Usability and improvements for linear and 2D fisheye techniques

Benoît Baliguet
Department of Software Engineering
The University of Auckland
Private Bag 92019
Auckland, New Zealand
bbal019@ec.auckland.ac.nz

ABSTRACT
Fisheye in human-computer interaction is a graphical visualization technique to browse large datasets of information. It uses a spatial distortion of the fisheye lens to allow a user to see both the surrounding context and a focus area on the same view. The purpose of this technique is to locate the focus area in its context to improve user experience in browsing or searching elements in a large dataset, with a smooth transition between the focus and its context. Fisheye view can be implemented on linear browsing, for example in list in menus or in two dimensions browsing to explore a set of files or graphs. One of the most known implementation of fisheye techniques for non-expert is the Mac OS-X dock which displays a linear set of clickable icons as a quick access menu on the desktop. On this seminar, I will focus on the usability of these techniques. I will report on the formalization and the implementation of the fisheye technique in linear and two dimensions views, and introduce usability studies of this technique. This will lead to two points of interest concerning the usability of the fisheye view: the use of the context with textual content and the accuracy in choosing an element in the dataset. The seminar will finally present solutions and improvements related to these two points: the enriched context for textual dataset, the use of landmarks, the focus lock and the velocity coupling for fisheye views.

Keywords
Focus+context, distortion-oriented visualization, fisheye technique, usability.

INTRODUCTION ON FISHEYE TECHNIQUE
The fisheye technique is a method of graphical visualization for human-computer interaction. This method is based on the render photographs obtain when using a special lens called to “Fisheye Lens”. This lens is a wide-angle lens which produces hemispherical pictures (Figure 1).

Figure 1: Picture of a street, using a Fisheye Lens. Copyright 2000, Dan Slater (Slater, 2000).

The pictures generated by a fisheye lens shows at the same time a focus area in the middle of the picture, and a 90° view on each side of the focus area. In a standard picture, such an angle would result in a picture with infinite dimensions when projecting it on a flat surface. To be able to visualize such a picture, the lens induces a progressive distortion of the surrounding of the focus point: the level of details of the picture decreases with the distance from the focus point.

The fisheye technique in human-computer interaction is the use of this distortion to visualize a large dataset of information. A formalization of this effect is proposed by George Furnas (Furnas, 1982). In order to be able to display a dataset in a fisheye, he defines necessary properties of the content which need to be defined: a level of details (the “LOD” function) and a distance from the focus point (the “D” function). The level of details is defined so that a global element has a greater level of details than a detail element. Using these two functions, the content of a dataset can be associated with a degree of interest function (the "*DOI*" function). For an element *x*, and a focus point *p*:

$$DOI(x,p) = F(LOD(x),D(x,p))$$

F is chosen monotone increasing in the first argument, and decreasing in the second. Using this function, the interest of the user for an element can be quantified in relation with its global importance and its distance from the focus. Then the degree of interest can be linked to a display function (the "*DISP*" function) for each element *x* of the dataset and a focus point *p*:

$$DISP(x,p) = G(DOI(x,p))$$

If *G* is defined so that the sum of the *DISP* of the elements of the whole dataset is equal to the available surface on which the dataset should be displayed, a fisheye representation of the dataset is obtained.

The purpose of using this technique is the same as using a fisheye lens in photography: it enables the display of a focus area and its context in the dataset. Using the distortion, elements with a high degree of interest from the context can be displayed while keeping their spatial relation with the focus element. This improves the knowledge on the focus area, by giving some information on the relation between a focus element and the global context. This technique can be implemented on linear representations such as menu list (Bederson, 2000), or two dimensions to display graphs (Sarkar & Brown, 1994) or icon elements (Carr, Hedman, & Nässla, 2004).

An simple implementation of this technique is due to B. Bederson, who studies the usability of the fisheye menu (Bederson, 2000) for displaying alphabetically ordered menu list. This implementation is performed using a linear fisheye technique which aims to enrich the user experience with the position of the focus element in an alphabetic reference. Bederson defines the distance *D(x,p)* as the number of elements between the element *x* and the element *p*, and chooses a level of details *LOD(x)* equal for each element. He defines the degree of interest using the scheme of Figure 2.

