Applications of Distortion-Oriented Presentation Techniques in GIS

Neville Churcher

Department of Computer Science, University of Canterbury,
Private Bag, Christchurch, New Zealand
+64 3 3642353 (voice)
+64 3 3642569 (fax)
neville@cosc.canterbury.ac.nz

Presented at AURISA/SIRC '95—The 7th Colloquium of the Spatial Information Research Centre, University of Otago, in association with AURISA New Zealand and Massey University, April 26th–28th 1995

Abstract

The internal components of a typical Geographic Information System (GIS) have evolved dramatically in recent years. However, the man-machine interface has remained relatively unchanged and is now a limiting factor in performance and usability. Distortion-oriented presentation techniques employ transformations which emphasize regions and features of interest while suppressing, yet retaining, global detail. They have been applied successfully in a variety of application domains to combat problems of visual information overload and to enhance user benefits from graphically presented information. In this paper the field of distortion-oriented presentation techniques is surveyed briefly and two areas in which these techniques could benefit GIS, schema design and Human-Computer Interaction (HCI), are examined in greater detail. Preliminary results from a long-term project are reported.

1 Introduction

The last decade has seen dramatic advances in Geographic Information System (GIS) technology coupled with an equally dramatic rise in the number of GIS users. While database engines, data structures, hardware performance and many other factors have improved greatly in favour of enhanced GIS performance, the user interface has remained comparatively unchanged. The concept of the computer screen as a glass window onto a large seamless map has endured.

It is generally impractical to display in its entirety a system represented in a GIS since the details would be lost because of limited screen size and resolution. This has led to the development of various Human-Computer Interaction (HCI) models incorporating facilities for improving user access to visually presented information. The typical use of a GIS interface involves frequent scrolling, zooming and opening of new windows which can lead to a state of visual information overload.

Distortion-oriented presentation techniques[1] represent an alternative approach to the management of large volumes of visual data and are particularly effective where visual information overload is a potential difficulty. These are precisely the conditions under which a GIS must function. Fisheye techniques, named after the fisheye (extremely wide-angle) lenses used in photography, generally employ transformations which emphasize regions of interest while suppressing, yet retaining, global detail.

[1] In this paper the term 'fisheye' is used—for convenience at the expense of complete accuracy—interchangeably with 'distortion-oriented'

Fisheye views have recently been employed in a number of application domains to enhance user benefits from graphically presented information. Examples include the 3D visualisation of a file system (Robertson *et al.*, 1991), the browsing of graphs (Sarkar and Brown, 1992, 1994), the visualisation of 'linear' information structures (Mackinlay *et al.*, 1991) and of maintenance data (Mitta and Gunning, 1993).

The applications cited use fisheye views as a presentation technique for delivering views of essentially static systems. Churcher (1995) has investigated the suitability of fisheye techniques for applications in interactive environments where the system, as well as the view, is subject to change. A number of issues specific to interactive environments were identified and these will be discussed in the context of GIS applications.

Today's typical GIS deals only with 'flat' data (though transformations may have been performed to produce 2-dimensional images). Nevertheless, many concepts central to distortion-oriented presentation techniques have been used routinely in the production of maps. The information presented visually to the end user has undergone processing which may have altered its appearance considerably. The use of circles whose size indicates the population of the cities they represent rather than their physical area is a well known example. Another common example is tourist maps in which the sizes and positions of landmarks and other features are very different from their undistorted values while essential topological relationships are preserved.

Two major potential applications for fisheye techniques in GIS are explored in this paper. One is viewing and querying maps and other visually presented data. The second is interactive diagramming environments such as database schema design.

The remainder of this paper is set out as follows. Some of the limitations of the standard techniques used for visual data presentation are discussed in section 2. Distortion-oriented techniques are outlined in section 3 and their application to map viewing is the subject of section 4. GIS-related issues in schema design are discussed in section 5 and one technique which has been used in GIS schema design is described in section 6. A fisheye implementation of this technique together with a discussion of the extension of fisheye techniques to dynamic systems is contained in section 7. Some conclusions and suggestions for future work are presented in section 8.

2 Viewing models

The concept of the computer screen as a glass window onto a large seamless map has endured. The need to provide a sufficiently detailed display of part of the larger system while minimizing loss of context information and usability has led to the development of techniques including:

- scrolling
- zooming
- context maps
- multiple windows

Scrolling displays allow maps to be viewed as 'seamless' rather than as being made up of many separate pages linked together. Instead of following off-page connectors to navigate from one page (screen) to another—a frustrating process familiar to anyone who has ever used a city map in book form—the user may scroll smoothly to any point on the underlying map. Scrolling has many advantages but a number of user interface issues remain problematic.

Scrolling is generally implemented with several levels of granularity. These may include a small unit for fine adjustment, a larger step for page-size adjustments and a continuous motion.

Both page and continuous scroll steps lead to breaks in the continuity of the display. This is partly due to the computational overheads involved and partly due to the user's perception. Even small scrolling adjustments may involve considerable computation, particularly where complex data structures—such as quadtrees in GIS—are involved. While it is possible to anticipate user actions by pre-fetching data for neighbouring regions there remain many outstanding difficulties.

There is a finite (sometimes considerable) response time and some important features may no longer be visible in the updated view. Such effects obscure global structure and cause disruptions in the user's train of thought and, ultimately, lead to lower productivity and increased possibility of error.

