab}(\alpha)\mathbf{H} \tag{9}$$

for a single plane rotation or by

$$\mathbf{Y} = \mathbf{R}_{ab}(\alpha)\mathbf{R}_{cd}(\beta)\mathbf{R}_{ef}(\gamma)\mathbf{H} \tag{10}$$

for a succession of three plane rotations. The rotated hyperobject **Y** is projected into three-dimensional space either by perspective projection given by

$$\begin{bmatrix} X_1(1) & \cdots & X_1(m) \\ X_2(1) & \cdots & X_2(m) \\ X_3(1) & \cdots & X_3(m) \end{bmatrix} = (R - F) \begin{bmatrix} \frac{Y_1(1)}{R - Y_4(1)} & \cdots & \frac{Y_1(m)}{R - Y_4(m)} \\ \frac{Y_2(1)}{R - Y_4(1)} & \cdots & \frac{Y_2(m)}{R - Y_4(m)} \\ \frac{Y_3(1)}{R - Y_4(1)} & \cdots & \frac{Y_3(m)}{R - Y_4(m)} \end{bmatrix} \tag{11}$$

or by parallel projection given by

$$\begin{bmatrix} X_1(1) & \cdots & X_1(m) \\ X_2(1) & \cdots & X_2(m) \\ X_3(1) & \cdots & X_3(m) \end{bmatrix} = \begin{bmatrix} Y_1(1) & \cdots & Y_1(m) \\ Y_2(1) & \cdots & Y_2(m) \\ Y_3(1) & \cdots & Y_3(m) \end{bmatrix} \tag{12}$$

The stereoscopic pair of two-dimensional perspective projections of the three-dimensional projection of the hyperobject are calculated by the computer and automatically plotted on a single frame of film. The hyperobject is then rotated through an incremental angle, and the various projections from four dimensions to three dimensions and from three dimensions to a pair of two-dimensional projections are repeated finally resulting in yet another frame of the movie.

Examples

The hypercube is the n-dimensional generalization of either a two-dimensional square or a three-dimensional cube. It is bounded by pairs of parallel $(n-1)$-dimensional hyperplanes which are all the same distance apart. The three-dimensional cube is bounded by three pairs of two-dimensional faces or squares while the four-dimensional hypercube is bounded by four pairs of three-dimensional hyperfaces which now are cubes. The n-dimensional hypercube has 2^n vertices and $n \cdot 2^{n-1}$ edges so that a four-dimensional hypercube has 16 vertices and 32 edges.

The 16 vertices specifying a four-dimensional hypercube were calculated and ordered into 33 points which when connected sequentially by straight lines would produce the hypercube's 32 edges. These points formed the object-matrix specification of the hypercube, and the computer then produced three-dimensional movies of the parallel and perspective projections of the hypercube rotating in four-dimensional space. Selected frames from the movie of the perspective projection are shown in Figure 3.

The perspective projection of a four-dimensional hypercube is a cube within a cube with corresponding vertices connected together. This and the motion caused by rotation can better be understood by analogy with a three-dimensional cube. The perspective projection of a cube is a square within a square with pairs of edges connected together, because the face, here a square, closest to the viewing point will appear largest. Similarly, the hypercube's hyperface (now a cube) closest to the viewing point will appear largest so that a cube within a cube is obtained.

If a cube rotates in a two-dimensional plane perpendicular to a line passing through both the origin and the viewing point, the perspective projection simply rotates as a whole. If, however, this line does not pass through the viewing point, then the position of the faces changes so that the projections of the faces change their size as the cube rotates. The four-dimensional extension of this is that the three-dimensional cube within a cube similarly rotates as a whole if the line passing through both the origin and the viewing point is perpendicular to the rotation plane. However, when this line is not perpendicular to the rotation plane, the cubes change their size as they rotate so that the hyperface (cube) closest to the viewing point is always largest. In the actual movie, the viewing point is situated on the x_4-axis, and three different matrix transformations are used for the rotations: (1) three complete revolutions in the 1-3 plane, i.e., $\mathbf{R}_{13}(\alpha)$; (2) three complete revolutions in the 2-4 plane, i.e., $\mathbf{R}_{24}(\alpha)$, and (3) three successive matrix transformations, i.e., $\mathbf{R}_{23}(\alpha)\mathbf{R}_{13}(\beta)\mathbf{R}_{34}(\gamma)$.

For the parallel projection from four dimensions to three dimensions there are no perspective distortions, and therefore the nearest and farthest hyperfaces are both the same size. Thus, the parallel projection of the hypercube is two cubes joined together to produce a cuboid. As the hypercube rotates, no perspective distortions occur as a result of the projection from four dimensions to three dimensions. A few selected frames from the movie are shown in Figure 4.