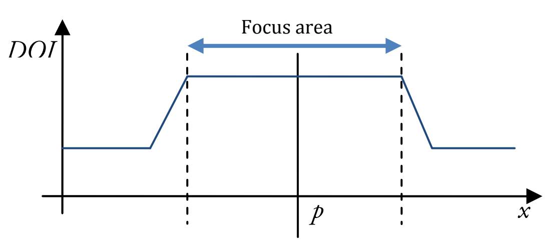

Figure 2 : DOI for linear fisheye in Bederson implementation. (Bederson, 2000)

Then, the *DISP* function associate a font size and a space between two elements to each value of the *DOI* function. The minimum and maximum associated font sizes are adjusted depending on the length of the focus area and the available space to display the menu. This implementation results in the visual aspect shown in Figure 3.

Figure 3: a fisheye menu in Bederson's implementation (Bederson, 2000)

The approach chosen by Bederson is a simple case of implementation of the fisheye technique which uses simple *D, DOI* and *LOD* functions. The two dimensions implementation allows much more complexity in the definition of the *D*, *DOI* and *LOD* functions.

A simple 2D solution if called the bifocal fisheye technique: the same rules as Bederson linear fisheye are applied both on vertical and horizontal scale. The use of the bifocal fisheye with a focus area reduced to one element and no transition area results in this fisheye view for thumbnails displayed in Figure 4.

Figure 4: simple bifocal fisheye view (Carr, Hedman, & Nässla, 2004)

But much more complicated implementations of the fisheye technique exist, especially for graphs visualization. For example, Sarkar-Brown fisheye visualisation (Sarkar & Brown, 1994) commonly used to display graphs defines the D, LOD, DOI and $DISP$ according to various external parameters: a distortion factor for the position of the element, a scale factor for their size, a minimum level of visual worth to control the amount of information displayed, the influence of the *a priori* importance of each node, etc. This extended implementation allows a better control of the repartition of the nodes of a graph in a fisheye representation (Figure 5).

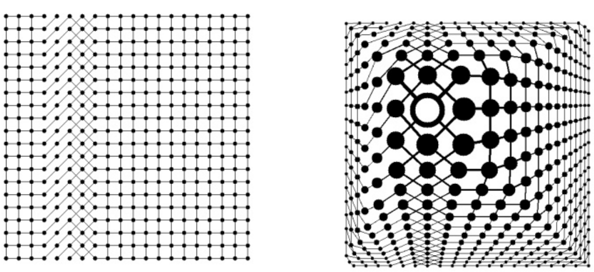

Figure 5: flat and Sarkar-Brown fisheye visualization of a graph (Sarkar & Brown, 1994)

This last implementation shows a visual aspect that is closer to what someone can obtain using a fisheye lens, but implies at the same time more computation to calculate the size and the position of the different elements.

THE USABILITY OF FISHEYE TECHNIQUE

In these implementations, the fisheye view is supposed to improve information seeking and data browsing, as the user can enhance his search with a mental visualization of the localisation of the desired element in the global context provided by the list. In this context, the fisheye technique has to be compared with known efficient methods used in dataset visualization.

For a linear dataset, the concurrent solutions are mainly based on the display of a zoom area of the list with handling functions, such as a scrollbar or arrows. Another solution is the hierarchical menu which can be used if the elements of the dataset can be categorized (Bederson, 2000). Concerning the two dimensions visualization, there is no equivalent for the hierarchical menu but the zoom solution can be implemented with controls to move the zoomed area on the vertical axes and the horizontal axes. A zoom variation solution, where the user can zoom out on the dataset and then zoom in on a precise area is also a possible solution for visualizing large areas of data (Carr, Hedman, & Nässla, 2004).

A compared study performed by B. Bederson (Bederson, 2000) on the usability of a list of elements alphabetically ordered with scrollbar, a hierarchical menu and a linear fisheye proves that the performances of the fisheye menu were close to the performances of the hierarchical menu and that it was better than the scroll bar solution. But it appears that during this study some users have problems using this kind of interface to browse a dataset, without any explanation concerning the reasons of these difficulties.

The later usability study by M. Hertzum and K. Hornbæk of Bederson's fisheye menu compared to the hierarchical solution allows a more precise description of the behaviour of the user while he uses a fisheye visualization, and reveals some weakness of the simple implementation of the fisheye visualisation (Hertzum & Hornbæk, 2007).