Further issues arise when the user needs to vary the scale at which the map is displayed. Some features may only be relevant in a particular range of scales or may require different representations according to the current scale. As an example, consider the Auckland harbour bridge. This feature might not appear at all on a map of the North Island such as that suitable for inclusion in a school atlas. As the resolution increases, the bridge might be represented first by a point, then by a line, a polygon and so on right up to the degree of detail represented on the blueprints. Text labels may appear at certain scales, and other display properties such as colour font size and shading will also depend on scale. This is essentially the idea of viewing levels (Penny *et al.*, 1989; Ewing, 1992).

Typical use of a GIS involves a user 'zooming in' to examine part of the display in greater detail. Such zooming occurs abruptly and is often jarring due to the finite response time of the system. This hampers the user's continuing interaction with the system, as described above in the context of scrolling, and the relationship between the zoomed area and its surroundings may be lost.

In some situations a 'you are here' context map showing the entire map at very low magnification, may provide some assistance (at the expense of a more complex user interface and increased processing overhead) but in the general case navigation remains a problem.

Similarly, solutions based on allowing multiple windows to be opened, each containing part of the map at a particular scale, lead to the proliferation of overlapping windows and, ultimately, to frustration and lowered productivity—even on the largest of monitors. Considerable mental effort is required for the user to mentally integrate parts of several views, each in its own window and potentially at a different scale.

The issues raised in this section have been encountered in other fields, such as the management of large diagrams in software engineering. While no single solution has yet been proposed, fisheye techniques appear to provide a way to combine detailed views of features of interest while retaining context information, preserving topological relationships and without the necessity for cluttering the display by introducing multiple windows.

3 Distortion-oriented techniques

Most current work in distortion-oriented presentation techniques is based on ideas proposed by Furnas (1986). Rather than attempting to provide a display which reproduces faithfully a map or other document, a distortion-oriented system emphasizes the connections between the various components and highlights those which are of particular significance.

A key concept is the degree of interest function (DOI) which includes contributions from the *a priori* importance (API) of each component as well as contributions depending on its 'distance' from the focus. Fisheye views are then constructed by displaying the components in decreasing DOI order until either all components are shown or a cutoff value is reached. The exact nature of the various functions involved depends on the application domain—a DOI suitable for binary trees will not be suitable for program listings or maps—and the success or otherwise of the application depends critically on an appropriate choice being made.

We consider a rectangular region containing a focal point (x_f, y_f). It is convenient to work with normalised coordinates $(\hat{x}, \hat{y})$ whose magnitudes range from 0 at the focus to 1 at the boundary. In general, transformations have both geometric (spatial) and non-geometric components. Spatial transformations of the general form

$$\hat{x}' \equiv \mathcal{G}(\hat{x}) \tag{1}$$

are used to compute the positions of objects in the fisheye view. The first derivative $\mathcal{G}'(\hat{x})$ gives the magnification, or scale factor. Non-geometric transformations involve the re-computation of size and detail for each object, given its new location.

Transformations may have both continuous and discrete components. Component labelling is an example of the latter category: as the component size decreases (continuously), the size of the font used for labels is reduced to the next available size, typically by multiples of some discrete unit such as point size. Eventually, the component may become 'too small' and its label may be omitted. Information about the colour, shape and border thickness of components is also likely to be treated with the use of cut-offs.

The approach used by Sarkar and Brown (1992, 1994) refined Furnas' basic ideas for application to graphs. The spatial transformation used was

$$\mathcal{G}(\hat{x}) \equiv \frac{(d+1)\hat{x}}{d\hat{x}+1} \tag{2}$$

where d is a constant known as the *distortion factor*. The same transformation is applied independently to the $\hat{y}$ coordinates.

Sarkar and Brown (1992, 1994) also discuss a corresponding polar transformation. While this is arguably more appropriate for some map viewing applications, the form of equation 2 has been used in the present work for simplicity and consistency.

Many different transformation functions are possible provided that they are 'well-behaved' over the range of $\hat{x}$ under consideration. The particular functional form used depends on application-specific factors (see e.g. Leung and Apperley, 1993).

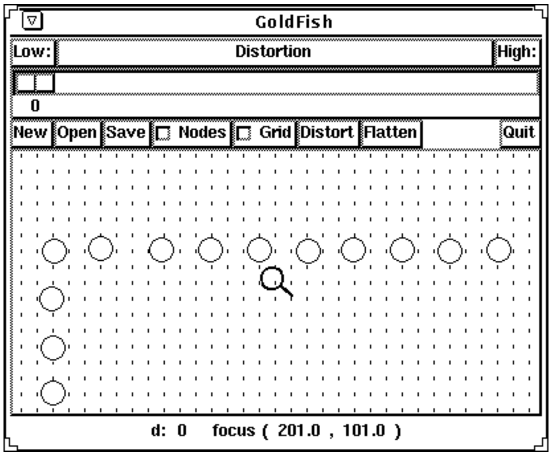
For example, the polyfocal transformation (Kadmon and Shlomi, 1978) was derived specifically for cartographic application and uses

$$\mathcal{G}(\hat{x}) \equiv \hat{x} + \frac{A\hat{x}}{1 + C\hat{x}^2} \tag{3}$$

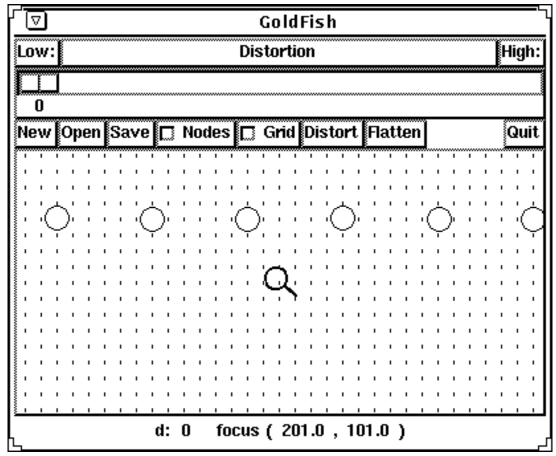
where A and C are constants. The aim of this transformation is to permit the representation of thematic variables by distortions of the features which they describe. Thus, a map representing the distribution of unemployment for the U.S. might have a focus at the centroid of each state whose 'power' (i.e. values of A and C in equation 3) would be derived from the unemployment figures for the corresponding state.

