No-click browsing of large hierarchical data

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—We introduce a novel simple interaction technique for finding and browsing data in a huge hierarchical database using only two keys.**

Finding a data from a huge hierarchical database takes time because users should traverse the tree using a mouse or a keyboard, and click a "select" button to check the content. If the data was not appropriate, he should go to the next entry and perform the selection task again.

In the past, when only a small number of "channels" were available from broadcast TV stations, users could simply rotate a television dial to select a channel and watch the program. We introduce a simple interaction technique called "Gear", with which users can select an entry from a large hierarchical database and browse it instantly, just like we could select a channel and enjoy a program using old television dials.

***Index Terms*—Input device, Information navigation, No-click browsing, Hierarchical data, Gear, Serencast**

I. Introduction

Huge amount of movies, musics, and other programs are available on the web, but finding and viewing a program is not easy because (1) finding a program takes time, and (2) a "confirm" operation (clicking something or typing a key) is required for viewing the content.

The confirm operations was not required on old TVs and radios, because we could enjoy TV/radio programs as soon as we rotated a TV channel dial or a tuning dial of the radio.

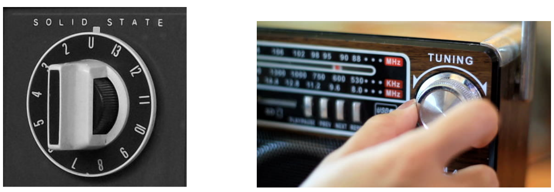

Fig. 1. An old TV channel dial and a radio tuning dial

It is always convenient to view the contents of the selected item as soon as we find it. We call such viewing style as "***no-click browsing***".

We can use the no-click browsing style on personal computers. Using the Finder.app on MacOS, we can use the "Cover Flow"[1] viewer to browse the contents in a folder with simple swiping gestures (Figure 2).

[1]https://en.wikipedia.org/wiki/Cover_Flow

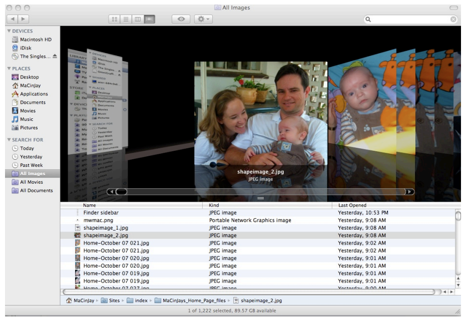

Fig. 2. Cover Flow on MacOS

We can also enjoy no-click browsing on the "column view" mode of Finder.app. In the column view mode, the contents of the file is shown at the right side of the window as soon as we select a file in a folder (Figure 3). When we type an arrow key and select another file, the contents is shown instantly without a confirm operation.

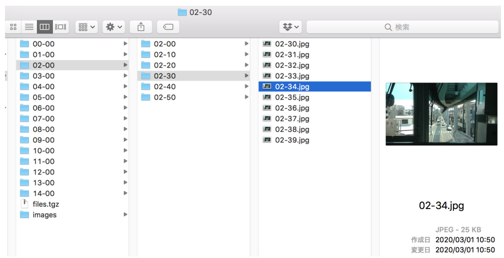

Fig. 3. No-click browsing on the "column view" of Finder.app.

We can use no-click browsing on old TVs and radios because the number of broadcast channels are limited. Likewise, no-click browsing on Finder.app is useful when the number of files in a folder is not very large.

No-click browsing is not popular on conventional GUI of personal computers. People are used to click-based browsing on web browsers and other information visualization systems. People can navigate through a huge information space by clicking links, zooming by dragging, and using other GUI techniques. These techniques are useful for finding an item in a large information space, but many clicks and drag operations are required, and the contents of the items are usually not shown during the search.

When a user cannot use a pointing device, key-based interaction techniques are used for finding information in hierarchical data. For example, a user can 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 title from the song list. Users can use two keys for choosing an entry from the list and use another key for selecting an entry and moving to the next level. It is also necessary to use another key for going back to the previous level. This 4-way navigation is popular on small devices and remote controllers for selecting a song from the hierarchical data. 4-way navigation is also available on the column view of Finder.app.

Fig. 4. Small devices supporting 4-way navigation

Using 4 keys might be okay on these devices, but it would be better if we could perform the navigation task using simpler devices that have only two keys. In that case, we can use a rotating device like a disk or a wheel for the navigation, since such devices can generate two signals based on the rotation direction.