The first point of this study is that Bederson's fisheye menus can be interesting as a browsing method, but the amount of information provided by the context is very weak, whereas it is supposed to be used as a guide by the user when browsing the dataset. The basic fisheye representation is indeed limited by the nature of the elements it displays. On a picture, the reduction of the size is efficient because a picture allows different meaning depending on its degree of details. For example, a zoom on a picture of a forest reveals a single tree: two degrees of details have different meanings – "tree" vs. "forest" – and there is a continuous transition between the two elements. This is not the case for a list of text elements. The "zoom-out" action on a textual element results in the not readability of this element, and the sum of non-readable elements is indeed not readable. In the end, the only information the user can retrieve from the context is the position of the element in the dataset, which is only useful if the menu list displays a perfectly ordered dataset. This leads to the question of the interest of keeping such a context around the focus area as it has not the efficiency it should have.

The second point comes from the results of the usability study on the fisheye technique applied on list: the accuracy and the time to select an element in the list are higher in the case of the fisheye view than in the hierarchical menu, even if the user knows precisely the position of the element he is searching for. In particular, most of the errors made while selecting an item in the fisheye view were due to some difficulties in clicking on the final item after finding it.

This point is confirmed by Hertzum and Hornbæk analysis of the time the user spends on each part of his search of a precise element.

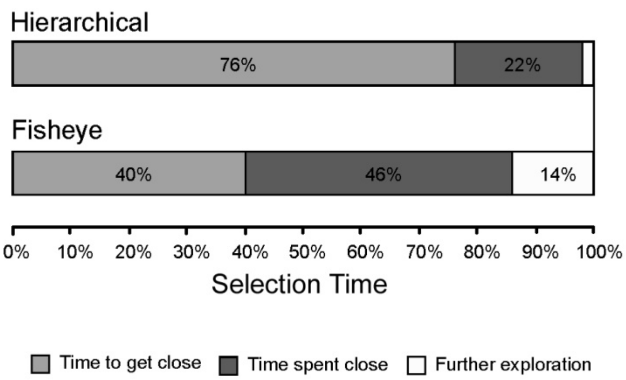

Figure 6: Relative distribution of time to select an item in the menus (Hertzum & Hornbæk, 2007)

In Figure 6, the time to get close corresponds to the time when user is searching the element. The time spent close is the time to reach the element after finding it. Further exploration is the time spent moving away the target element after entering its vicinity. On a hierarchical menu, when the user as find the element, most of the work is done, whereas on the fisheye technique a lot of time is wasted around the desired element.

This problem of accuracy is mainly explained by the way fisheye views are handled. The focus is directly linked to the vertical position of the mouse in the menu, and the whole length of the displayed dataset can be browsed in a single movement of the mouse on the menu. This implies that when the number of elements in the dataset increase, the difference of vertical position between two elements decrease. At a certain point, just a small movement off the mouse can induce a big change in the position of the focus area in the dataset. In the contrary, in a hierarchical menu the browsing is divided in two or more distinct mouse movements: first of all, the user chooses the appropriate category, which is equivalent to a rapid browsing of the database, and then chooses the target element in a sub-menu. With one level of hierarchy the user can virtually use two different speeds to browse the database: high speed on the top level and precise browsing in the menu. The lake of speed adjustment has indeed a lot of influence on the problem of accuracy when selecting an item in a fisheye view.

Another element brings incertitude in the choosing of an element in the fisheye view: the distortion. For the hierarchical menu, as soon as the sub-menu is open the target element remains in a constant position. This means that the user can predict the movement he has to make with his mouse to reach the target. This action is performed easily especially because such an action is often practiced by the user in human-computer interaction: people are used of clicking on static icons. In the case of the fisheye visualization, the target element is moving while the mouse is aiming to it, due to the distortion effect. With such behaviour of the elements of the menu, the user has difficulties in predicting the final position of its mouse and this decreases his efficiency.

The two problems – accuracy and use of the context – I describe below were underline on a linear implementation but the same issues appear for two dimensional representation. On 2D visualization, the position of the focus is linked to the vertical and horizontal position of the mouse which implies a direct relation between the sensitivity of the mouse and the speed of browsing. A compared study on 2D visualizations by D.A. Carr, A. Hedman and H. Nässla shows that this results in the same problem of accuracy (Carr, Hedman, & Nässla, 2004).

For graph node the user is confronted to the un-readiness of elements as the sum of unreadable nodes has no meaning, like text elements. In this case, the non-display of the context becomes critical. In a linear browsing, if the user starts from the beginning of the list and browses it until the end, he will necessarily encounter the element he is looking for. For the 2D graph display, the user can go from a node to another without finding the target element: the context is not just a guide but necessary information to allow an efficient use of the visualization.