The application of fisheye transformation to map viewing is outlined in figure 1 which shows a number of nodes representing stations on an imaginary subway system. A square grid is included for orientation. Figure 1(a) shows an undistorted (or 'flat') view and the more detailed view of the central region obtained by a zoom operation with a magnification factor of 2 appears in figure 1(b). The reference grid remains unchanged. While greater detail is apparent in the central region of figure 1(b) many of the nodes have moved outside the window.

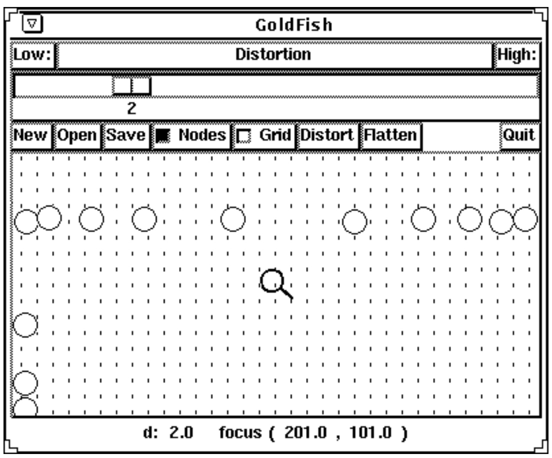
An alternative means of obtaining greater detail by using a fisheye transformation is shown in figure 1(c). The transformation used is that of equation 2 with $d = 2$ and the focus at the



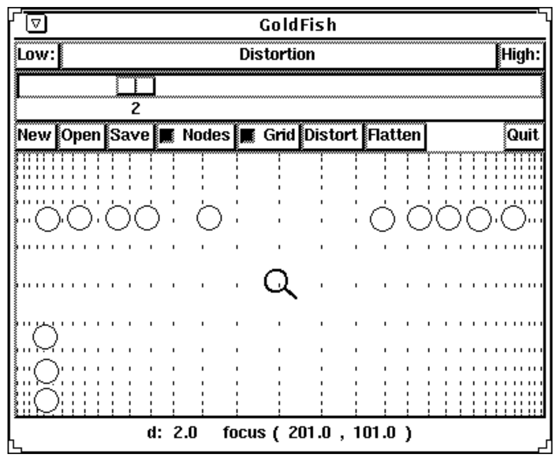
(a) Flat



(b) Zoom (×2)



(c) Fisheye ($d = 2$)



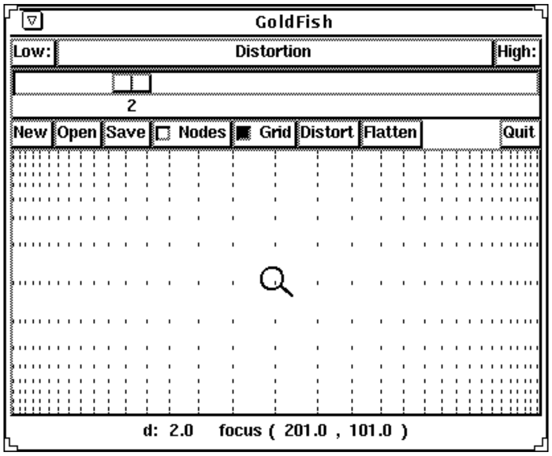
(d) Fisheye ($d = 2$) World

Figure 1: Several views of a network

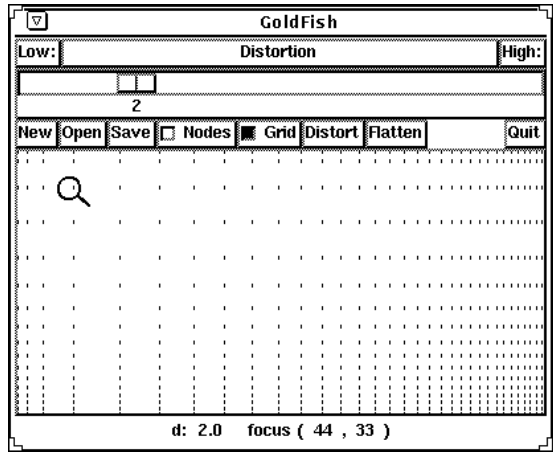
grid point lying within the magnifying glass symbol in the figure. Once again, the reference grid remains unchanged. As with zooming, greater detail is apparent in the centre of the figure. However, the major difference between figures 1(b) and 1(c) is that all nodes remain visible in the latter. It is clear that, unlike figure 1(b), the regular spacing between the nodes has not been preserved by the transformation. Figure 1(d) includes the transformed grid to illustrate the effects of the transform with its focus at the position shown. Note that the point at the focus has not moved.

