LensList:BrowsingandNavigatingLongLinear Information Structures

HongzhiSong,YuQi,YunLiang,HongxingPeng,and LiangZhang

HCI Group, College of Informatics,
South China Agricultural University, Guangzhou, 510642, China
Email: {hz.song,yuliangqi,yliang,xyphx}@scau.edu.cn,
200530740631@stu.scau.edu.cn

Abstract..List is a simple and useful graphical user interface (GUI) component for organizing linearly structured information when it contains only a few elements. However long lists are difficult to use because only a small part of the list is shown each time, to gain an overview or select elements from a long list the user usually needs to scroll the list many times. This problem becomes more serious as the list gets longer. This paper presents a novel solution, LensList, by applying the focus and context technique to the view of a list. LensList dynamically changes the elements to bigger sizes to form a focal area around the mouse cursor while keeping the elements in the peripheral area in smaller sizes as context. This enables it to display a longer list within the same screen area. Therefore it can be more efficient for performing browsing and navigation tasks.

Keywords: Information Navigation, List Navigation, Long List, Focus + Context, Multiple Foci

1 Introduction

List is a simple yet powerful mechanism for organizing linearly structured information. It is very useful when there are only a small number of elements in the list. However as the quantity of information grows, lists are getting longer and longer. A list sometimes contains tens or even hundreds of elements, such as lists used for selecting monetary exchange rates, countries, character encodings, fonts and favourite web sites etc. Long lists are not as usable as short lists even though they are normally equipped with scrolling mechanism. Because only a small part of the list is shown each time, to gain an overview or select some elements from a long list the user usually needs to scroll the list several times. For example, in an alphabetically ordered list of all countries, how can one choose such three countries as Afghanistan, Saudi Arabia and Yemen. Since the list is long, which contains 266 countries and areas, and also these three countries separate apart from each other in the list, selecting three of them needs more interaction. This navigation problem becomes more serious as the list gets longer.

One solution to the navigation problem mentioned above is to divide the elements in the list into different categories and change the list into a hierarchy. This would be a good solution when the elements can be well classified, which means the categories are intuitive and the user would easily know which element should be in which category. However this approach prolongs the access path to the target elements by providing intermediate elements. Moreover selecting many elements in different categories would be more difficult.

Another solution is to compress the elements into small sizes to form a compact view in order to show more elements at the same time. Although more elements can be shown, the sizes of the elements could be too small to read if the total number of elements is big.

Other solutions are also possible, such as dividing the list into several columns to display, but it does not save any screen space. This paper presents a novel solution, called LensList, by applying the focus and context technique to the view of a list.

2 Related Work

Focus and context (sometimes in the form of focus + context, also often interchangeably referred to as the fisheye distortion technique) has been existing for many years since first introduced by Furnas in 1986 [1]. In that paper the cognitive aspects of how people view and remember information was discussed. The fisheye distortion technique was then applied to a variety of applications [2–5]. Several variations of the fisheye technique have been explored. They have been used in one dimension for word processing [6], access to time [7], and for long lists [8,9]. They have been used in two dimensions for tables [10], graphical maps [11] and space scale diagrams [12]. They have even been used in three dimensions for document browsing [13]. Some applications of fisheye distortion techniques have been carefully evaluated [14], often finding a significant advantage to fisheye views [15–17]. Masui's work [8,9] is more relevant to ours in terms of handling long lists. However they adopted the overview + detail strategy, which is essentially different to the focus and context method.

3 Objectives

Despite the careful investigation of fisheye distortion techniques, and their application to a broad set of complex tasks, fisheye views have never been applied to the widely used lists. By introducing fisheye distortion, LensList becomes fundamentally different from a traditional list, because every element may change size and they are not static anymore. This makes the design of LensList more difficult than traditional lists.

