FISHEYE: A LENSLIKE COMPUTER DISPLAY TRANSFORMATION*

by L. Kleinrock and K. Stevens
Computer Science Department
University of California, Los Angeles, California

ABSTRACT

The success of graphical interaction with a computer is sensitive to the manner in which the data is presented. The display must provide useful detail while maintaining a global view of the surrounding areas. When current windowing techniques are applied to provide such detail, they inevitably sacrifice global vision. The Fisheye transform, however, conserves global vision by compressing the outlying regions of a display while presenting a selected region under magnification for detail. While various compression algorithms are applicable, the Rectangular Fisheye promises reduced distortion (hence simplified curve generation) and increased screen utilization for a large class of display applications.

Keywords

Computer Graphics
Graphical Displays
Fisheye Transform
Data Compression

---

*This work was supported by the Advanced Research Projects Agency of the Department of Defense (DAHC-15-69-C-0285).

FISHEYE: A LENSLIKE COMPUTER DISPLAY TRANSFORMATION*

by L. Kleinrock and K. Stevens

Computer Science Department

University of California, Los Angeles, California

I. INTRODUCTION

A computer graphics system provides a natural means for casual and interactive manipulation of graphical data through a CRT interface. The success of such interaction is subject to the manner in which the graphical data is presented, and more efficient interaction can be obtained through improved techniques for display presentation. In this paper we describe a new display transform, which has been investigated by the authors at the UCLA Computer Science Department.

Display Presentation and Analysis

The process of graphical interaction is inherently iterative. The iteration involves two steps:

1. an analysis of the graphical display in order to formulate the next operation or task to be performed by the machine.

2. the process of guiding the machine to the completion of the task (formulated in 1.) as specified through the use of certain input devices.

Continued display analysis then provides feedback on the results of the

---

*This work was supported by the Advanced Research Projects Agency of the Department of Defense (DAHC-15-69-C-0285).

previous commands, thus directing new interaction. Assuming a rich, well established set of interactive tools, one's ability to interact depends heavily on one's ability to *analyze* the presented display.

While the precise nature of human display analysis is unknown, it appears analogous to the operation of parsing a sentence in order to gain information from its structure (Narasimhan, 1968 [1]; Shaw, 1968, [2]). That is, a display is analyzed through a study of its component parts and their interrelations on a global level. We distinguish two mechanisms: detail (or local) vision; and global vision. If one is conscious of how he observes a scene, he is aware of an inherently hierarchical and iterative process of detail analysis. When presented with a display, the eye is trained upon various areas of complexity.[1] When one focuses on such a region, the field of view decreases. The goal is to first recognize the detail and then to place it in relation to the surrounding areas as the field of view is once again increased. If, however, one cannot recognize the detail, the field of view is further decreased and the process repeats. This "parse" continues until one is satisfied with his analysis. Of course, this model is simplified, for both processes probably proceed in parallel. However, since both mechanisms are necessary for display analysis, *interaction would be impaired unless one is given sufficient capabilities not only for detail observation but also for simultaneous global viewing*.

A difficulty arises since display hardware is limited in both

---

[1] The eye is drawn toward such regions in expectation of gaining information. The mind is conditioned to consider relatively plain or simple areas as uninteresting in this respect (Warnock, 1968 [3]).

resolution and display area. Beyond this, moreover, current display techniques often result in an *unnecessary sacrifice of global vision.* This stems from the very limited resolution of graphic *input* devices in the following way. In the process of interaction, one addresses points on the CRT screen for subsequent reference in various tasks (see 2, above). These points should be determined with an accuracy commensurate with the precision of the graphical data from which the display was derived. To compensate for the inaccuracy of the input devices, one generally resorts to magnifying the display.[2] If, however, the total screen is already required to present the data, then the usual practice is to *clip* out a portion of the data (delimited by a *window*), then magnify and display it. This process of "trimming to fit" implies a loss of global vision, and the observer is forced to interact with a display of only a fraction of the total graphical data.

While the usual consequence of impaired global vision is inefficient interaction (resulting from the necessity of multiple windows when dealing with extensive data), there exist cases in which interaction is virtually impossible with windowing. An example involves the manipulation of lines which cannot fit wholly within a window, thus preventing one from dealing with the graphical data as a single entity with the desired accuracy. In light of such difficulties, two variations on windowing have been developed in attempts to provide greater flexibility: zooming and scrolling. Zooming provides the necessary detail vision; however one

---

[2] Such precision can be achieved by direct input of the coordinates (if known) or through trigonometric constructions in certain cases. However, a light pen or similar cursor device is generally employed for arbitrary points.

field of view is restricted. One must "un-zoom" whenever he desires a global vie of the data. Similarly, scrolling provides magnification, yet one cannot retain a global sense without considerable scanning, a prohibitively expensive task if the clipping is performed by software. If one combines zooming and scrolling, then with adequate computational power, a powerful display technique can result. However, pending the availability of moderately priced windowing hardware [4], one can state that current display techniques lack the necessary capabilities for simultaneously presenting a display with

1. a magnification of that portion of the data currently involved in interaction, and
2. a global view of the surrounding regions of the data.