4 Fisheye views of maps

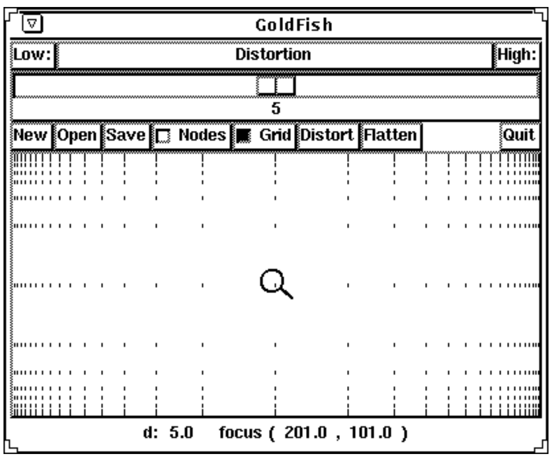
Some work on the application of fisheye views to maps has already been done. Sarkar and Brown (1992, 1994) give both polar and Cartesian transformations on a map of the U.S. and several authors (e.g. Hollands *et al.*, 1989; Leung, 1989; Leung and Apperley, 1993) have used examples



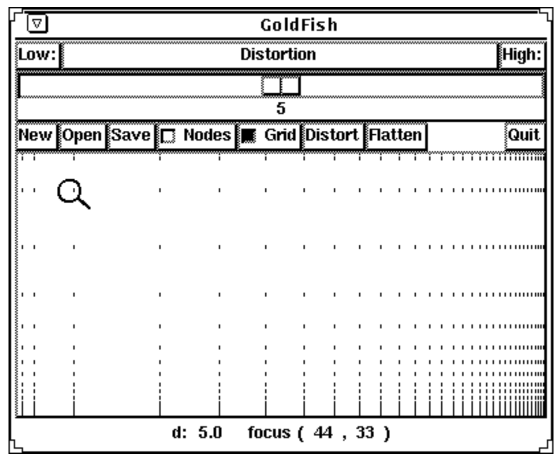
(a)



(b)



(c)



(d)

Figure 2: Effect of moving the focus and altering distortion

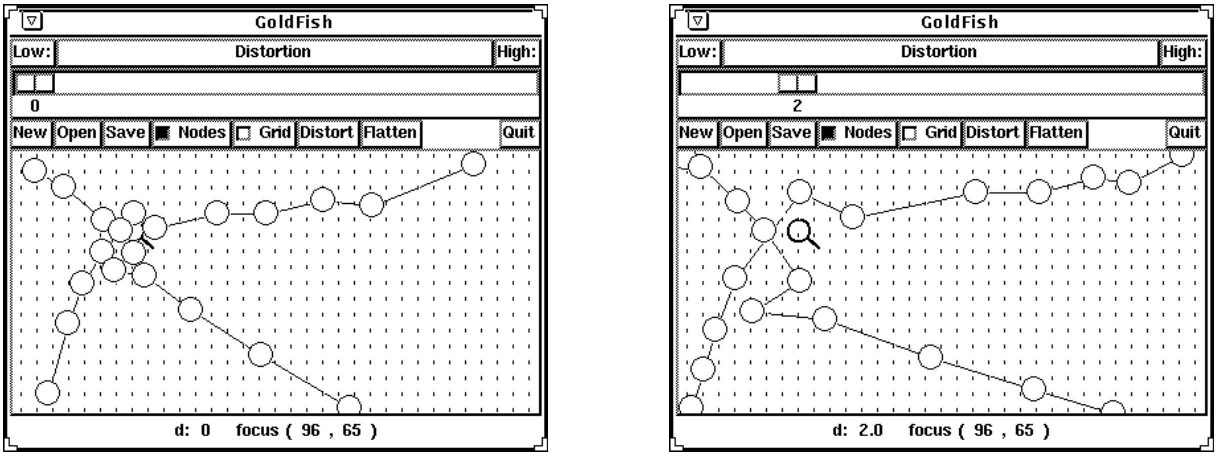
involving maps of subway systems.

Activities such as map browsing require that the position of the focal point and the 'power' of the transformation be under user control. The effects of moving the focus and altering the parameter d in equation 2 are shown in figure 2.

One possible use of a fisheye HCI model is illustrated in figure 3. There are two subway lines—one running from North-West to South-East and one running from South-West to North-East. The density of stations in the region where the lines cross is—as is typical in real cities—too great to allow questions such as "Which line is this station on?" or "At which station can I change lines?" to be answered readily when a flat view is used as in figure 3(a).

Zooming and/or scrolling add to the interface complexity without necessarily improving utility. For example, a user planning a route between the terminal stations of each route may wish to obtain answers to the above questions, by examining part of the map in greater detail, while keeping the stations representing the start and end points of her journey in view.

Figure 3(b) shows how a fisheye view might be used in such a situation.



(a) (b)

Figure 3: Planning a subway journey

Some tasks involve more than one focus. Multiple foci present some difficulties, although some transformation functions (such as the polyfocal transformation described in section 3) are able to cope with these. An alternative approach, called 'zooming' was developed by Churcher (1995) to allow several patches of the diagram to be viewed in greater detail while retaining the conceptual advantages of a single focus.

The zooming operations leave the focus unchanged but cause a further transformation to take place at half the current distortion—regarding the current, possibly already distorted, view as 'effectively flat'—with its apparent focus at the current mouse position. Figures 6(c) and 6(d) illustrate the effect of zooming transformations. The popup menu is used to distinguish clearly the user action for initiating this command from that for simply changing the focal point.