The goal of this work was to alleviate the browsing and navigation difficulty of traditional GUI lists in handling long linear information structures. To achieve this goal, four objectives were set up as below:

1. A GUI component was to be designed which should be compatible with traditional lists in functionality. This is helpful in reducing the users' learning time.
2. An efficient graphical fisheye distortion algorithm should be implemented, which relates the mouse movement to locating the visual focus. Since mouse event listening is not a fast process, an efficient algorithm is necessary to guarantee the smooth rendering of screen scenes.
3. The screen space occupation of this component ought to be fixed in size, no matter the focus located anywhere. This is to ensure there is no overlapping with other on screen components.
4. Multiple foci selection should be supported by this component so that working on multiple elements simultaneously is possible.

4 Methods

In LensList, elements near the focus are displayed at larger sizes, and elements further away from the focus are displayed at smaller sizes. In addition, the interline spaces between elements are also increased in the focal area, and decreased further away from the focal area. In this manner, more elements can fit into the same screen area. The elements are dynamically scaled so that as the cursor moves, a "hump" of readable elements moves with the cursor. The "hump" is like a magnifying lens over the list, and it is thus named as LensList. The effect can beseen in Figure1.

4.1 Locating Focus

LensList changes elements sizes by a distortion function, and the distortion function is based on the location of focus. Thus how to locate the focus becomes an important issue. A simple way is to relate the focus with the mouse cursor. There are two parameters to determine the position of the cursor, p, horizontal coordinate p_x and vertical coordinate p_y. P_x does not matter in LensList, because a list displays elements in vertical manner, the horizontal position of an element has nothing to do with distortion. P_y is the one that the designer should be concerned with. An element's position p_e, also has two coordinates p_{ex} and p_{ey}, again it is the p_{ey} that is relevant to distortion. P_{ey} is the parameter to determine the focus, but it has to be acquired through cursor position, p_y. When p_{ey} is equal to p_y, the element at p_e is selected as the focus.

4.2 Degree of Interest

The common approach to implementing fisheye distortion is to compute a "Degree of Interest" (DOI) function for each element to be displayed. The DOI function calculates the element's size. A typical DOI function includes both the distance of an element from the focal point as well as the element's a priori

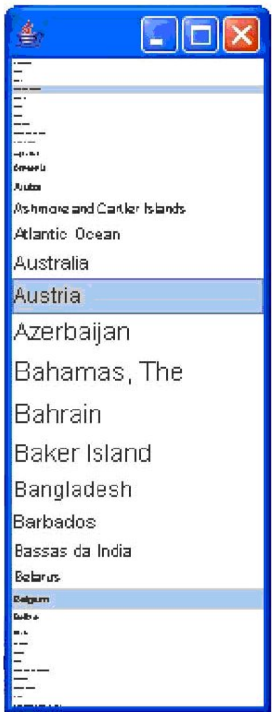


Fig. 1. LensList displaying a number of countries with three countries being selected.

importance [1]. Thus certain landmark elements may be shown at a large size even though they are far from the focal point.

The fisheye view of LensList was controlled by a simple Degree of Interest (DOI) function. The two parameters that determine the degree of the lens effect are the maximum and minimum sizes of the elements. LensList uses a simple DOI function to calculate elements' sizes. The function is shown in Figure2, the horizontal axis represents the distance to focus, and the vertical axis is the font size. It keeps the element at the focal point at the maximum size. Then elements get smaller, one point in font size at a time until the minimum font size is reached at which time, all more distant elements stay at the minimum font size.

The user may change two parameters of the DOI function, i.e. the maximum font size and minimum font size. Since elements away from the dynamic focus get smaller one point in font size at a time, the slope of the function is constant. Increasing or decreasing the maximum font size would also change the distortion distance. The result can be seen in Figure 2(a). The middle line is the initial distortion function.

When increasing the maximum font size from 6 to 7, the distortion distance increases by 2, from -4~4 to -5~5. The function changes to the top line. The bottom line is the function after changing the maximum font size from 6 to 5. Changing the minimum font size has the opposite effect. The distortion function will change as shown in Figure 2(b). The middle line is the initial distortion function. When increasing the minimum font size from 2 to 3, the distortion distance decreases by 2, from -4~4 to -3~3. The function changes to the top line. The bottom line is the function after changing the minimum font size from 2 to 1.