II. FISHEYE

The windowing operation inevitably reduces global vision; hence it should be performed only when necessary (e.g., thinning excessively complex data sets to keep screen flicker within bounds). However, windowing is often performed as a matter of course merely to magnify a portion of a display, resulting in the loss of global vision.

If, instead of windowing, one transforms the display such that

1. the desired region is magnified to show sufficient detail, and
2. the surrounding regions are suitably compressed to fit within the available screen area,

then one can retain global vision. This transform, called Fisheye (first conceived by L. Kleinrock and researched under his direction by K. Stevens) results in an ultra-wide angle effect similar to photographs taken through a "fisheye lens."

A Fisheye transform promises enhanced graphical interaction, for one can simultaneously study a select region under magnification while observing its relations to the surrounding regions. Of course, the price paid for this enhancement is the compression and distortion of the outlying regions of the Fisheye scene. This transform, however, is intended to provide a *continuous* change of scale (or demagnification) as one views scenes further and further away from the magnified region. In this manner, one maintains a feel for the detail region with respect to the total data as one "takes in" the more global aspects of the display. Consequently, Fisheye provides a more satisfactory combination of detail observation and global vision than is possible with conventional display techniques.

Of course, one cannot retain complete global vision, for the display is increasingly compressed toward the periphery of the data. However, the most important global vision involves that region immediately surrounding the magnified area, for one most often references "auxiliary" detail from this region, as opposed to more distant regions. Indeed, all that is generally needed of such outlying regions is knowledge of the more predominant structural features, or landmarks. The Fisheye transform is successful in this respect, giving slight compression near the magnified region, with an increase in compression as one moves toward the periphery, permitting one to pick out the necessary landmarks. Of course, the success of this technique is limited, for if one increases either 1) the extent of the graphical data to be displayed, or 2) the magnification in the linear region, then he must expect increased edge compression. In the limit, the periphery will become saturated. That is, the compression will become so severe as to render the periphery useless. One may then resort to either (1) a thinning operation (e.g., removal of unresolvable features), or (2) windowing in order to reduce the complexity along the periphery. However, it should be noted that Fisheye places no limits on the extent of the graphical data, for a suitable transform can be devised that would map a point at infinity onto the periphery of the display.

Let us now discuss one version of the Fisheye transform; first, certain terminology must be introduced. A Fisheye display is generated by applying a Fisheye transform to a graphical data set (GDS). Both the Fisheye display and the GDS will be considered to lie in two dimensions relative to Cartesian coordinate systems. The extent of the GDS will be considered the real-world (RW) space. Within this space, an arbitrary point is represented by (RWX, RWY). The Fisheye display will be considered to lie within a viewport, where an arbitrary point is defined by (FEX, FEY). The Fisheye transform φ can then be expressed as

$$\varphi(\mathrm{RWX},\mathrm{RWY}) = (\mathrm{FEX},\mathrm{FEY})$$

where every point in the real-world space maps into the viewport. This provides one with global vision of the total GDS. However, the mapping is also intended to generate a magnification of a particular region. This RW region, referred to as the window, will be scaled so as to map into a corresponding linear region in the viewport.

A geometric model has been devised that aids visualization of this

Fisheye transformation. This model, a generalization of the gnomonic (or central) projection, maps a point R in the real-world space into its corresponding point F in the viewport. This transform involves a surface g as shown in Figure 1.

