Strotor: A Minimalistic Approach to Exploring Hierarchical Data

ABSTRACT

We introduce a new simple information navigation technique that enables users to explore large hierarchical data structure using only two keys or one rotating device that can generate two different signals based on the rotation direction. Using the rotation-based input device called "Strotor", users can find an entry in a huge hierarchical database easily only by rotating a disk or a cylinder in two directions. Strotor can be easily installed in sofas, kitchens, cars, etc. where standard keyboards and remote controllers do not fit.

KEYWORDS: Input device, Information navigation, Hierarchical data

INTRODUCTION

Large data are often represented as a hierarchical data structure, and various navigation methods are provided for exploring the data. For example, files on Unix are structured hierarchically, and various commands (cd, ls, etc.) and APIs are provided for exploring the file system. Large dictionary database can also be treated as a large hierarchical data, since the name of the entry can be treated hierarchically: e.g. an entry for "dictionary" can be stored under "d" and "d/i".

Many information visualization techniques and navigation techniques have been proposed for handling large hierarchical data. On personal computers, multiple GUI methods are provided for exploring the hierarchical file structure, because there is no best interaction technique that fits to all the users and situations.

When a user cannot use a pointing device, selection-based interaction techniques are used for finding information in hierarchical data. For example, a user can hierarchically find a music data by selecting an artist from the list of artist names, selecting an album title from the title list, and selecting a music data from the song list. Users can use two keys for choosing an entry from the list (e.g. selecting an artist from the artist list), and often another key is used for selecting an entry and moving to the next level (e.g. selecting an artist and show album titles). It is also necessary to provide a way to go back to the previous state, and yet another key can be used for that task (e.g. showing the artist list for artist selection). This 4-way navigation is popular on small mp3 players[1] and remote controllers like AppleRemote[2] for selecting a song from the hierarchical data. 4-way navigation is also available on desktop software like Finder.app on Mac.

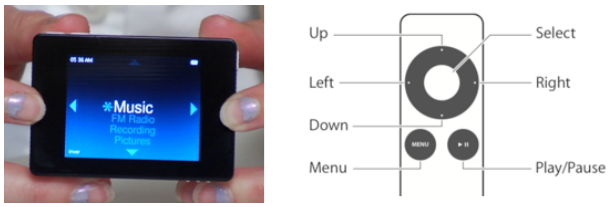

Figure 1: 4-button controllers on a small mp3 player (left) and AppleRemote (right).

Using 4 keys might be okay on these devices, but it would be much better if we could perform the navigation task using only 2 keys. In that case, we can use a rotating device like a disk or a cylinder for the navigation, since the device can generate 2 signals based on the rotation direction (Figure 2). We have developed a new navigation technique and device called "Strotor", where users can explore a large hierarchical database by simple rotating operations.

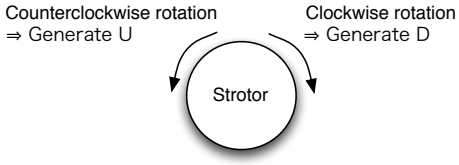

Figure 2: Using a rotation device for generating key events.

NAVIGATION METHOD AND EXAMPLES

Interaction with Strotor is based on the following simple principles. Two types of operations, "U" (up) and "D" (down),

[1] e.g. http://www.iriver.com/product/view.asp?pCode=003&pNo=37
[2] http://en.wikipedia.org/wiki/Apple_Remote

are used for the navigation. They can be generated either by rotating the Strotor device or typing a keyboard.

1. A portion of the hierarchical data is shown to the user.
2. One element in the list is selected and highlighted.
3. When an elemente is selected, its siblings, its ancestors and their siblings are displayed around the selected element.
4. Users can issue U to select the element above the currently selected element, or issue D to select the element below the selected element. Whenever the selection is changed, 3 is performed. If the depth of the newly selected element is different from the currently selected element, siblings of the currently selected element disappear.
5. When a newly selected element has children and the user performs no further action, the first child of the selected elemente is newly selected, and 3 is performed. As a result, all the siblings of the newly selected element (the first child of the currently selected element) appear in the list.

We show how Strotor works, using a hierarchical data structure of the shops list in a shopping mall shown in Figure 3 as example data. Rectangles with thick border represent categories, and other rectangles represent individual shops.

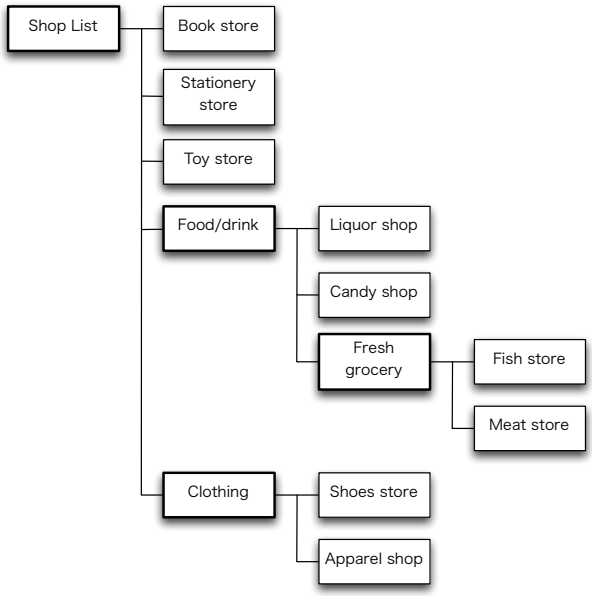

Figure 3: Sample data: shops in a shopping mall.