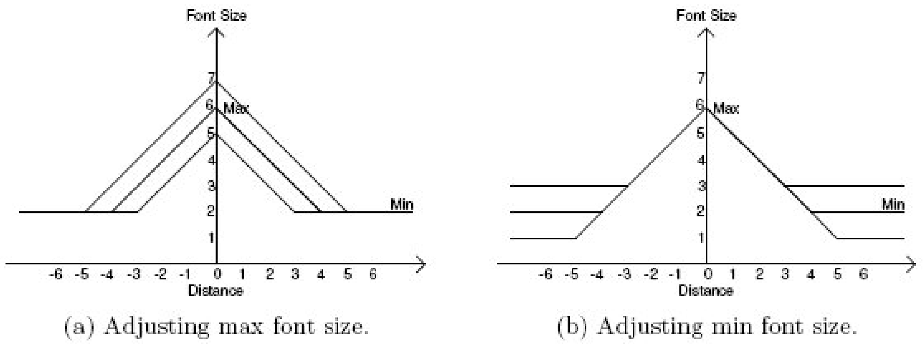


Fig. 2. LensList DOI function.

The graphical fisheye algorithm uses an array to store the sizes of all the elements. When the focus changes the array is updated accordingly. The elements sizes are retrieved from the array when elements are painted on the screen.

4.3 Multiple Foci of Selection

Locating foci by the mouse cursor is a practical solution for a single focus situation. However it would be different to achieve multiple foci because there is only one cursor.

The first concern is how to acquire the other foci if the first focus is obtained using the mouse cursor. Since it is not practicable to set predefined foci, the number and location of foci must be chosen by the user interactively. As the first focus and the rest of the foci are acquired differently, two terms are defined to distinguish them. The first focus is acquired through the mouse cursor, and it changes whenever the cursor moves. It is therefore called *dynamic focus* (only one). The other foci are selected by the user, and their positions do not change with cursor movement. They are hence named as *static foci* (may be many).

The second concern is whether all the foci should apply the same distortion function. If they all apply fisheye distortion, more space will be taken to accommodate the magnified elements. One of the objectives of LensList is to increase information density of the traditional lists. If all the foci use the same distortion function as the dynamic focus, this objective would be compromised. To keep high information density only the dynamic focus uses fisheye distortion.

Elements at static foci only change to highlighted color without magnifying the peripheral elements. The effect can be seen in Figure 1. Element *Bahamas* is the current dynamic focus. The three elements in highlighted color are selected as static foci.

When an element is selected by the user, it changes from the dynamic focus to a static focus, and highlighted in different color until it is deselected. Alternatively we can also think of it as a landmark, because it is set at a high a priori importance. In Figure 1, the static focus *Austria* is in the distortion range of the dynamic focus *Bahamas*, but there is no conflict between them because they are emphasized by different mechanism. This solution not only keeps high information density but also avoids conflict between foci.

4.4 Length Compensation

As the focus moving towards one end of the list, the length of the focal area is decreasing. The minimum length, half of the usual length, reaches when the first or last element becomes the focus. In this process the tail of the list will change its position, either moving up or moving down. We call this phenomenon "waving tail" effect for easy understanding. The screen space occupation algorithm compensates the length of the focal area by increasing the elements sizes on the opposite side of the focus moving direction. Hence the screen space that LensList occupies is fixed. The effect can be seen in Figure 3. As the mouse cursor moving from top to bottom, the length of the list stays unchanged.