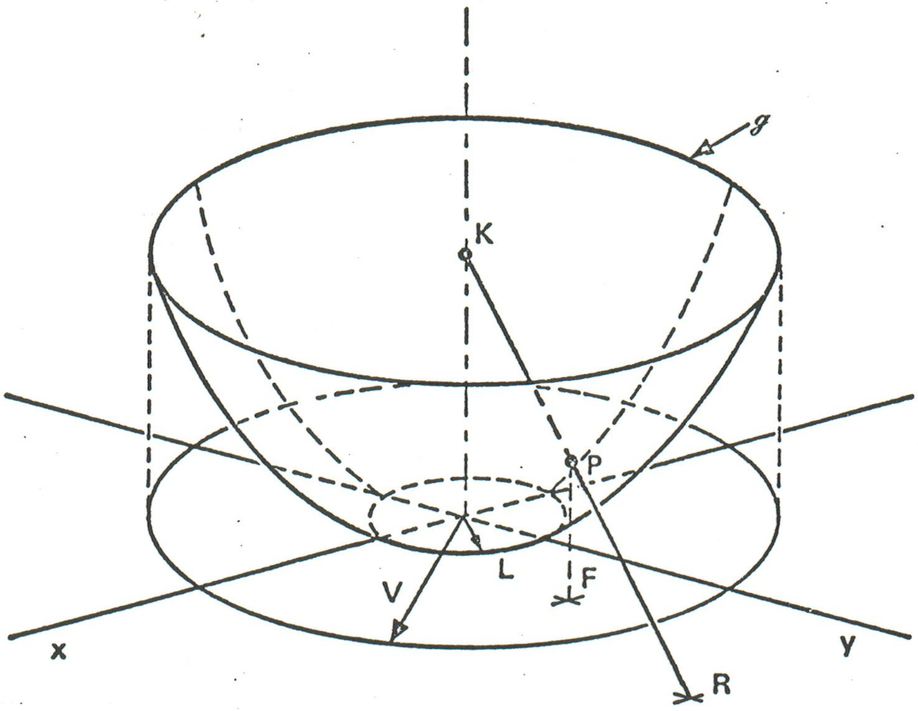


Figure 1. Generalized Gnomonic Projection

An arbitrarily chosen point K is defined as shown, and the point F is then generated from a point R by the orthogonal projection of point P (the intersection of line KR and the surface g) onto the xy plane. It is evident that as R approaches infinity, F approaches the periphery of the viewport. In this illustration, the viewport is a circle of radius V; however in general it would be the closed curve expressed by the (orthogonal projection of the) intersection of the surface g and the plane z=K. It is important to note that g is flattened;

that is, the surface comes in contact with the xy plane. In Figure 1, this region of tangency is circular with radius L. As a consequence, any point R within this region maps onto itself, thus constituting a window/linear region representation. Obviously the magnification is unity in this case; however it is a simple matter to scale the real world first in order to obtain an arbitrary magnification.

Thus we see that the Fisheye transform generates a graphical scene in which both detail and global vision are available. Since in general this compression is non-linear, one can expect distortions; that is, an arbitrary straight line will map into a curved line in the Fisheye display as shown below in Figure 2.

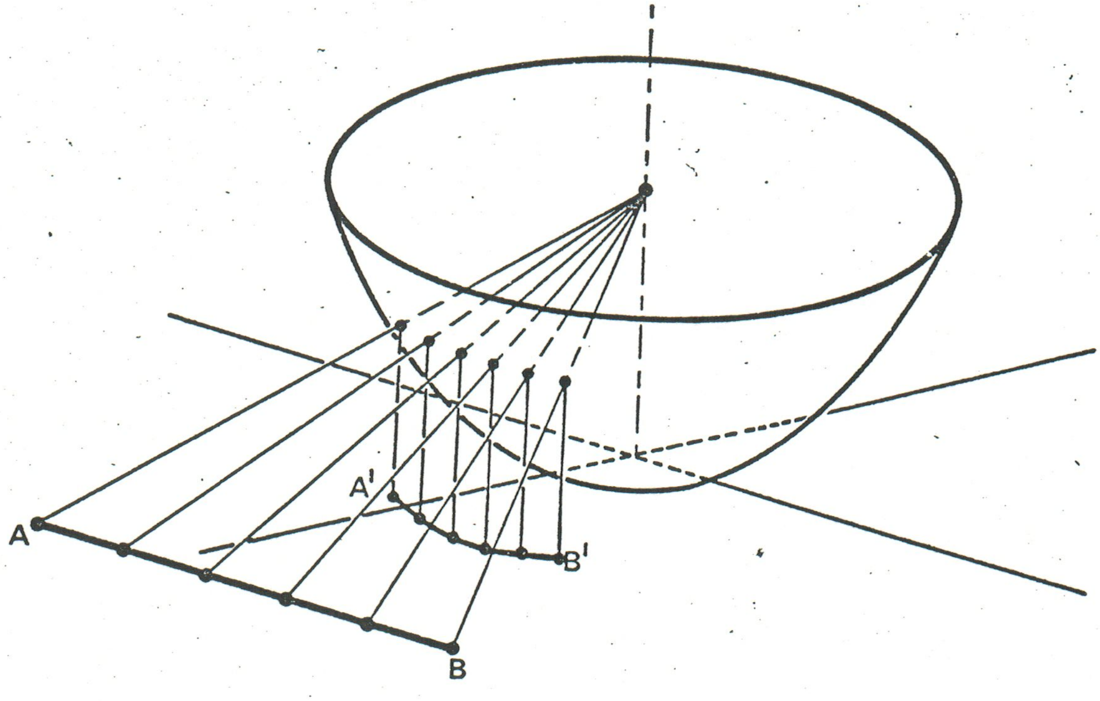


Figure 2. Demonstration of Line Curvature

This effect has two important consequences:

1. curve generation is usually expensive, and
2. the distortions induced in the Fisheye display detract from one's ability to analyze the original graphical scene.