We have developed a new navigation technique called "Gear" for no-click browsing, where users can explore a large hierarchical database by using only two keys.

II. Navigation Method of Gear

Interaction with Gear is based on the following simple principles. Two types of operations, ▲ (up) and ▼ (down), are used for the navigation. They can be performed either by pressing keys or rotating a dial.

1) A portion of hierarchical data is shown to the user as a list.
2) One element in the list is selected and highlighted.
3) When an element is selected, its siblings, its ancestors and their siblings are displayed around the selected element.
4) Users can issue ▲ to select the element above the currently selected element, or issue ▼ 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 element 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 Gear works, using a hierarchical data structure of a shops list in a shopping mall shown in Figure 5 as example data. A rectangles with a thick border represent shop categories, and other rectangles represent individual shops.

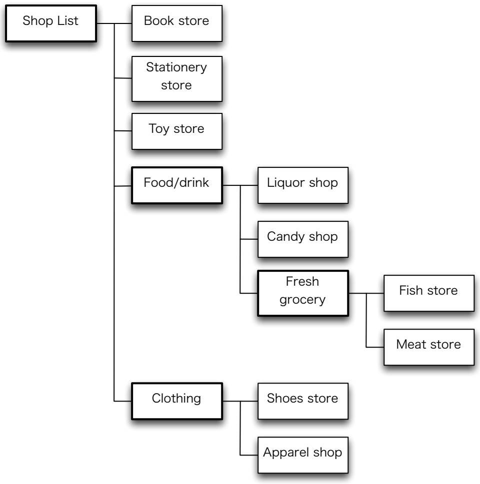

Fig. 5. Sample data: shops data of a shopping mall.

When a user starts the exploration, only the shops and categories at the top level are displayed (Figure 6). When the user issues ▼, the second element (Stationery store) is selected (Figure 7).

Fig. 6. Initial display.

Fig. 7. Typing ▼.

If the user issues ▼ two more times, Food/drink category is selected (Figure 8). 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 9).

Fig. 8. Selecting Food/drink.

Fig. 9. Selecting Liquor shop.

When the user issues ▼ twice here, Fresh grocery category is selected (Figure 10).

Fig. 10. Selecting Fresh grocery.

If the user keeps issuing ▼, the list will change to Figure 11, without expanding the children of Fresh grocery.

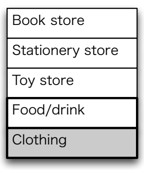

Fig. 11. Selecting Clothing.

If the use stops issuing ▼ at Figure 10, the shops under category Fresh grocery is automatically selected (Figure 12).

Fig. 12. Selecting Fish store.

When the user issues ▲ here, Fresh grocery is selected, and the shops under Fresh grocery disappears, resulting in the same state as Figure 10. If the user issues ▲ two more times, the display changes to the state shown in Figure 9, and one more ▲ will set the system to the state of Figure 8.

If the user issues ▼ in Figure 8, the next visible entry (Clothing) is selected (Figure 11), and then the state changes to Figure 13.

Fig. 13. Selecting Shoes store.

When the user issues ▼ twice in Figure 12, Clothing is selected, and the category under Food/drink will shrink (Figure 11).

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

III. Serencast: A No-Click Browsing Service

To prove that Gear is useful for selecting an entry from a large hierarchical database, we have developed the "Serencast" service, where users can enjoy movies, musics, and other web services on the browser with the Gear interface without selecting (clicking) an item. Serencast is implemented in JavaScript, and runs on modern web browsers

Serencast uses the "Scrapbox" wiki service[2] for listing the contents of the hierarchical database. Scrapbox is an interactive wiki system where users can share and edit texts on the web browser. A user can create a Scrapbox "project", and create pages that consists of texts and links. People can share the page and edit it interactively on their browsers just like the Google Docs system[3].

[2]https://Scrapbox.io/
[3]https://www.google.com/docs/about/

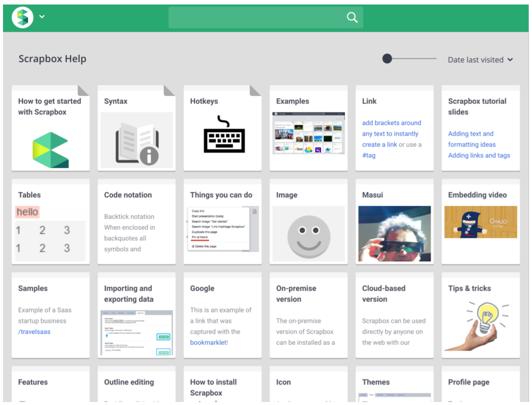

Fig. 14. An example of a Scrapbox project.