One can readily imagine other advantages of fisheye views of maps. For example, in a cadastral GIS, the entire boundary of a region can be kept in view while the interior is examined in arbitrarily high detail. Transformations such as those developed by Sarkar and Brown (1992, 1994) provide a natural way to deal with the changes appropriate to different viewing levels as discussed in section 2.

5 Database schema design

As GIS have become more advanced, the need for careful schema design, and the consequences of poor schema design, have become better understood. The importance of a good data model of spatial data structures was stated clearly by van Roessel (1987). Some of the implications of data dependencies for GIS, particularly for non-spatial data, were outlined by Pascoe and Churcher (1990, 1991).

Unfortunately, schema design can be a particularly complex problem—many NP-complete problems lurk below the surface—and the choice of appropriate design heuristics can significantly affect the quality of the resulting schema. GIS is an application domain which may be regarded as having some redeeming features. For example, the separability of spatial and non-spatial entities (i.e. entity types or relations) allows the independent development of schemas involving fewer data items. The non-spatial entities tend to have simple relationships to the spatial ones (e.g. several attributes such as population and rainfall describe properties of a polygon representing a province) and only rarely participate in relationships with other non-spatial entities.

On the debit side, practitioners may have little formal training in data dependencies and rigorous schema design or may not have updated their knowledge to include more modern data modelling concepts such as object-oriented techniques. Such people can benefit from a schema design technique which requires little formal mathematical background, gives consistently good results and is amenable to software support.

One such technique is that used by van Roessel (1987) having been developed by Smith (1985). Smith's method is outlined in section 6.

Only a handful of references to Smith's method are to be found in the literature. It is sensitive to the precise choice of dependency list description and, without considerable computer assistance, becomes unmanageable for all but the most trivial systems. Constructs such as N:M or ternary relationships may not have unique or obvious representations in terms of the textual descriptions of dependencies used in the method. As has been argued elsewhere (Wilson, 1989), it is generally preferable for the user to work at the level of the corresponding diagram and for the dependency lists to be maintained by software.

However, for the reasons outlined above, the schemas arising in typical GIS applications are suitable for the use of Smith's method.

6 Smith's method

Smith's method (Smith, 1985) is a semi-formal synthesis procedure for constructing a database schema from a set of elementary facts. Although it has its weaknesses (Neil, 1988; Wilson, 1989), Smith's method is potentially suitable for small to medium sized problems and, particularly with software assistance, is capable of delivering good results while requiring little theoretical knowledge. Effective use of other synthesis techniques (Kent, 1984; Bernstein, 1976) requires a greater formal background in areas such as data dependencies.

The method consists of three steps. First a dependency list is constructed to record the elementary facts. These include attribute names, single-valued dependencies (SVDs), multi-valued dependencies (MVDs) and composite attributes.

This information is then collated in the form of a bubble diagram. Figure 4 shows the diagram corresponding to the polygon layers example described by van Roessel (1987).

Individual attributes are denoted by labelled bubbles and groups of attributes (composites) are enclosed by unlabelled bubbles. SVDs are represented by single-headed arrows, MVDs by double-headed arrows. The additional levels of bubbling around some attributes represents the independence of dependencies involving that attribute. For example, the double bubble around attribute PNTID corresponds to the two dependencies *each* NODEID *is associated with a unique* PNTID and, independently, *Each* PNTID *identifies a specific* X, Y *co-ordinate pair*.

Finally, a set of procedures is applied to derive the appropriate relation schemas. Figure 5 shows some of the relations corresponding to the diagram of figure 4.

7 Experience with `photi`

Applications to date, such as those cited in this paper, have concentrated on the provision of fisheye views of essentially static systems. This is highly appropriate for GIS since the data they