A second property of the general Fisheye transform is less-than-complete screen utilization. It is evident that the viewport configuration is a function of the surface g. For example, Figure 1 illustrates a circular viewport which is not the most efficient display boundary in terms of screen utilization, for generally CRT screens are rectangular.

Rectangular Fisheye

The above properties of a general Fisheye stem from the fact that

$$\text{FEX} = \varphi_x(\text{RWX},\text{RWY})$$

$$\text{FEY} = \varphi_y(\text{RWX},\text{RWY}).$$

That is, both coordinates (RWX,RWY) are used in computing each coordinate FEX and FEY. It has been found that if the coordinates are computed by independent functions

$$\text{FEX} = f_x(\text{RWX})$$

$$\text{FEY} = f_y(\text{RWY})$$

then several benefits result. This transform scheme, the "rectangular" Fisheye, is represented by two surfaces in Figure 3. This figure shows the two surfaces g_1 and g_2 that map RWX to FEX and RWY to FEY, respectively. Each surface extends indefinitely along its respective axis. Associated with each surface is a parallel line (K_1K_1' for g_1 and K_2K_2' for g_2) which performs the analogous function of K in Figure 1.

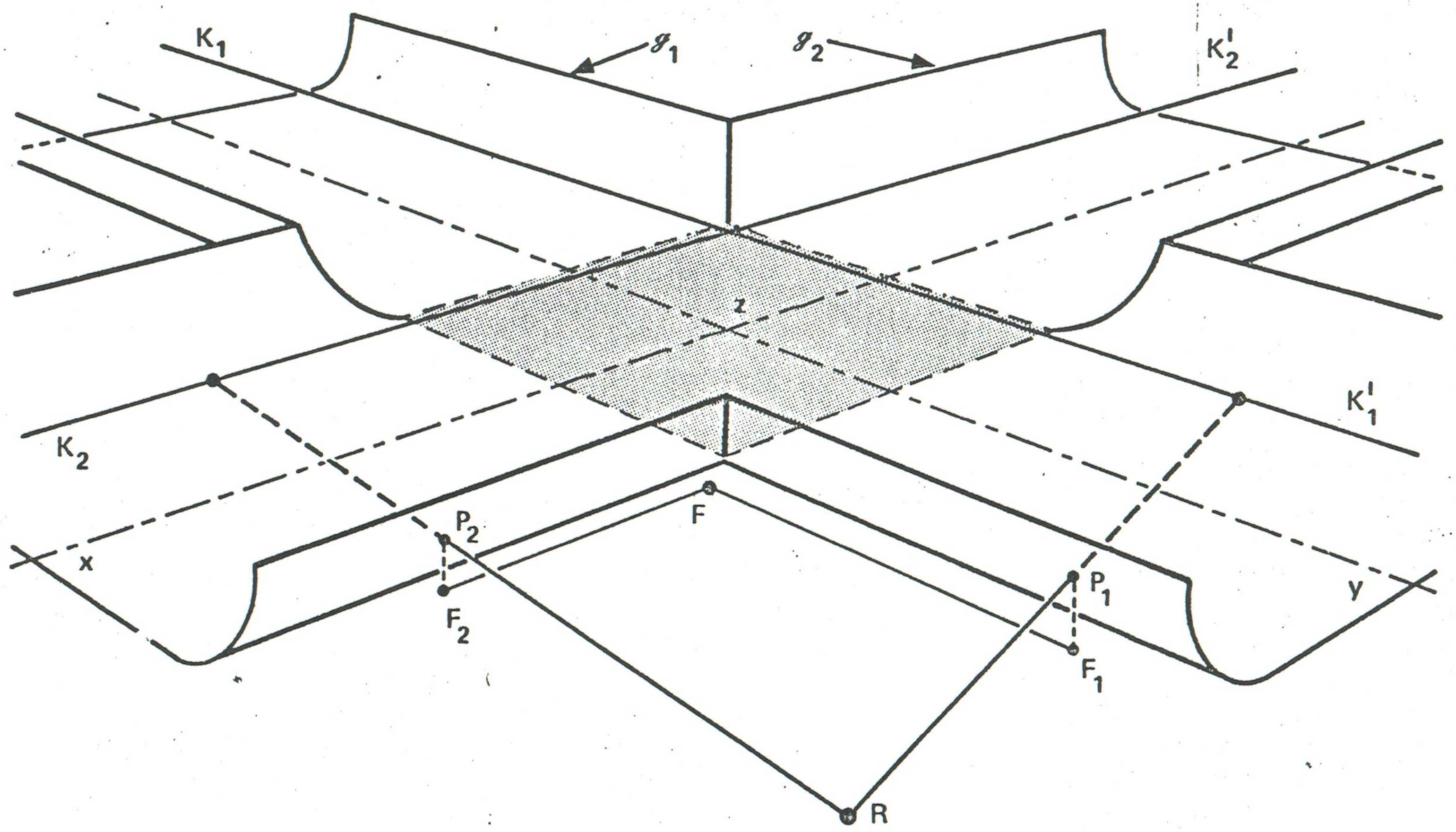


Figure 3. Representation of Rectangular Fisheye

The coordinates FEX and FEY are computed independently to map from R to F.