Three-dimensional movies of the perspective projection of a four-dimensional simplex, hypertetrahedron, and hypersphere were also generated by the computer. The hypersurface of the hypersphere was specified by randomly scattering points on its surface which were plotted as dots by the computer in the final movie. The points were scattered so as to have a uniform distribution over the surface of the hypersphere.

A five-dimensional hypercube was projected perspectively from five dimensions to four dimensions, and the four-dimensional projection was projected perspectively to three dimensions. However, the final three-dimensional projection, which appeared as a cube-within-a-cube within a cube-within-a-cube was extremely complicated so that the distortions resulting from the rotation were very difficult to follow. Thus, four dimensions would seem to be a

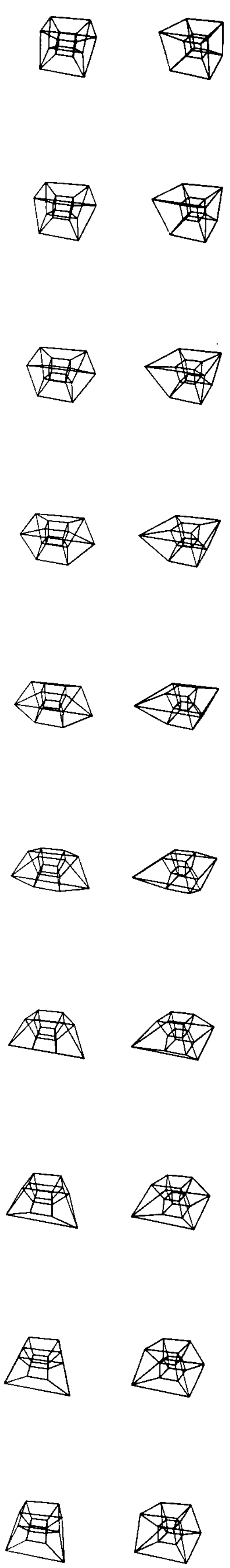

FIG. 3

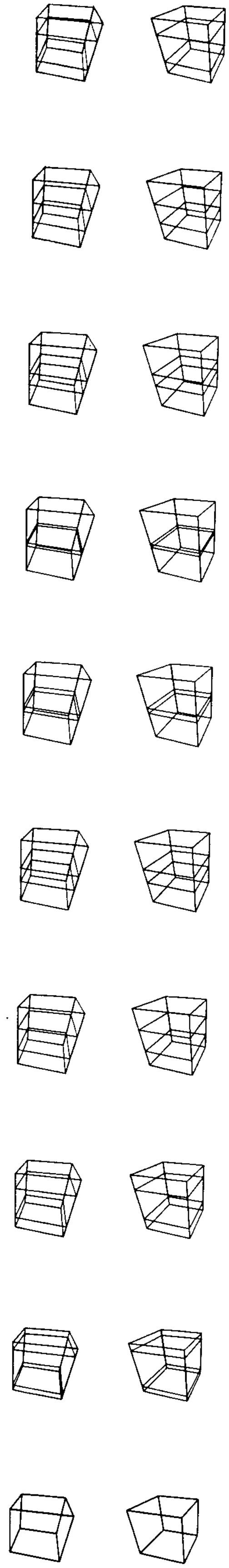

FIG. 4

FIG. 3. Selected frames from a computer-generated three-dimensional movie showing the three-dimensional perspective projection of a four-dimensional hypercube. The hypercube is being viewed from the x_4-axis and is rotating in the x_1 - x_3 plane. To view this figure in 3-D, place a sheet of paper on edge between one stereo pair. Position your head so each eye sees only one image. The pictures should then seem to merge and appear three dimensional

FIG. 4. The three-dimensional parallel projection of a four-dimensional hypercube

practical limit since higher-dimensional objects are presently too detailed to be displayed adequately by the computer.

Discussion

At first it was thought that the computer-generated movies of the four-dimensional hyperobjects might result in some "feeling" or insight for the visualization of a fourth spatial dimension. In particular, perhaps some visualization of a solid four-dimensional hyperobject would be gained from the distortions in the three-dimensional perspective projection. Unfortunately, this did not happen, and we are still as puzzled as the inhabitants of Flatland in attempting to visualize a higher spatial dimension.

However, the importance of the techniques presented in this paper is the use of a digital computer to generate visual displays of the three-dimensional projections of the hyperobjects. Such displays of rotating hyperobjects could be produced most efficiently by a computer since the projections and drawing would be too tedious and impractical to produce by any other method. Although no actual mental visualization of the fourth dimension resulted from the computer-generated displays, it was at least possible to visually display the projections and be puzzled in attempting to imagine the rigid four-dimensional hyperobject. Of course, these techniques should be useful in displaying data with more than three variables.