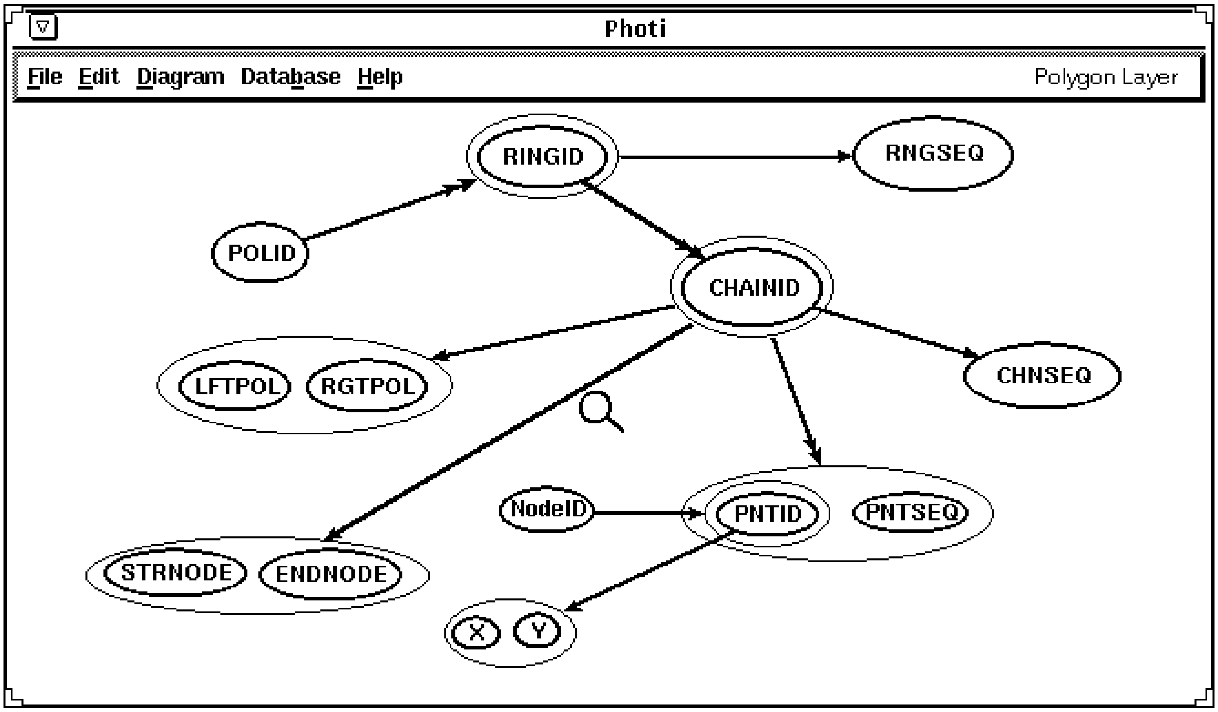


Figure 4: Bubble diagram for the polygon layer model

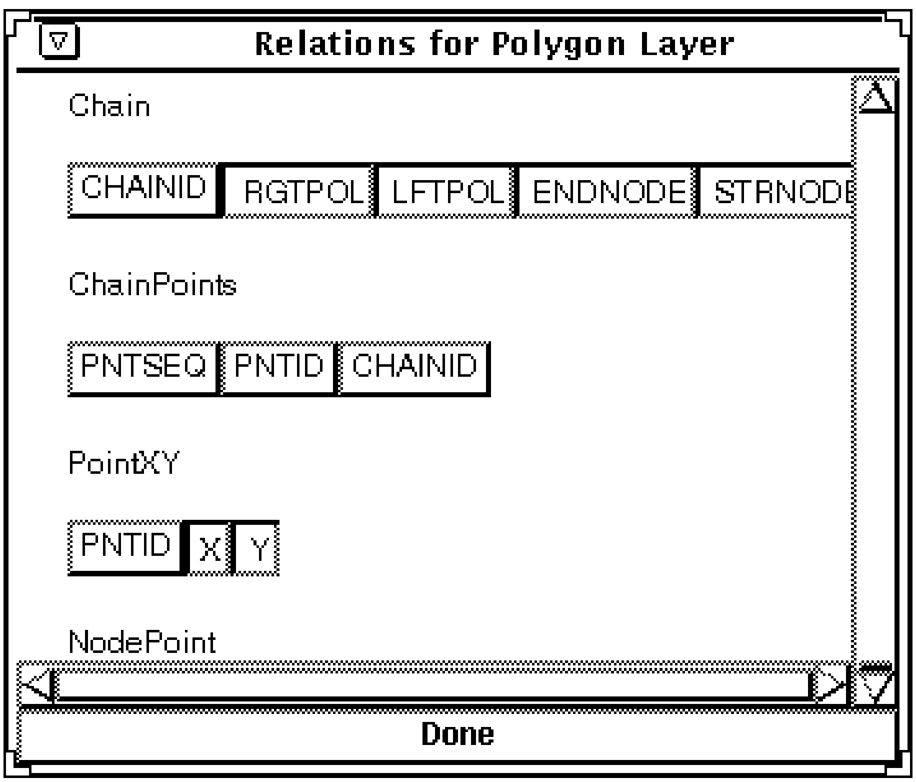


Figure 5: Relations for the polygon layers model

contain is ‘read only’ for most users. However, the situation is very different in the case of interactive systems, where fisheye views must be produced for a system which is evolving through the creation, modification and deletion of components and their connections. Software for diagramming techniques is typical of such systems.

A recent investigation of the potential applications of fisheye techniques to interactive diagramming (Churcher, 1995) used Smith’s method as an example and described `photi`, a fisheye view diagrammer for Smith’s method.

Figure 6 illustrates a number of Photi’s capabilities using an example diagram from Smith’s paper (Smith, 1985). The undistorted view is shown in figure 6(a) with the focal point located near the bubble labelled **B**. Leaving the focus unchanged while increasing the distortion produces the view of figure 6(b). Figures 6(c) and 6(d) show the effect of a ‘zoom’ operation, as described in section 4, centred on the upper left corner of the popup menu. The bubble group containing **L** and **M** has been expanded while retaining the emphasis on the region surrounding the focal point. The effect of moving the focal point, while keeping the distortion constant, is shown in figure 6(e).

Churcher (1995) concluded that Fisheye techniques do appear to have considerable potential for interactive diagramming applications though major differences from their use in static systems are apparent. Several HCI questions arise, including:

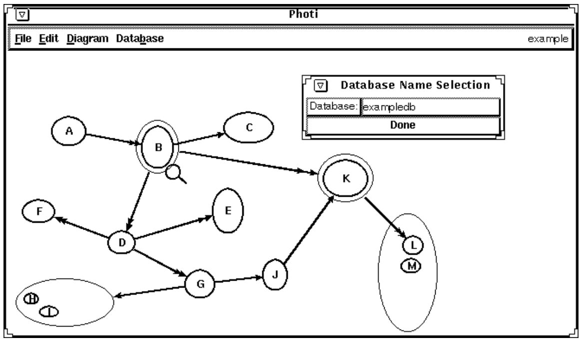
- “Are components created in a distorted view to be regarded as distorted (i.e. congenitally deformed)?”
- “Should components re-scale as they are moved?”
- “How can the user be given complete, flexible control of the view without re-introducing interface complexity by requiring adjustment of many transformation parameters?”