In studying Figure 3, two major advantages of the rectangular scheme are apparent. First, the real world maps into a rectangular viewport. This has obvious benefits in optimizing display screen utilization. Secondly, due to the independent mapping process, all horizontal and vertical lines remain straight when mapped into the viewport, alleviating the need for curve approximation in such cases. This is most fortunate, for generally graphical scenes are comprised of such lines to a greater extent than lines of arbitrary slope. These properties of a rectangular Fisheye tend to make this approach promising for an actual implementation.

III. IMPLEMENTATION AND APPLICATION OF FISHEYE

A rectangular Fisheye has been implemented in FORTRAN at UCLA. The transform interfaces between the user's program and the graphics system as a "preprocessor". That is, each point in the GDS is passed through the transform before being introduced to the graphics system. This approach leads to a straightforward implementation that is relatively independent of the graphics system. Therefore, such a Fisheye can, with minor modifications, be applied to other graphics systems. This method is demonstrated in Figure 4.

Figure 4. Flow of Control From Fisheye Routine FPLACE to Standard Graphics Routine PLACE

The standard graphics routine PLACE(IX,IY) is called in order to position the CRT beam at the absolute point (IX,IY). In using the Fisheye pre-processor, the corresponding routine FPLACE(RWX,RWY) is substituted, which "positions the beam" at the point (RWX,RWY) in the user's real-world space (which may be many times the size of the CRT screen). When executed, this function calls FTRAN which maps the real-world point into its associated point (FEX,FEY) in the Fisheye space. The transformed point is then passed to the PLACE routine which generates the hardware vector command to position the beam at (FEX,FEY).

A similar process is performed in drawing a line in the real-world space. However, care must be taken to generate a suitable curve approximation for those lines which are neither horizontal nor vertical. In this case the most straightforward approach has been to divide the line segment into n portions[3] and to transform each segment in turn.

The actual Fisheye transformation occurs in the FTRAN routine. This routine transforms each coordinate independently according to separate functions whose coefficients are "tuned" to reflect the relative size and position of the window, linear region, etc. This tuning is demonstrated in Figures 5 and 6 where the function f_x(RWX) is shown for two window positions.

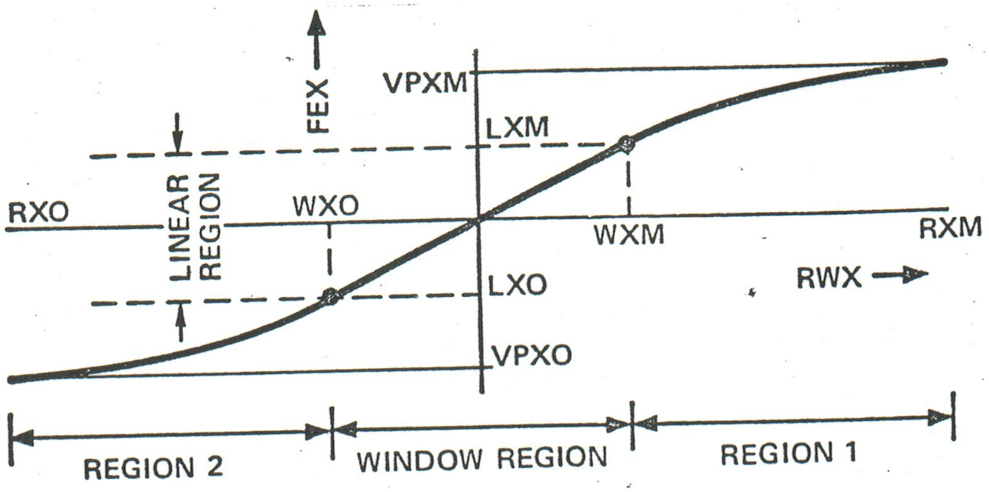


Figure 5. Function f(RWX) for a Centrally Positioned Window

---

[3] Where the choice of N reflects the precision required. The value may be fixed empirically or vary locally along the line according to the degree of curvature experienced (Stevens, 1971 [5]).