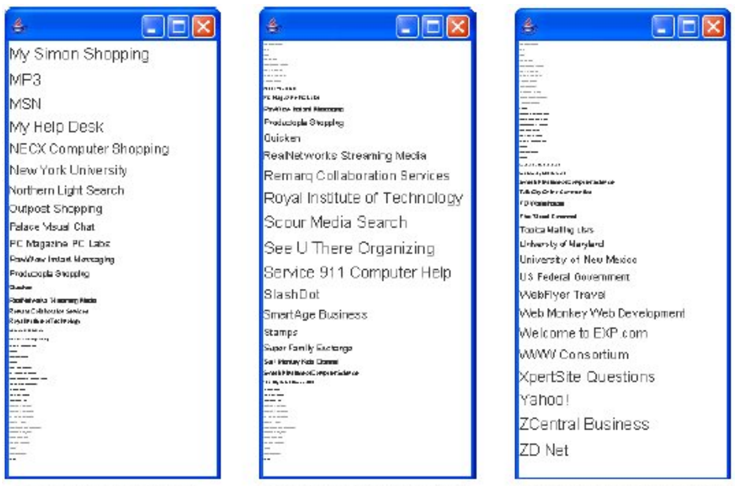
5 Implementation

LensList was developed in Java. It subclasses the Java *JList* component in the Swing GUI toolkit, so it can be used as a replacement for *JList*. The example used to generate Figure 1 is the list of countries. While the system is up and running, it responds to user interaction immediately. The time lag for computing the distortion and refreshing the screen is unnoticeable.

6 User Test

A couple of users were invited to try the LensList tool, and they were all experienced computer users. The trial was more subjective than objective, and the goal was mainly to find user preference of the new tool comparing to its traditional peers. The task was set to browse the list of countries and a list of web sites.

The users showed more preference towards LensList comparing to the traditional list. Some of them thought the LensList view was "visually appealing". This could be caused by their first image of the fisheye effect in graphical user in-terfaces. Most users thought that it was easier to gain an overview with LensList since it was able to display a longer list. Multiple foci was also thought to be a good feature because more elements in the list can be processed in a batch.



(a) Focus at the top. (b) Focus in the middle. (c) Focus at the bottom.

Fig. 3. Moving focus on LensList from top to bottom.

Moreover, some users stated that it was easier for them to concentrate on the foci without being disturbed by other elements.

A problem was also found, which was the difficulty in selecting an element when the minimum size of elements was set too small. This was an important issue in the literature. Because it is a common problem for almost all fisheye distortion based visualization techniques, a good solution is yet to be proposed.

7 Discussion

The goal of this project was reached by fulfilling the four preset objectives, which are discussed as follows:

1. The GUI component, called LensList, was implemented as a subclass of the Java JList component. The functionality of JList is fully supported in LensList, and it can be generally used as a substitute of JList.
2. The graphical fisheye algorithm had been designed twice and modified several times. The current version is O(n) in memory space consumption. The time efficiency is O(1) and dependent on the processing speed the mouse event. It was believed that this algorithm is optimized and can be used on any linear data structures.
3. The length compensation algorithm was designed independently from the graphical fisheye algorithm. This algorithm effectively solved the unexpected “waving tail”

effect. It works well in one dimensional fisheye views, and can be extended to two or three dimensions.
4. Multiple foci selection was implemented in LensList. This is a useful feature when the user wants to process many elements altogether.

8 Conclusion and Future Work

Browsing and navigating lists are important and frequent tasks in modern GUIs. Long lists bring challenges to the traditional GUI list component. Fast navigation is becoming more expected with the increasing quantities of information to be processed. LensList, an enhanced GUI list is presented in an attempt to gain better performance in performing this type of tasks. LensList pursues this goal by introducing the focus and context technique, fisheye distortion, into its view. LensList dynamically changes the elements to bigger sizes to form a focal area around the mouse pointer while keeping the elements in the peripheral area in smaller sizes as context. This enables it to display a longer list than the traditional lists within the same screen area. Therefore scrolling is less often used and it can more efficient for performing browsing and navigation tasks.

The merit of this technique had been seen at this stage, therefore it was planned to continue studying this technique by conducting a series of task oriented user evaluations and refinements.

Acknowledgements. This work was jointly sponsored by the National Science Foundation of Guangdong Province of China under grant number 06300433, and Talent Introducing Fund from South China Agricultural University under grant number 2005K099. The feedback and suggestions to improve LensList from voluntary users are appreciated.