A new consideration, *a posteriori* importance, specific to interactive systems was identified. For example, the API of an attribute in a schema may be greater if it is prime. However, it is not possible to determine when it is created whether a given attribute will become or remain prime. The incorporation of *a posteriori* importance in visual worth (DOI) computation is illustrated in figure 6(f) which shows the effect of highlighting prime bubbles.

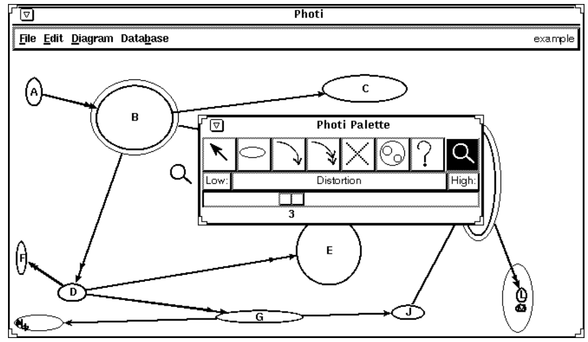
Both `photi` and the application used to produce the remainder of the figures in this paper are written using Tcl/Tk (Ousterhout, 1994). Tcl (Tool Command Language) is a powerful and flexible interpreted language while Tk provides a widget set for GUI development. Tcl is available for a large number of Unix variants, and has been ported to DOS and other operating systems. Tk originally required the X11 windowing system but has recently been ported to Windows and a Macintosh version is reported to be under development.

Tcl is a suitable language for prototyping and implementation. Its advantages include ease of integration with other applications (the Ingres DBMS in Photi’s case) and the ability to allow arbitrarily complex interaction with the user via the direct or indirect entry of new scripts. This allows individual users to tailor the transformation functions as required without requiring extensive knowledge of programming.

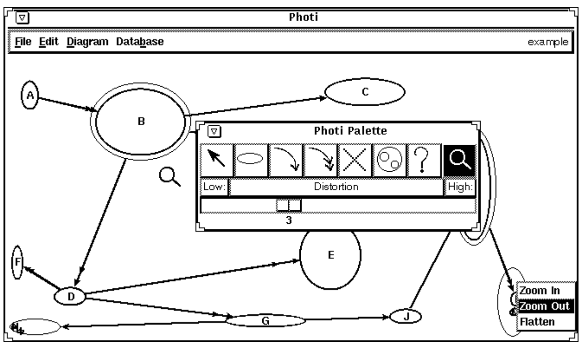
Clearly, performance is a major concern for fisheye applications. If desired, the interpreter can be augmented with user procedures implemented in C rather than Tcl. In applications such as Photi, this is most appropriate for commands which are called repeatedly in response to mouse motion events. The computation of the crossing points of dependencies and bubbles, which are continually changing when bubbles are moved, has been implemented via a C function. Satisfactory performance has been obtained for a number of small to medium sized systems (up to



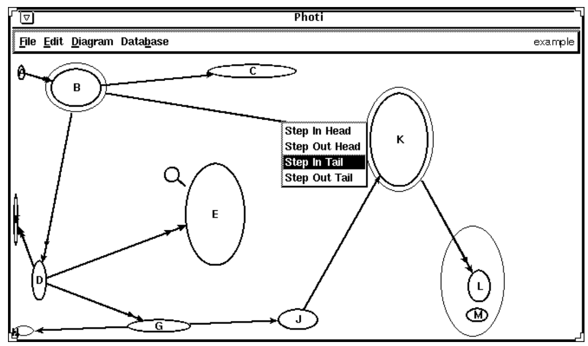
(a) Original (flat) form



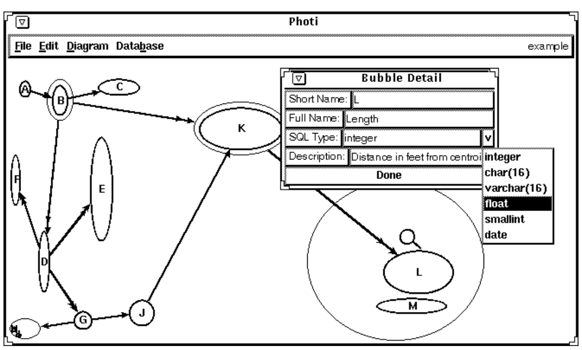
(b) Distorted view ($d = 3$), focus unchanged



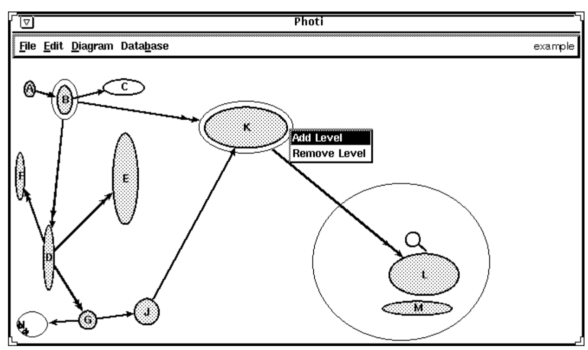
(c) Before zoom operation



(d) After zoom, focus unchanged, dependency management



(e) Focus moved, distortion unchanged



(f) Prime bubbles highlighted, level management

Figure 6: The interface of `photi`

200 bubbles). On a Sun ELC a new view resulting from a distortion change, in which all components are recomputed and redrawn, is completed in less than 2 seconds for the diagrams shown in this paper. However, experimentation suggests that significant performance gains are possible as more commands are implemented in C.