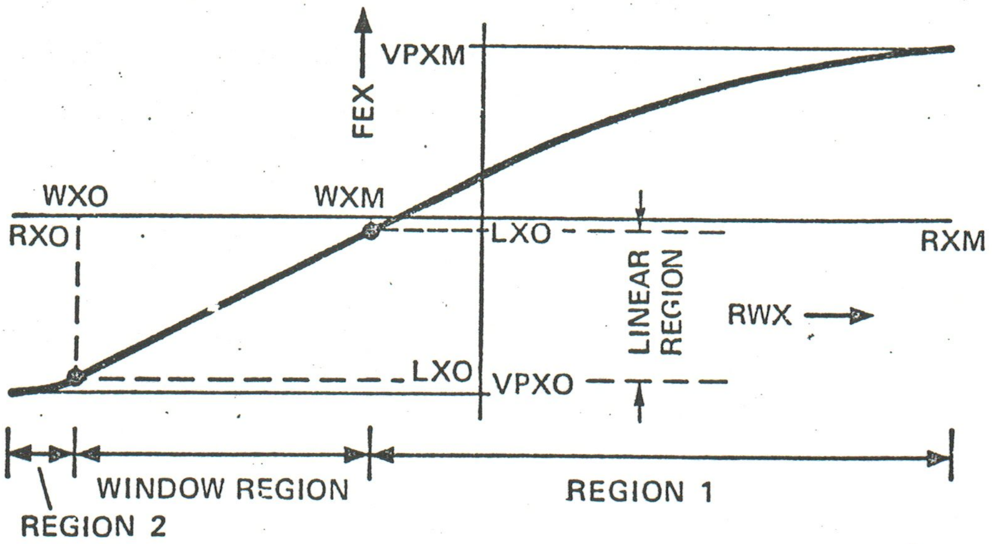


Figure 6. Function f(RWX) When Window Is Positioned Near Left Extreme of GDS

It is evident that the range of each coordinate is divided into three regions. In one form of implementation, f_x(RWX) may be expressed in terms of exponentials:[4]

$$\mathrm{FEX} = \mathrm{LXO} - B_2[1 - e^{-A_2(\mathrm{WXO} - \mathrm{RWX})}] \qquad \text{(region 2)}$$

$$\mathrm{FEX} = \frac{\mathrm{LXM} - \mathrm{LXO}}{\mathrm{WXM} - \mathrm{WXO}}(\mathrm{RWX} - \mathrm{WXO}) + \mathrm{LXO} \qquad \text{(linear region)}$$

$$\mathrm{FEX} = B_1[1 - e^{-A_1(\mathrm{RWX} - \mathrm{WXM})}] + \mathrm{LXM} \qquad \text{(region 1)}$$

A point (RWX,RWY) therefore is transformed in two steps. First, RWX is tested to see within which range it lies, and then the corresponding function is applied to generate FEX. A similar process transforms RWY according to the appropriate f_y(RWY).

A series of application programs have been written [5] employing the rectangular Fisheye, and we illustrate these in the following figure.

---

[4] Exponentials provide elegant formulations of the compression functions; however, they are computationally expensive. Therefore linear approximations to exponentials may be substituted for more efficient transformations.