References

1. Furnas, G.W.: Generalized fisheye views. In: Proc. ACM CHI ’86, Boston, Massachussetts, USA, ACM Press (1986) 16–23
2. Dill, J., Bartram, L., Ho, A., Henigman, F.: A continuously variable zoom for navigating large hierarchical networks. In: Proc. IEEE International Conference on System, Man and Cybernetics, San Antonio, Texas, USA, IEEE SMC Society Press (1994) 386–390
3. Mitta, D., Gunning, D.: Simplifying graphics-based data: Applying the fisheye lens viewing strategy. Behaviour & Information Technology 12(1) (1993) 1–16
4. Spence, R., Apperley, M.: Database navigation: An office environment for the professional. Behaviour & Information Technology 1(1) (1982) 43–54
5. Spenke, M., Beilken, C., Berlage, T.: FOCUS: The interactive table for product comparison and selection. In: Proc. User Interface and Software Technology (UIST), Seattle, Washington, USA, ACM Press (1996) 41–50

6. Greenberg, S., Gutwin, C., Cockburn, A.: Sharing fisheye views in relaxed-wysiwig groupware applications. In: Proc. Graphics Interface (GI '95), Toronto, Canada, Morgan-Kaufmann (1995) 28–38
7. Koike, Y., Sugiura, A., Koseki, Y.: Timeslider: An interface to time point. In: Proc. User Interface and Software Technology (UIST '97), Banff, Alberta, Canada, ACM Press (1997) 43–44
8. Masui, T.: LensBar: Visualization for browsing and filtering large lists of data. In: Proc. IEEE Symposium on Information Visualization '98, North Carolina, USA, IEEE Computer Society Press (1998) 113–120
9. Masui, T., Minakuchi, M., Borden, G.R., Kashiwagi, K.: Multiple-view approach for smooth information retrieval. In: Proc. User Interface and Software Technology (UIST '95), Pittsburgh, Pennsylvania, USA, ACM Press (1995) 199–206
10. Rao, R., Card, S.K.: The table lens: Merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In: Proc. Human Factors in Computing Systems, CHI '94, Boston, Massachusetts, USA, ACM Press (1994) 318–322
11. Sarkar, M., Brown, M.H.: Graphical fisheye views of graphs. In: Proc. CHI '92: Human Factors in Computing Systems, Monterey, California, USA, ACM Press (1992) 83–91
12. Furnas, G.W., Bederson, B.B.: Space-scale diagrams: Understanding multiscale interfaces. In: Proc. ACM CHI '95, Denver, Colorado, USA, ACM Press (1995) 234–241
13. Robertson, G.G., Card, S.K., Mackinlay, J.D.: The document lens. In: Proc. 1993 ACM User Interface Software and Technology, New Orleans, Louisiana, USA, ACM Press (1993) 101–108
14. Cockburn, A., Karlson, A., Bederson, B.B.: A review of focus and context interfaces. HCIL Tech Report 2006-09, Department of Computer Science, University of Maryland, College Park, MD 20742, USA (2006)
15. Donskoy, M., Kaptelinin, V.: Windows navigation with and without animation: A comparison of scrollbars, zoom and fisheye view. In: Proc. Extended Abstracts of Human Factors in Computing Systems ACM (CHI '97), Atlanta, Georgia, USA, ACM Press (1997) 279–280
16. Hollands, J.G., Carey, T.T., Matthews, M.L., McCann, C.A.: Presenting a graphical network: A comparison of performance using fisheye and scrolling views. In: Proc. 3rd International Conference on Human-Computer Interaction, Boston, Massachusetts, USA, Elsevier Science Press (1989) 313–320
17. Schaffer, D., Zuo, Z., Bartram, L., Dill, J., Dubs, S., Greenberg, S., Roseman,M.: Comparing fisheye and full-zoom techniques for navigation of hierarchically clustered networks. In: Proc. Graphics Interface (GI' 93), Toronto, Ontario, Canada, Canadian Information Processing Society (1993) 87–96