8 Conclusion

Fisheye techniques are an effective means of managing the presentation of complex visual information and have considerable potential for application in GIS. Two areas where they could be readily and immediately applied to GIS, namely map viewing and schema design, have been discussed in this paper.

Several other application areas within GIS are possible. For example, one might consider working with periodic co-ordinates and wrapping the display on a sphere to produce a 'fisheye globe'. Another possibility is the development of groupware applications to allow several users to design co-operatively a schema or to view and discuss a map. The availability of a Tcl-based groupware development kit (Roseman and Greenberg, 1992) makes the latter a particularly attractive project and it is currently under investigation.

Much work remains to be done on human factors aspects of fisheye technology. Further studies are required to establish clearly the conditions under which fisheye views are more effective than conventional presentation mechanisms. Ultimately, once such questions have been clarified, it will be appropriate to invest effort in the development of high performance fisheye applications. In the meantime, exploratory environments, based on languages such as Tcl, are an effective way to explore the exciting possibilities offered by fisheye technology.

References

Bernstein, P. (1976). Synthesizing third normal form relations from functional dependencies. *ACM Transactions on Database Systems*, **1**(4), 277–298.

Churcher, N. (1995). Photi—a fisheye view of bubbles. *Information & Software Technology*, **37**(1), 31–37.

Ewing, G. (1992). *Experiments in Interactive Map Retrieval*. Ph.D. thesis, University of Canterbury.

Furnas, G. (1986). Generalised fisheye views. In *Proc ACM SIGCHI '86 Conference on Human Factors in Computing Systems*, pages 16–23.

Hollands, J., Carey, T., Matthews, M., and McCann, C. (1989). Presenting a graphical network: A comparison of performance using fisheye and scrolling views. In G. Salvendy and M. Smith, editors, *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, volume 12B of *Advances in Human Factors/Ergonomics*, pages 313–320, Amsterdam, The Netherlands. Elsevier. proc. Third International Conference on Human-Computer Interaction, Boston Massachusetts.

Kadmon, N. and Shlomi, E. (1978). A polyfocal projection for statistical surfaces. *The Cartographic Journal*, **15**(1), 36–41.

Kent, W. (1984). Fact-based data analysis and design. *Journal of Systems and Software*, **4**, 99–121.

Leung, Y. (1989). Human-computer interface techniques for map based diagrams. In G. Salvendy and M. Smith, editors, *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, volume 12B of *Advances in Human Factors/Ergonomics*, pages 361–368, Amsterdam, The Netherlands. Elsevier. proc. Third International Conference on Human-Computer Interaction, Boston Massachusetts.

Leung, Y. K. and Apperley, M. D. (1993). A unified theory of distortion-oriented presentation techniques. Mathematical and information sciences report A93/1, Massey University.

Mackinlay, J., Robertson, G., and Card, S. (1991). The perspective wall: Detail and context smoothly integrated. In *Proc ACM SIGCHI '91 Conf. on Human Factors in Computing Systems*, pages 173–179, New Orleans, Louisiana.

Mitta, D. and Gunning, D. (1993). Simplifying graphics-based data: applying the fisheye lens viewing strategy. *Behaviour & information technology*, **12**(1), 1–16.

Neil, S. (1988). Survey of a database design technique and guide to database design methods. Project Report, Department of Computer Science, University of Canterbury.

Ousterhout, J. K. (1994). *Tcl and the Tk Toolkit*. Addison-Wesley, Reading, Massachusetts.

Pascoe, R. and Churcher, N. (1990). Sharing geographical data. In *Proc. Second Colloquium of the Spatial Information Research Centre*, pages 274–287, Dunedin, N.Z.

Pascoe, R. and Churcher, N. (1991). Modeling the sharing of geographic data. *N.Z. Journal of Computing*, **2**(1), 45–53.

Penny, J., Ewing, G., and Pascoe, R. (1989). Aspects of interactive systems for computer mapping. In P. Nickolas, editor, *Proc. ACSC-12*, volume 11 of *Australian Computer Science Communications*, pages 287–295, University of Wollongong, Australia.

Robertson, G., Mackinley, J., and Card, S. (1991). Cone trees: Animated 3d visualizations of hierarchical information. In *Proc. ACMSIGCHI '91 Conf. on Human Factors in Computing Systems*, pages 189–194, New Orleans, Louisiana.

Roseman, M. and Greenberg, S. (1992). Groupkit: A groupware toolkit for building real-time conferencing applications. In *Proc. CSCW '92*.

Sarkar, M. and Brown, M. (1994). Graphical fisheye views. *Communications of the ACM*, **37**(12), 73–84.

Sarkar, M. and Brown, M. H. (1992). Graphical fisheye views of graphs. In *Proc. ACM SIGCHI '92*, pages 83–91.

Smith, H. C. (1985). Composing fully normalised tables from a rigorous data dependency diagram. *Communications of the ACM*, **28**(8), 826–838.

van Roessel, J. W. (1987). Design of a spatial data structure using the relational normal forms. *International Journal of Geographic Information Systems*, **1**(1), 33–50.

Wilson, R. P. (1989). DB Drafter. Project Report, Department of Computer Science, University of Canterbury.