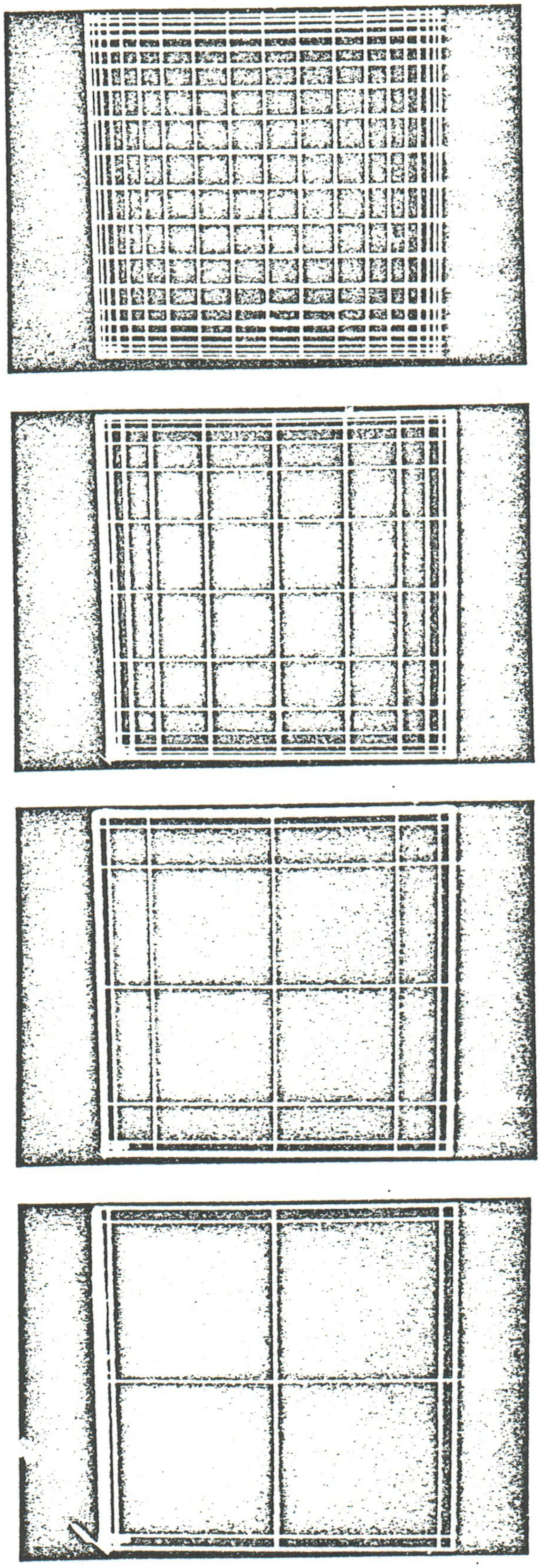
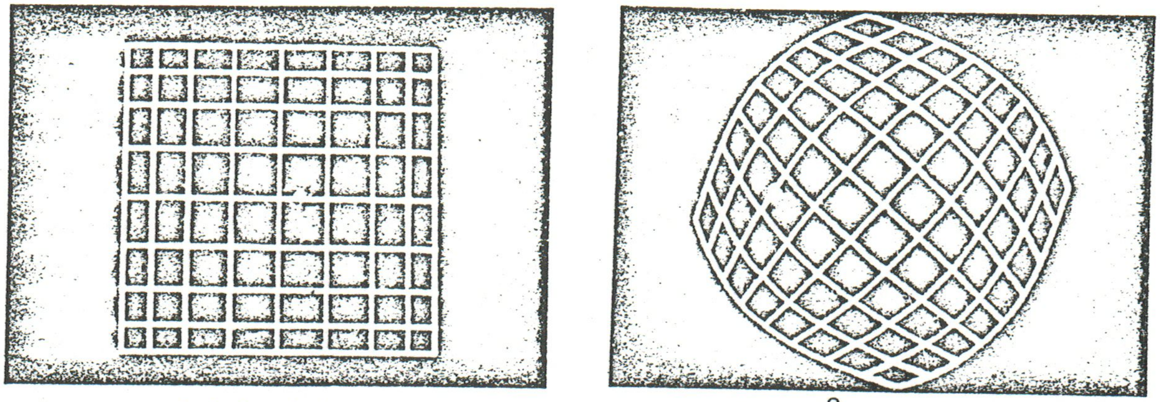
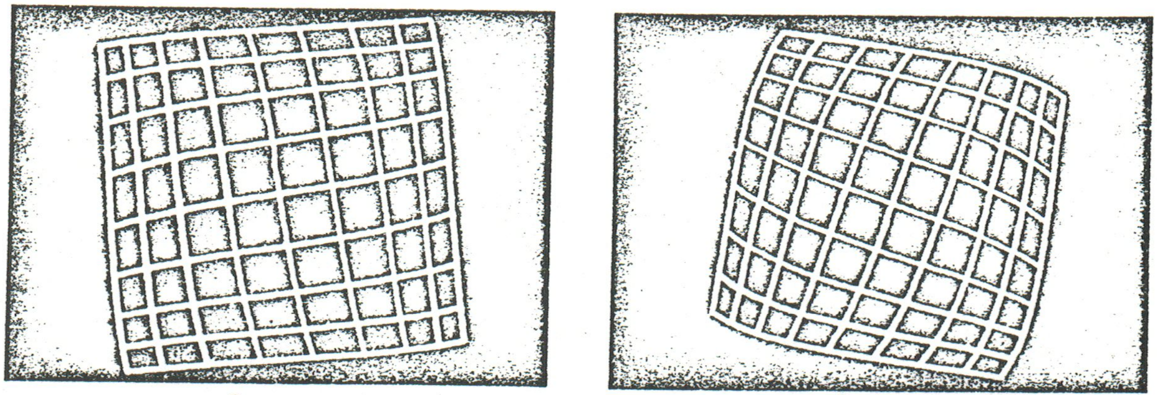


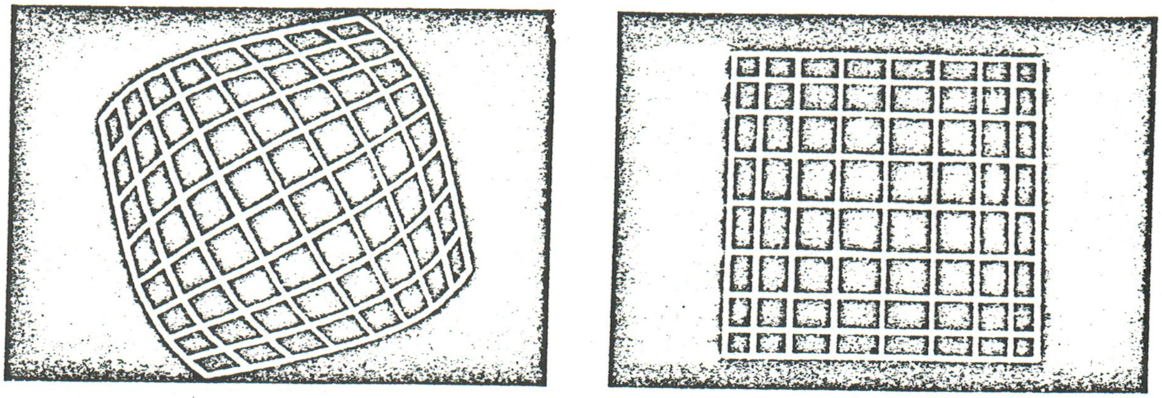
Figure 7. Zoom Sequence Toward Center of 20 x 20 Line Grid