To conclude, the points of interest concerning the usability of fisheyes visualizations are the use of the context and the accuracy in choosing an element.

ENRICHING THE CONTEXT OF A FISHEYE TECHNIQUE

The question of enriching the context of the fisheye visualization is especially relevant in the case in which the elements to display do not natively allow the "zoom-out" action.

A first approach for enriching the content is to define some degree of importance for the element in the dataset to be able to specify the LOD function on each element. Then the DOI function can be improved so it would take in consideration the *LOD* function (Hertzum & Hornbæk, 2007). In the case of a list of elements that can be displayed by a hierarchical menu, top elements can be defined in the structure of the dataset. Then the context can enrich by using the *DOI* function shown in Figure 7.

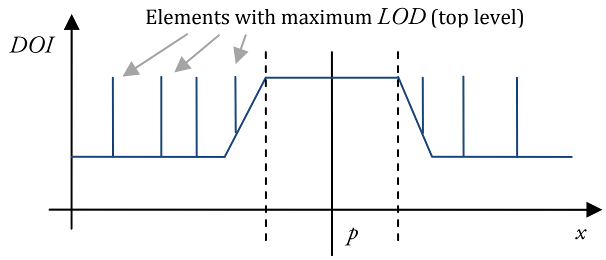

Figure 7: Improved DOI function (Hertzum & Hornbæk, 2007)

If a second level of importance can be defined in the dataset, with a lower *LOD*, the *DOI* function of Figure 7 can be improved again in order to display the second level element around the focus area (Figure 8).

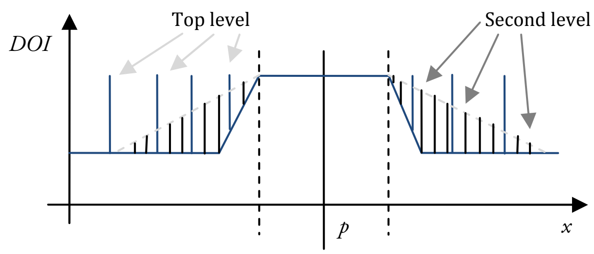

Figure 8: DOI improvement with two levels of details (Hertzum & Hornbæk, 2007)

With this new *DOI* function a new linear fisheye display is obtained where the context has more than just a position meaning. In fact this solution defines what the dataset should look like on the "zoom-out" action. With the three levels of details (top level, second level, and other elements) the last solution improves at the same the usability of the transition area, between the context and the focus area. This last point justifies the use of a smooth transition between the content and the focus area, as this area was previously as useless as the context. However, the usability study on this solution by Hertzum and Hornbæk shows the enriching the context as no significant effect on the efficiency of this solution compared to a basic fisheye list (Hertzum & Hornbæk, 2007).

The same improvement can be made on the bifocal fisheye technique, but is already implemented in the Sarkar-Brown fisheye visualization as this visualisation take in consideration the *a priori* importance of each element to calculate the size of the corresponding node in the distort representation (Sarkar & Brown, 1994).

However, as describe in the previous chapter, enriching the context of the two dimensional fisheye technique is a critical task as the search in a plan surface cannot be performed without contextual information. In the case of a flat display of a graph, the user bases his search of an element on a mental map of the graph (Gutwin & Skopik, Finding Things in Fisheyes: Memorability in Distorted Spaces, 2003). The user remembers the absolute position of some elements and can find an element using this knowledge and the spatial relation between the nodes. In the case of the fisheye visualization, the distortion of space induces damages in this mental map: there is a loose of the spatial meaning of the context of the target with the distortion, whereas this spatial context is the main guide in the search of a target.

In the case of the Sarkar-Brown visualization, the level of details of the different nodes helps the user in his search of an element. This level of details is contingent upon the total available space for the display of the graph, so most of the details are shown in the vicinity of a node. Nevertheless the search of an element may need information of the nodes which are not in the vicinity in order to be able to use the mental map of the graphs. In order to fulfil this requirement, C. Gutwin and A. Skopik propose to enrich to context using landmarks (Gutwin & Skopik, Finding Things in Fisheyes: Memorability in Distorted Spaces, 2003).