Figure 14 shows an example of a Scrapbox project. The title of the project is "Scrapbox help", and other pages like "Features" exist in the project.

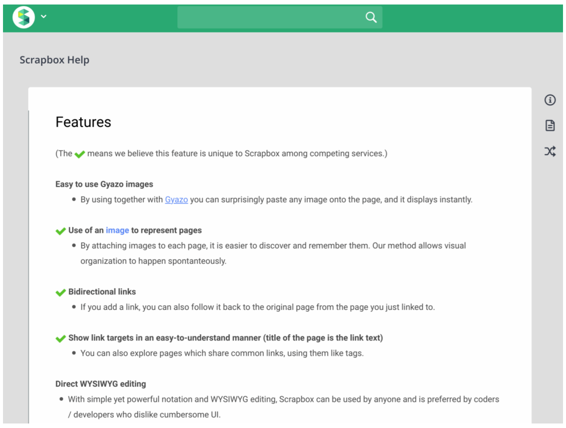

Fig. 15. An example Scrapbox page.

Figure 15 is an example of a Scrapbox page. The page is based on plain text just like an WikiPedia page is based on plain text. Users can use special markup tags like "[" and "]" to show bold texts, icons, URL links, and links to other pages. In this example, a URL link to http://gyazo.com is used just like an A tags in HTML.

The raw text of the Scrapbox page looks like this:

```
Features
(The [Check.icon] means we believe this feature is unique \
 to {\SB} among competing services.)

[[Easy to use Gyazo images]]
 By using together with [Gyazo https://gyazo.com] you can \
 surprisingly paste any image onto the page, and it \
 displays instantly.

[Check.icon] [[Use of an [image] to represent pages]]
 By attaching images to each page, it is easier to discover \
and remember them. Our method allows visual organization to \
happen spontaneously.
```

When we list the URLs of Web pages of movies and musics on a Scrapbox page, We can use Serencast service for using the Gear interface to select a program from the list on Serencast pages, and automatically play it on the browser.

Figure 16 shows the movie list defined in a Scrapbox page.

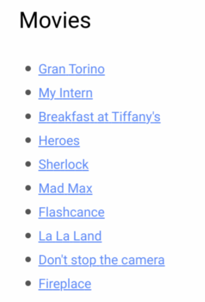

Fig. 16. A movie list page on Scrapbox.

The raw text of this page is shown below:

```
Movies
 [https://www.netflix.com/watch/70105600 Gran Torino]
 [https://www.netflix.com/watch/80047616 My Intern]
 [https://www.netflix.com/watch/330201 Breakfast at Tiffany's]
 [https://www.netflix.com/watch/70080178 Heroes]
 [https://www.netflix.com/watch/70174781 Sherlock]
 [https://www.happyon.jp/watch/100048550 Mad Max]
 [https://www.netflix.com/watch/60010351 Flashdance]
 [https://www.netflix.com/watch/80095365 La La Land]
 [https://www.netflix.com/watch/81191988 Don't stop the camera]
 [https://youtube.com/embed/AtZr9g2vwrI?autoplay=1 Fireplace]
```

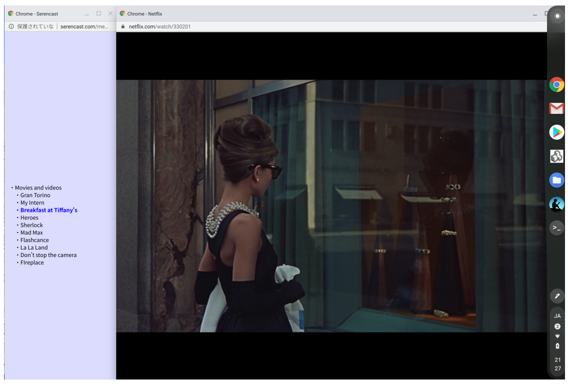

Fig. 17. Watching "Breakfast at Tiffany's".