When a user starts the exploration, only the shops and categories at the top level are displayed (Figure 4). When the user issues D, the second element (Stationery store) is selected (Figure 5)[3].

[3]Readers can try this at http://jsbin.com/lecuroku.

Figure 4: Initial display.

Figure 5: Typing D.

If the user issues D two more times, Food/drink category is selected (Figure 6). If the user stops the operation and waits for a moment, the shops under the Food/drink category are automatically displayed, and the first entry (Liquor shop) is selected (Figure 7).

Figure 6: Selecting Food/drink.

Figure 7: Selecting Liquor shop.

When the user issues D twice here, Fresh grocery category is selected (Figure 8).

Figure 8: Selecting Fresh grocery.

If the user keeps issuing D, the list will change to Figure 9, without expanding the children of Fresh grocery.

Figure 9: Selecting Clothing.

If the use stops issuing D at Figure 8, the shops under category Fresh grocery is automatically selected (Figure 10).

Figure 10: Selecting Fish store.

When the user issues U here, Fresh grocery is selected, and the shops under Fresh grocery disappears, resulting in the same state as Figure 8. If the user issues U two more times, the display changes to the state shown in Figure 7, and one more U will set the system to the state of Figure 6.

If the user issues D in Figure 6, the next visible entry (Clothing) is selected (Figure 9), and then the state changes to Figure 11.

Figure 11: Selecting Shoes store.

When the user issues D twice in Figure 10, Clothing is selected, and the category under Food/drink will shrinke (Figure 9).

In this way, users can explore the hierarchical structure only by issuing U and D at the right timing.

IMPLEMENTATION

A prototype of Strotor was built using JavaScript on modern browsers. A large list of news articles, movies, anime films, musics, and e-book titles are listed in one browser window and the content is displayed in another window (Figure 12).

Figure 12: Selecting an Internet radio station.

We used PowerMate[4] from Griffin Technology for the rotating device. PowerMate generates standard keyboard events when rotated, and no special programming techniques are required. Users can explore all the contents just by rotating the device clockwise and counterclockwise.

Figure 13: A PowerMate used for the prototype.

[4]http://griffintechnology.com/support/powermate

DISCUSSIONS

Comparison with existing navigation methods

4-way navigation is commonly used for exploring hierarchical data structure, and it is familiar to computer users. Some mobile phones and PDAs are equipped with a jog dial with a push button, where the dial is used for choosing an item from the list and the push button is used for fixing the selection. All the existing methods require more than 2 keys/buttons, and to the best of the authors' knowledge, Strotor is the only interaction method for exploring hierarchical data structure with one rotating device.

Using a conventional hierarchical menu, child elements are displayed automatically when their parent element is selected by a mouse. The behavior is well understood by computer users, and Strotor's automatic transition (e.g. from Figure 6 to Figure 7) looks familiar to users.

Comparison with InfoVis techniques

Various information visualization techniques like Treemap[2], Hyperbolic Tree[3], and Sunburst[5] have been proposed for visualizing large hierarchical data. Zooming user interface (ZUI) systems like Pad[4] and Pad++[1] can also be used for handling large hierarchical data laid out in a 2D space. These visualization techniques and descendent technologies have become popular these days and all of these systems are useful for understanding the structure of large hierarchical data, but users of these systems have to use a pointing device to take full advantage of their advantages.

It seems to be a good idea to use Strotor on more conventional visualization techniques like TreeView[5], by replacing 4-way navigation to Strotor.

Timing control

The biggest disadvantage of Strotor is that the behavior of Strotor depends on the speed of the user's operations. In Figure 6, if a user wanted to select Clothing but couldn't issue D quickly enough, he will see Figure 7 instead of Figure 10. This is a tradeoff between usability and simplicity, and the best parameter should be set based on the user and the Strotor device.

EVALUATION

No formal evaluation have been done yet, but Strotor has been used in the author's living room for six months, and the author's family members are using it everyday for watching anime films and listening to music. Besides PowerMate, a wireless mouse is also used, since we can carry it anywhere and select a film or a music just by rotating the mouse wheel.

We demonstrated Strotor at an exhibition held in November 2013, and asked more than 100 people to try it (Figure 14). Since the only thing a user can do with Strotor is to rotate the device, users seemed to be able to understand the behavior of Strotor with trials and errors. Using only one rotating device for data navigation is a new experience for all the subjects, but the hard restriction of the device seemed to have worked in this case.

Figure 14: People at the right trying Strotor at an exhibition.

CONCLUSIONS

We have developed a new simple interaction method "Strotor" for exploring large hierarchical data structure. A Strotor user can easily find an entry in a huge hierarchical database only by using a rotating input device which can be installed at wide range of locations where conventional keyboards and switches do not fit. We are hoping to install various implementations of Strotor and try them at various places like kitchens, restrooms, etc.

REFERENCES

1. Benjamin B. Bederson and James D. Hollan. Pad++: A zooming graphical interface for exploring alternate interface physics. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, UIST '94, pages 17–26. ACM, 1994.
2. Brian Johnson and Ben Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd Conference on Visualization '91*, VIS '91, pages 284–291. IEEE Computer Society Press, 1991.
3. John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 401–408. ACM Press/Addison-Wesley Publishing Co., 1995.
4. Ken Perlin and David Fox. Pad: An alternative approach to the computer interface. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 57–64. ACM, 1993.
5. John Stasko and Eugene Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, INFOVIS '00, pages 57–65. IEEE Computer Society, 2000.

[5] http://en.wikipedia.org/wiki/Tree_view