On previous stage the *LOD* function was improved, in this case the *DISP* function is modified so that it modifies the visual aspect of the nodes so that some nodes become landmarks. These landmarks can be:

- A specific colour for a node or a group of node
- A specific shape of a group of node
- The position on the graph of a node : on the edges / on a corners
- A composite of landmarks (ex: shape + colour)

The main point is that a landmark remains understandable after a "zoom-out" action because it is based on a visual improvement of the nodes. In particular the usability study on landmarks performed by Gutwin and Skopik shows that the colour landmarks were proved to be a useful improvement of the context (Gutwin & Skopik, Finding Things in Fisheyes: Memorability in Distorted Spaces, 2003).

IMPROVING THE ACCURACY IN FISHEYE TECHNIQUES

Improving the focus targeting can be done in several ways. The first solution is the copy the behaviour of the hierarchical menu by implementing on the fisheye technique a way to control the speed of browsing.

This can be done easily on the linear fisheye list: in the linear implementation only the vertical position of the mouse is used to control the scrolling of the dataset. Then a solution is to use the horizontal position as a speed controller.

For example, Bederson's implements in his fisheye menu a feature called the "focus lock" mode (Bederson, 2000). This consists in adding an area on the right side of the focus area. When the mouse enters this area, the link between the mouse position and the focus area index is broken and the user can choose an element easily as their position remains constant. However this solution implies two restrictions. First of all, Bederson's usability study on this menu shows that this feature is not intuitive whereas the use of the fisheye menu is quite obvious (Bederson, 2000). This means that the fisheye list must be delivered with a small explanation of how it works. Then this solution uses the horizontal position of the mouse, so this cannot be extended to the 2D visualization. Using such a solution on a bi-dimensional fisheye view would require further controls such as keyboard controls, which would increase the complexity of use of this technique.

However, C. Gutwin postulates that the problem can be solved if the fisheye view can understand the motivations of the user (Gutwin, Improving Focus Targeting in Interactive Fisheye Views, 2002). When a user searches for an element, his search can be divided in two distinct stages: *motion*, where the user browses the dataset to find the target, and *acquisition*, where the user positions his pointer on the target and selects it.

The only element available to make the distinction between these two stages is the movement of the mouse. The analysis by C. Gutwin of the velocity of the pointer returns the results shown in Figure 9.

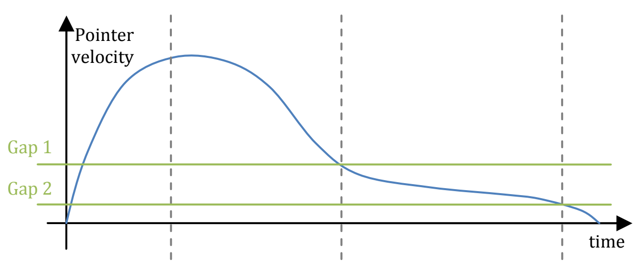

Figure 9: velocity profile for the selection of an item in the fisheye view (Gutwin, Improving Focus Targeting in Interactive Fisheye Views, 2002).

First of all, the velocity of the pointer increases to reach a maximum, and start decreasing as the user finds the target. Then the user enters the vicinity of the element he wants to choose, which corresponds to the part of the profile where the velocity is slowly decreasing. Finally the user selects the target and the velocity falls to 0.

Using this profile the *motion* and *acquisition* stage can be defined according to the velocity and acceleration of the pointer. While the velocity increases and then while it is above a "Gap 1" value, the user is in stage *motion* so he needs to browse the dataset with high speed. Then while the velocity is between "Gap 1" and "Gap 2", the user in the *acquisition* stage, close to the target, so he need low speed to be able to select the item. Finally, the user selects the item.

Using this profile, a focus lock mode can be implemented on the fisheye visualization with any number of dimensions. A solution could be to activate the focus lock mode as soon as the user enters the acquisition stage and deactivate it when the user selects the item.

Another solution is to link the distortion factor to the velocity of the pointer. The problem of accuracy is due to the distortion effect, so decreasing the factor of distorting improves the accuracy of the selection of an item. For the Sarkar-Brown visualization such a possibility can be directly implements as this fisheye technique takes in consideration the distortion factor (Gutwin, Improving Focus Targeting in Interactive Fisheye Views, 2002). This velocity and distortion coupling is named the Speed-coupled flattening (SCF), as the velocity causes a "flattening" of the graph representation. The relation between the pointer velocity and the distortion level is shown in Figure 10.