When we access this Scrapbox page from Serencast, "Gran Trino" from Netflix automatically starts in the right side of the screen, just like a TV program starts when we rotate the channel dial of an old television. When the user pushes the ▼ key twice, "Breakfast at Tiffarny's" is selected and the movie starts at the right side of the screen. (Figure 17)

In the above example, we can select only one of the 10 movies from the movie list. But we can add more data by hierarchically providing contents in Scrapbox pages.

In one page, we can specify a hierarchical structure by using indentations. Figure 18 shows a portion of a list of music CDs and titles by "Weather Report" and "Wes Montgomery" are shown[4]. For example, “Black Market” is the first title in the album “8:30” of “Weather Report”, and “Scarlet Woman” follows “Black Market”. “Wes Montgomery” is listed in the same layer as “Weather Report”, and the album “Tequilla” is listed one level below the level of the name of the artist.

Fig. 18. Music list.

The title of this Scrapbox page is “Music”, and we can collect Movie page, Music page, and other pages by listing them in another page (“Top” page, in this case).

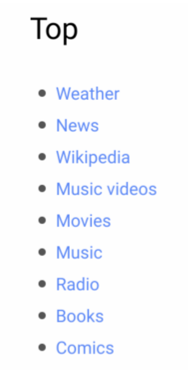

Fig. 19. Top page.

Many pages (Music, Movies, etc.) are listed in the Top page, and the pages are treated as children pages of the Top page. By constructing data hierarchically like this, We can build a very large hierarchical database for watching movies, videos, news, etc. A hierarchical database with As many as 100,000 pages can be easily constructed on a Scrapbox pages, and we can select one of the entries only by using only ▲ and ▼ keys, using the Gear technique described in section II.

[4]Weather Report is the name of a famous Jazz group, and Wes Montgomery is a famous Jazz guitarist.

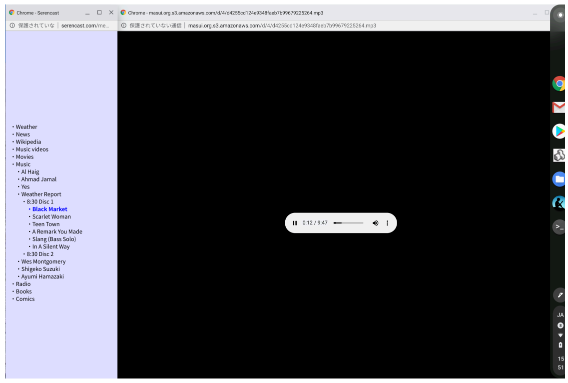

Fig. 20. Playing a music.

Fig. 21. Reading a comic book.

Since anybody can edit Scrapbox pages and construct the hierarchical data, people can enjoy creating their own hierarchical database by editing their Scrapbox pages. When someone wants to advertise their works, he can create his Scrapbox pages and give them to other people so that they can watch the pages. People can freely include other people’s Scrapbox pages in their own list and enjoy all the contents just by using Gear.

IV. Discussion

A. Input devices for Gear

Since only two switches are required in the Gear interface, almost any kind of input devices can be used for Gear. We have tried various input devices and collected experiences.

1) Keys and buttons: The simplest input device for Gear may be keyboards and push buttons. Gear works pretty comfortably with arrow keys on a PC. Various HID devices[5] are available on the market, and all such devices can be used as an input device for Gear. For example, a foot switch for turning music score pages[6] can be used as an input device for Gear. Even a special foot pedal can be used for Gear (Figure 22).

[5]https://en.wikipedia.org/wiki/Human_interface_device
[6]https://airturn.com/products/airturn-pedpro

Fig. 22. A foot switch for turning a music score.

2) 2-way lever: We have also tried to use a paddle-like device for generating ▲ and ▼. Users can push the puddle to the right to generate ▲ and left to generate ▼. Pressure sensors are installed on the puddle, and many ▲'s are generated when the user pushes the puddle strongly.

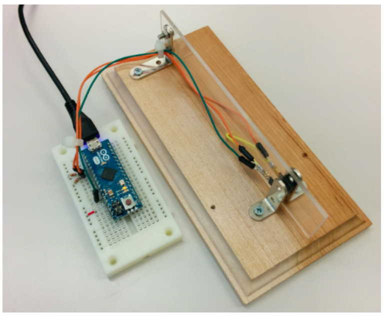

Fig. 23. Paddle device for Gear.