The movies have already been useful in extending knowledge of three-dimensional perspective projections to higher dimensions. The techniques have been applied to real-time graphical displays so that the user can rotate, translate, and manipulate hyperobjects and hyperdata and immediately see the results on a graphical display.

Acknowledgments. Grateful acknowledgment is made to Dr. D. E. Eastwood, Dr. M. V. Mathews, Dr. M. R. Schroeder, and Dr. M. M. Sondhi for their lively discussions, enthusiasm, and mathematical assistance in the multidimensioned aspects of this hyperdimensional project.

RECEIVED FEBRUARY 1967; REVISED APRIL, 1967

REFERENCES

1. ABBOTT, EDWIN A., *Flatland.* Dover Publications, Inc., New York, 1952.
2. BOERNER, HERMANN. *Representations of Groups.* North-Holland Publishing Co., Amsterdam, 1963.
3. COXETER, H. S. M. *Regular Polytopes.* The Macmillan Co., New York, 1963.
4. KENDALL, M. G. *A Course In the Geometry of n Dimensions.* Hafner Publishing Co., New York, 1961.
5. MURNAGHAN, FRANCIS D. *The Unitary and Rotation Groups.* Spartan Books, Washington, D. C., 1962.
6. NOLL, A. MICHAEL. Computer-generated three-dimensional movies. *Comput. Autom. 14*, 11 (Nov. 1965), 20-23.
7. SOMMERVILLE, D. M. Y. *An Introduction to the Geometry of N Dimensions.* Dover Publications, Inc., New York, 1958.
8. STROMBERG, GUSTAF. Space, time, and eternity. *J. Franklin Inst. 272*, 2 (Aug. 1961), 134-144.

LETTERS—Continued from p. 463

without passing any other label, and let $S_{i,j}$ be the sequence of assignment statements obeyed on this path. Define new Boolean variables $B_0, B_1, \cdots, B_n$. Then P is equivalent to the program:

$START$: $B_0 \leftarrow$ **true**; $B_1 \leftarrow$ **false**; $\cdots$; $B_n \rightarrow$ **false**;
L: **if** B_n **then go to** $EXIT$;
if $B_0 \wedge P_{0,0}$ **then begin** B_0 := **false**; $S_{0,0}$;
B_0 := **true end else**
...
...
if $B_i \wedge P_{i,j}$ **then begin** B_i := **false**; $S_{i,j}$;
B_j := **true end else**
...
...
if $B_{n-1} \wedge P_{n-1,n}$ **then begin** B_{n-1} := **false**; $S_{n-1,n}$;
B_n := **true end**;
go to L;
$EXIT$:

This program has a trivial flowchart of the form indicated above.

Böhm and Jacopini are interested in reducing as far as possible the number of concepts used and it is then reasonable to code up previous flow of control into variables, as this can usually be done within the existing framework. If, however, one's motivation is to simplify the program's structure so that we may better answer questions such as whether the program loops indefinitely, then this coding of the control into variables is no help at all. It remains true though that the block form is a very natural standard form to use, and it is certainly possible to transform many programs into equivalent programs in block form without resorting to the coding of control features as values of variables. Some preliminary conjectures along this line are reported in my paper referred to above.

DAVID C. COOPER
Carnegie Institute of Technology
Pittsburgh, Pennsylvania 15213

A Comment on Galler's Letter

EDITOR:

I find Mr. Galler's letter to the membership [*Comm. ACM 10*, 5 (May 1967)] a well-intended guide to penetrating the ACM power structure. "Fred Jones," a typical programmer with some ideas about file structures, starts in Mr. Galler's account as an unknown, and rises until "he may even find himself a subcommittee chairman." This up-note ending is as unquestioned as that in a classic Hollywood movie, until Mr. Galler adds, "It could happen to almost anyone—it did to some of us." To me, that is an unwittingly frank statement of a menace.

Committees don't often discover anything. If Fred Jones' ideas about file structures are genuinely good, he should indeed spread them around, and he should listen to and benefit from the related ideas of others; the network of committees and meetings is admirably suited for this. But he should also pursue his ideas further, which might best be done by *not* finding himself a subcommittee chairman. With the surplus of "joiner" activities in the computer field, a line should be drawn as to how many to take part in.

If such a line is not drawn, as in Mr. Galler's otherwise excellent editorial, the creative accomplishments possible outside the mechanisms of "the establishment" of committees, meetings, and block-diagram power structures will be killed. In fact, some totally, perversely independent types, I believe, might contribute at least as much to computer science as do the unquestioning joiners of committees. Self-directed, intensive thought and research, as well as some mellow, not-geared-to-the-minute reflectivity, is needed in this computer game.

JERRY A. RALYA
Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543