ELECTRONIC WORKSHOPS IN COMPUTING

Series edited by Professor C.J. van Rijsbergen

C.R. Roast and J.I. Siddiqi, Sheffield Hallam University, UK (Eds)

BCS-FACS Workshop on Formal Aspects of the Human Computer Interface

Proceedings of the BCS-FACS Workshop on Formal Aspects of the Human Computer Interface, Sheffield Hallam University, 10-12 September 1996

An Analysis of User Behaviour in an Interactive Auditory Field

T. Ebina and S. Igi

Published in Collaboration with the
British Computer Society

An Analysis of User Behavior in an Interactive Auditory Field

Tsuyoshi Ebina and Seiji Igi

Communications Research Laboratory, Ministry of Posts and Telecommunications
Koganei-shi, Tokyo, Japan

Abstract

Auditory user interfaces provide a possibility for visually disabled people to interact with computers. These interfaces are formalized by four characteristics: interaction, relationship, state change awareness and cocktail party effect. We propose such an interface termed as an "Interactive Auditory Field" (IAF) which satisfies these four characteristics and develop a prototype system that allows different types of auditory icons to be selected by the user. An experimental evaluation of the system revealed that users could not distinguish between different frequencies and therefore a sound synthesising method is required. Finally our informal approach to an interesting interface is offered to the workshop as a contribution for formalization.

1 Introduction

The development of Graphical User Interface (abbreviated to GUI) has excluded visually disabled people from accessing information using computer systems. An ordinary GUI system has some graphical objects, called icon. An icon is a metaphorical desk top figure, similar to actual desk tools, such as calculators, mailboxes, note pads, and so on.

Though icons are very useful for computer users who are not visually disabled, they are almost useless for visually disabled people. So auditory user interface design for visually disabled people is required.

2 Auditory User Interface Design

In this section we reviews some of auditory user interfaces and formalize the design of an auditory user interface.

2.1 Auditory User Interface Design Principle

The following characteristics must be taken into account in the design of auditory user interfaces:

Interaction Since blind people must find objects using auditory key, interactiveness must be considered to design auditory user interface. In this context, interaction means that user's action should be reflected by sound. For example, when the mouse pointer is close to a icon, the volume of the sound becomes high.

Relationship The relationship between the mouse cursor location and the icon location must be explained to the user. For example, when a icon is located on the monitor to the left of the user, user can hear a sound from the left side.

State change awareness When the state of a system have changed, the auditory user interface must make the user aware. For example, when the mouse cursor enters an icon, the user can hear some change of sound.

Cocktail party effect When a user can hear only one sound at a time, the user needs long time to search a desired sound. However, if a user can hear some kind of sounds simultaneously, he/she will be able to search a desired objects in a short time.

Figure 1: Concept Image of IAF

2.2 Related Work

Many auditory icon interfaces have been developed. For example, Brewster et. al. have developed Earcon[1]. Earcon is used to distinguish between icon metaphors based on instrumental sound. It explains the family attribute of the item as timbre, the type of attribute of the item as rhythm. Items in the same family and type are differentiated by pitch. For example, the paint application, the paint folder and paint files all had the same instrument. Though Earcon is effective, its auditory metaphors have not the attributes of interaction, relationship, and cocktail party effect.

Arthur I. Karsmer has designed a method for visually impaired people to navigate their way around a computer using appropriate tonal clues[2]. In his approach, some objects on the desk top (windows, folder icons, data file icons, etc) are presented by sounds (beeps, trumpet, flute, etc) and voice. His approach is similar to that of Earcon. However, his system does not possess the attributes of interaction, relationship and cocktail party effect.

Ludwig et. al. extended the notion of a Window System to Audio[3]. They proposed a one-, two- or three- dimensional source distribution using Convolution and Digital Signal Processors. Ludwig's system satisfies interaction, relationship, and cocktail party effect. However, their system requires many digital signal processors, and it is very expensive and time-consuming to develop a prototype system.