Controlling the speed of the scrolling of Gear by pressure is convenient, since sometimes users want to jump to an entry far from current position. That behavior is similar to the key-repeat feature of computer keyboards.

3) Rotating devices: We then tried various rotating devices for Gear. We tried mouse wheels first, and found that they are very suitable for Gear, because we are good at controlling our finger with precision.

Many kinds of rotating input devices have been used for controlling TVs and VCRs so far. For example, "jog dials" have been widely used for controlling VCRs, and we tried the device for Gear.

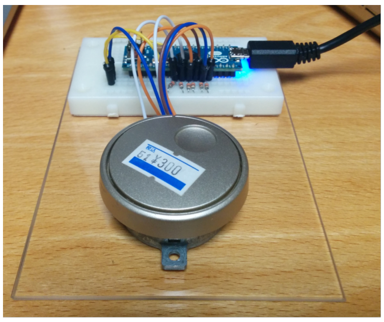

Fig. 24. Using a jog dial.

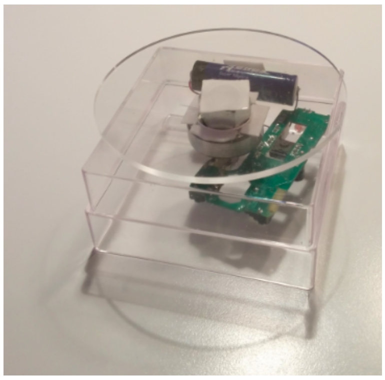

Fig. 25. A disk-based device for Gear.

We have also tried using a cylinder for Gear. The rotating cylinder shown in Figure 26 is touching the mouse wheel, and mouse wheel events are generated as the user rotates the cylinder using his arm or hand.

Fig. 26. Using a roller.

After trying these devices, we found that controlling rotating devices like jog dials and cylinders are not as easy as we expected, because precise control of an arm or a hand is fairly difficult. People are good at controlling fingers, but people are usually not very good at controlling arms and hands precisely.

After trying various devices, it became clear that simple buttons or wheels can be put at anywhere like on the table or under a chair, and we felt that the integration of furniture and input devices will be important in the IoT age. If we can install the devices neatly on furniture and beds, people with disabilities can easily use them,

B. Comparison with existing navigation methods

To compare the speed of Gear with conventional navigation method, we created two applications for checking how fast a user can select an entry from a large hierarchical database. Both applications behaves the same way: a user can set the date to a specific time to see a photo in a 15-minute monorail trip. Using the first one (App1), a user can set the time using Gear (Figure 27) with up/down arrow keys for ▲ and ▼.

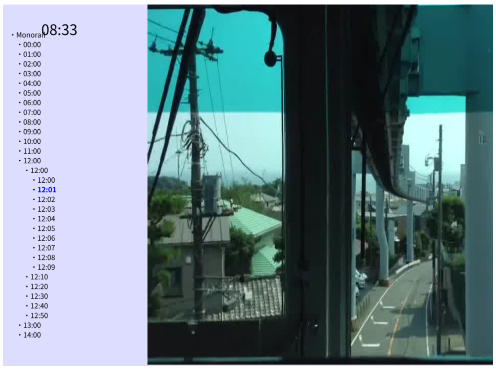

Fig. 27. App1: Gear-based application for finding a monorail picture.

Using the second one, a user can set the time using the `<input type="time">` tag of HTML (Figure 28). Users can first set the value of the "minute" value at the top-left using up/down cursor keys, and move to the right with the right-arrow key and set the "second" value with up/down arrow keys.

Fig. 28. App2: using input element.

On both applications, a random target time is displayed on the screen and the subjects are asked to set the time to the displayed target time. After repeating the task 5 times, the average time is shown to the user, and the value is reported.

We asked 5 people to try these tasks 10 times, and the result of the experiment is shown in Figure 29.

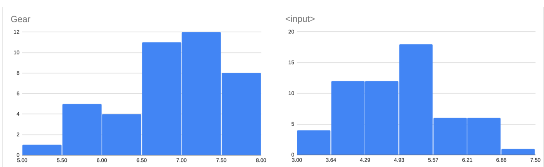

Fig. 29. Average search time of App1 and App2.