a. Original d. 54° Rotation



b. 18° Rotation e. 72° Rotation



c. 36° Rotation f. 90° Rotation

Figure . 90° Rotation of Grid in Fisheye Space

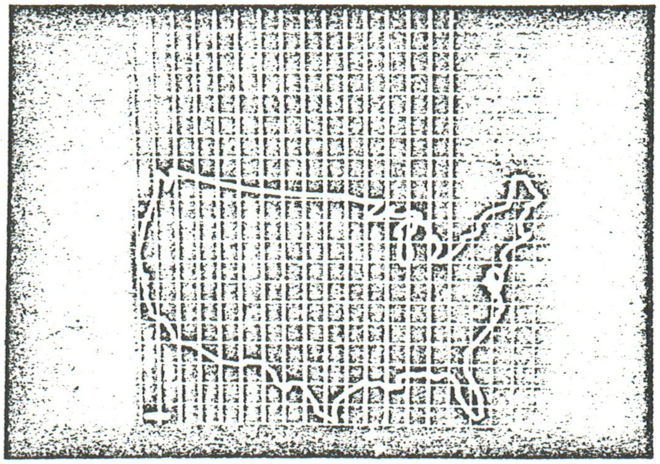


Figure 9. Undistorted View of USA

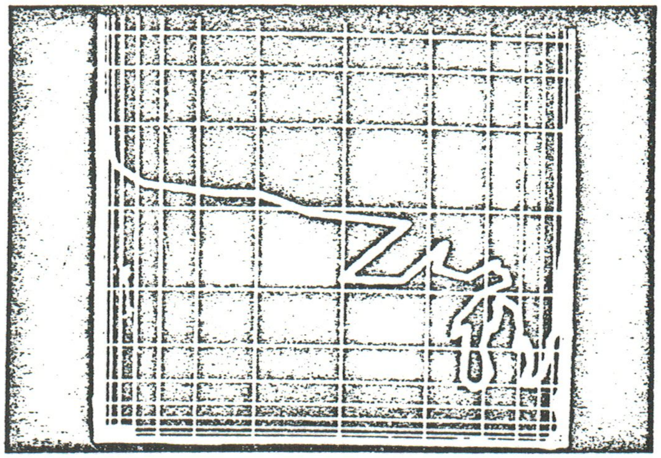


Figure 10. Scroll and Zoom Toward Lake Superior

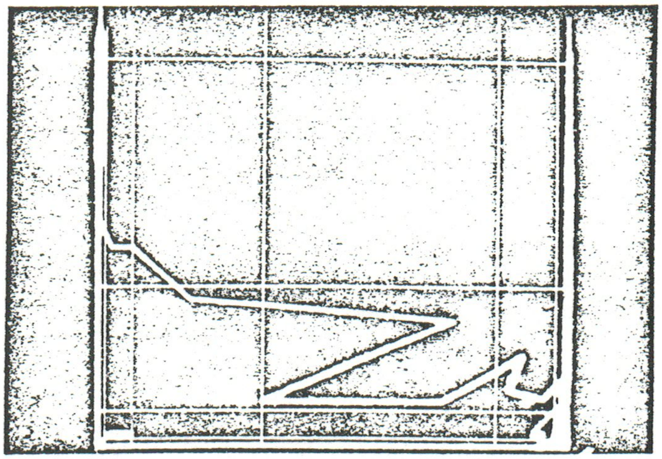


Figure 11. Detail View of Lake Superior

FIGURE CAPTIONS

1. Generalized Gnomonic Projection
2. Demonstration of Line Curvature
3. Representation of Rectangular Fisheye
4. Flow of Control from Fisheye Routine FPLACE to Standard Graphics Routine PLACE
5. Function f(RWX) for a Centrally Positioned Window
6. Function f(RWX) When Window Is Positioned Near Left Extreme of GDS
7. Zoom Sequence Toward Center of 20 x 20 Line Grid
8. 90° Rotation of Grid in Fisheye Space
9. Undistorted View of USA
10. Scroll and Zoom Toward Lake Superior
11. Detail View of Lake Superior