3 Interactive Auditory Field

We propose a mouse handling method using an "Interactive Auditory Field" (abbreviated to IAF). IAF is a virtual auditory field that explains visual icons as auditory icon metaphors. In IAF, user can walk around and find an appropriate auditory icon using auditory feedback, this is illustrated in figure 1. It is composed of sound metaphors and voice. Using cocktail party effect, the user can hear many sounds simultaneously. For example, in figure 1, a calculator icon is located on the monitor to the left of the user. The calculator auditory metaphor is presented as a "clickety-clack" (the sound of a calculator being pressed). The user can hear a "clickety-clack" sound from the left side. If a user moves the mouse cursor towards the direction of the sound, the volume increases, and the other auditory icons will change their location and volume. If a mouse cursor enters an icon, a sound will arise to announce that the user is in the icon.

3.1 Prototype System Design

We are developing IAF on an X Window System. The X window application communicates with the X Window System through events. An event is low-level data composed of window-ID, location of mouse cursor, and so on. The event can be categorized into different types of event: ButtonPress event, PointerMotion event, EnterNotify event, and so on. When a user moves a mouse cursor, a MouseMotionEvent is sent from X Window System to the X application. We extended Tab Window Manager(abbreviated to TWM) which manages all icon windows on the display. Each time a user moves a mouse, the TWM calculates the distance and direction from the mouse cursor location to each icon window. The method is as follows:

1. While the user moves a mouse, all sound parameters are recalculated and the following procedures will be executed.
2. The distance between the mouse cursor and an icon window is revealed according to volume. If an icon window is located near the mouse cursor, the volume of that sound is high. If an icon window is located far from the mouse cursor, the volume of that sound is low.

The angle from the mouse cursor to an icon window is explained by the direction of sound source. When an object is located on the left side of the mouse cursor, the user hears the object's sound from the left side speaker of the headphone. Likewise, when an object is located on the mouse cursor, the user hears the object's sound from the right side speaker of the headphone. If the mouse cursor is on an icon, the user can hear the object's sound from both speakers of the headphone. If the user moves the mouse cursor from the left side to the right side, the sound source also changes from the left side to the right side. These generated sounds are mixed together, and presented to the user.

3. If the mouse cursor is located on the left side of the root window, the user can hear telephone tone signal 4 from the left speaker. Accordingly, if a mouse cursor is located on the right side of the root window, the user can hear telephone tone signal 6 from the right speaker. If the mouse cursor is located on the upper end of the root window, the user can hear telephone tone signal 8 from both speakers. Similarly, if the mouse cursor is located on the lower end of the root window, the user can hear an telephone tone signal 6 from both speakers. Using these alarm sounds, the user will understand that the mouse cursor is on the top, bottom, left or right.

Though our prototype system design is very simple, it satisfies interaction, relation, state change awareness and cocktail party effect. However, these types of auditory icon interfaces have the following problems: how should the same type of icon instances be explained? and can people distinguish them?

To solve the problem, we adopted the following techniques. If the icon is explained using a sound, all icons belonging to the same type will have a different frequency. Though binaural techniques may be required, these techniques require long calculation time and many hardware resources, like Ludwig's window system[3]. To analyze user behavior, however we made a simple prototype system. Our prototype does not require any additional component to a normal workstation other than headphones.

4 Experiments

4.1 Experimental Hypothesis

Because our proposed system satisfy interaction, relation, state change awareness and cocktail party effect, visually disabled people seems to be able to select icons using auditory cues. If they can select icons effectively, their mouse motion distance should be short. Therefore two types of experiments, auditory cue condition experiment and without auditory cue condition experiment, are examined to confirm the effect of cocktail party effect.

The auditory cue condition experiment satisfies interaction, relation, state change awareness and cocktail party effect. But the nonauditory condition satisfies only state change awareness. Under nonauditory feedback conditions, mouse movement does not cause interaction. Under nonauditory conditions, sounds are generated when the mouse pointer enters in an icon or the mouse cursor hit the root window frame.

subject	average motion distance	min	max	standard deviation
A	14735(22666)	9056	24359(205060)	5454(38266)
B	14815	6281	29798	9063
C	13613	7997	28314	4916
D	17825	6891	33661	8648

Table 1: Auditory Feedback experiment

4.2 Experimental Condition

The experimental environment was as follows. Display size is 1152 pixels by 900 pixels. Before the experiment, the mouse speed was set at the slowest rate(xset m 1 1). The size of each icons was 100 pixels by 100 pixels. These icons were located at the left, right, top, or bottom of the display. Four persons were used as subject and they were not blind. The relationship between mouse movement and sound reaction was instructed to the subjects before these experiments.