The average time for finding an entry using Gear was 6.84s, and the average time using standard `<input>` was 4.96s. It takes about 40% more time than using conventional method. The standard deviations were 0.71s and 0.93s, respectively. This is a little bit disappointing, but we are happy to see that no-click browsing with Gear is not too slower than conventional methods.

We have also tried using Finder.app for the same data (Figure 3), and got similar results as App2.

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 Gear's automatic transition (e.g. from Figure 8 to Figure 9) looks familiar to users.

C. *Comparison with InfoVis techniques*

Various information visualization techniques like Treemap [1], Hyperbolic Tree [2], and Sunburst [3] have been proposed for visualizing large hierarchical data. Zooming user interface (ZUI) systems like Pad [4] and Pad++ [5] 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. However, users of these systems have to use a pointing device to take full advantage of these methods, and no-click browsing is not supported.

It seems to be a good idea to use Gear on more conventional visualization techniques like TreeView[7], by replacing 4-way navigation with Gear.

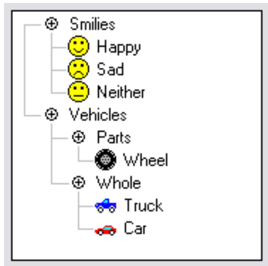

Fig. 30. A TreeView example.

D. *Timing control*

The biggest disadvantage of Gear is that the behavior of Gear depends on the speed of the user's operations. In Figure 8, if a user wanted to select Clothing but couldn't issue ▼ quickly enough, he will see Figure 9 instead of Figure 12. This is a tradeoff between usability and simplicity, and the best parameter should be set based on the user and the Gear device.

E. *Is the Gear interface new?*

Since the idea of Gear interface is so simple, we have been wondering if the same idea existed in the past. We could not find a research paper on navigation techniques based on two keys. Many key-based navigation techniques are found on patent database, but we could not find a navigation method

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

based on two keys. We are almost sure that The Gear interface had never been popular before.

F. Using Gear in the wild

One of the authors have been using Gear in his living room for several years with a small personal computer connected to a 60-inch TV monitor. The database is managed on Scrapbox, and a mouse wheel of a wireless Bluetooth mouse is used for Gear interface. With Gear, the user can enjoy all available contents on the web with the mouse wheel without selecting the source of the movies, animes, and musics from Amazon, Netflix, etc.

We can enjoy using Gear everywhere using a stick PC and a wireless mouse, just like Amazon FireTV can be used for the same purpose. Using Serencast is more fun for the author, since no-click browsing is more comfortable than selection-based interface on existing systems.

We demonstrated Gear at an exhibition and asked more than 100 people to try it (Figure 31). Since the only thing a user can do with Gear was to rotate a wheel, users seemed to be able to understand the behavior of Gear with trials and errors. Using only one rotating device for data navigation is a new experience for all the visitors, but the hard restriction of the device seemed to have worked in this case.

Fig. 31. People at the right using a wheel device at an exhibition.

V. Conclusion

We have developed a new simple interaction method "Gear" for exploring large hierarchical data structure. A Gear 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 Gear and try them at various places like kitchens, restrooms, etc.

References

[1] B. Johnson and B. Shneiderman, "Tree-maps: A space-filling approach to the visualization of hierarchical information structures," in *Proceedings of the 2nd Conference on Visualization '91*, ser. VIS '91. IEEE Computer Society Press, 1991, pp. 284–291. [Online]. Available: http://dl.acm.org/citation.cfm?id=949607.949654

[2] J. Lamping, R. Rao, and P. 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*, ser. CHI '95. ACM Press/Addison-Wesley Publishing Co., 1995, pp. 401–408. [Online]. Available: http://dx.doi.org/10.1145/223904.223956

[3] J. Stasko and E. Zhang, "Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations," in *Proceedings of the IEEE Symposium on Information Vizualization 2000*, ser. INFOVIS '00. IEEE Computer Society, 2000, pp. 57–65. [Online]. Available: http://dl.acm.org/citation.cfm?id=857190.857683

[4] K. Perlin and D. Fox, "Pad: An alternative approach to the computer interface," in *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '93. ACM, 1993, pp. 57–64. [Online]. Available: http://doi.acm.org/10.1145/166117.166125

[5] B. B. Bederson and J. 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*, ser. UIST '94. ACM, 1994, pp. 17–26. [Online]. Available: http://doi.acm.org/10.1145/192426.192435