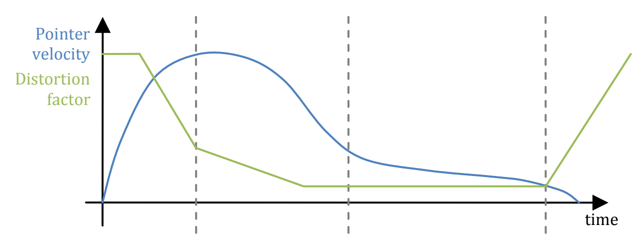

Figure 10: the relation between distortion and velocity of the pointer in SCF (Gutwin, Improving Focus Targeting in Interactive Fisheye Views, 2002)

In this case, the usability study performed by Gutwin on this implementation proves that adjusting the distortion factor in relation with the pointer velocity is an efficient solution to improve the accuracy and performances of the Sarkar-Brown visualization (Gutwin, Improving Focus Targeting in Interactive Fisheye Views, 2002).

SUMMARY

The first implementations of the fisheye techniques show two main points to improve the usability of this technique: the use and the importance of the context, and the accuracy when selecting an element in the dataset. The first point mainly concerns the efficiency of the technique, whereas the second point is the source user frustration.

The usability studies shows that enriching the context has little effect on linear view but that landmarks are efficient on the bi-dimensional visualization. In both cases the improvement is limited by the nature of the manipulated elements and the solution of visual enrichments is then the only efficient improvement.

Concerning the second point, the research of several authors shows that a lot of improvement can be done on this subject, either with additional controls or by analysing the user behaviour. This improvement are indeed compulsory has the nature of the fisheye view can induce some problems of use with people which does not perfectly control the mouse, which is a drastic limitation for industrial uses.

FUTURE WORK

To my point of view, the question of the use of the context seems to be quite complicated to resolve in the case of textual content. The use of the 2D landmarks on the linear fisheye menu can be studied, but I see no point why using these landmarks would improve the efficiency of the linear problem as Hertzum and Hornbæk already studied an enrichment which shows not improvement.

Concerning the accuracy of the fisheye, I mainly identify three points which worth to be studied. First of all the correlation between the velocity of the pointer and the stage in the search of an element could be extended to the linear fisheye and compared to the focus lock solution. As this solution seems to be efficient on two dimensions display, it could be interesting to try it on linear implementations.

The second interesting point could be an extension of the "focus lock" by implementing a progressive control of the distortion of the linear fisheye using the horizontal position of the pointer on the menu. This point should at the same time improve efficiency of Bederson's menu as the change of speed would be progressive.

The last point is an issue which is not studied by these authors: the behaviour of the fisheye technique on very large dataset. They point out the problem of accuracy but with a sufficient amount of elements in the dataset, fisheye view can generate a situation in which any movement of the mouse could change the focus point in a distance above the size of the focus area. This would mean that some elements could not be selected. A progressive speed could be a solution in this case, but a generalisation of the fisheye technique on any size of dataset could be an interesting start for an additional research.

REFERENCES

Bederson, B. B. (2000). Fisheye Menus. *Proceedings of ACM Conference on User Interface Software and Technology (UIST 2000)* (pp. 217-226). ACM Press.

Carr, D. A., Hedman, A., & Nässla, H. (2004). Browsing Thumbnails: A Comparison of Three Techniques. *Proceedings of the 26th International Conference on Information Technology Interfaces (ITI2004)*. Cavtat, Croatia.

Furnas, G. W. (1982). The FISHEYE view: a new look at structured files. *Bell Laboratories Technical Memorandum, #82-11221-22.*

Gutwin, C. (2002). Improving Focus Targeting in Interactive Fisheye Views. *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*. Minneapolis, Minnesota, USA.

Gutwin, C., & Skopik, A. (2003). Finding Things in Fisheyes: Memorability in Distorted Spaces. Conference *on Graphics Interface GI'03*. Halifax, Canada.

Hertzum, M., & Hornbæk, K. (2007). Untangling the Usability of Fisheye Menus. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *Vol. 14, Issue 2, Art. n°6.*

Sarkar, M., & Brown, M. H. (1994). Graphical Fisheye Views. *Communications of the ACM, Vol. 37, Issue 12*, pp. 72-83.

Slater, D. (2000). *Using the 6 mm Nikon fisheye lens with the Nikon D1 camera*. Retrieved 04 25, 2008, from http://www.nearfield.com.