In the first experiment, they were asked to select icon using auditory cue. They were not allowed to watch both a display and a mouse. Each subject repeated the following task 24 times. At the beginning of the task, a beginning auditory cue (train sound), which introduces the task start, was shown to the subjects. After the beginning auditory cue, subjects were allowed to select icon. Four auditory icons are presented to the subjects. These auditory icons were:

1. fast speed Cockadoodledo(Cockadoodledo1)
2. normal speed Cockadoodledo(Cockadoodledo2)
3. speech voice
4. flush sound

These icons were located in different position by tasks. All sound data was obtained from the /usr/demo /SOUND/sounds directory on a SunOS 4.X except for sound Cockadoodledo1. Sound Cockadoodledo1 was generated from Sound Cockadoodledo2 using thinning out sampling data at a rate 4/5. Subjects were required to select from auditory icon 1 to auditory icon 4 in this order. Subjects were also instructed that when they selected a wrong icon, they should select the high order auditory icon next. For example, if the subject selected auditory icon 2 first, he/she should select auditory icon 1 next. When the subject clicked the mouse button on the icon, a beep was generated and he/she knew that the icon had been selected. However, if a subject failed to select the icon, no sound was generated. If a subject selected all icons, the subject heard the end of the auditory cue (crash sound), and the task was finished.

After the first experiment, they took a rest of 10 minutes. Then, they were asked to perform the same task under nonauditory feedback conditions. Each subject also repeated this task 24 times.

4.3 Experimental Results

Tables 1 and 2 show the results of total mouse movement distance per task. Table 1 shows the experimental results under auditory feedback conditions, and Table 2 shows the experimental results without auditory conditions.

In Table 1 and Table 2, average motion distance indicates the average mouse cursor locus length per task. The min indicates minimum mouse cursor locus length and the max indicates maximum mouse cursor locus length. In the first task of the auditory feedback experiment, subject A moved the mouse too fast. Therefore A could not use the auditory cue, and the total motion distance recorded was 205060. Thus, we have excluded data from the first experiment for subject A in Table 1. In Table 1, the average motion distance of subject A (14735) indicates an average of 23 tests, while 22666 indicates an average of 24 tests. We could not find any significant difference in the experimental results between auditory and nonauditory feedback conditions. However, the difference between average movement distance in Table 1 and Table 2 suggests that all subjects were able to use auditory feedback effectively. Though we did not

subject	average motion distance	min	max	standard deviation
A	51060	13364	106825	38143
B	79515	31057	193890	52249
C	54151	19652	142978	30383
D	37731	10226	118555	30695

Table 2: Nonauditory Feedback experiment

Figure 2: Locus of a mouse

subject	auditory feedback condition		nonauditory feedback condition	
	correct	incorrect	correct	incorrect
A	17	6(1)	20	3(1)
B	14	10(0)	12	9(3)
C	15	9(0)	15	9(0)
D	16	8(0)	17	6(1)

Table 3: Number of correctly and incorrectly selected icon

conduct the experiment under visual conditions, the motion distance was estimated to be less than 4000. Therefore, we can say that users were able to use auditory cues effectively, but auditory feedback conditions are less effective than the visually sighted condition.

5 Discussion

Based on the above results, we will discuss IAF’s benefits and disadvantages in this section. Though auditory cues were seemingly used by all subjects, the average movement distance in Table 1 still exceeds 10000. This result suggest that our prototype needs to be improved.

The main problem with the current prototype system is that it does not explain the vertical direction relationship. Because the vertical direction sound source is not explained in our prototype system, subjects cannot directly create an image of a two-dimensional sound field. Consequently, when a user wants to move the mouse cursor to an icon, he first moves the mouse to the left or to the right. If the direction of the sound source has changed, he next moves the mouse forward or backward.

Figure 2 shows an example locus of a mouse cursor. The unfilled circled figure shows that the button was pressed, but it failed to select the icon. The filled circle figure shows that the button was pressed and the icon was selected. Though this method is less realistic than a binaural model, it requires fewer hardware options. To select an icon with a shorter path, we need to append vertical sound source expression techniques to the current prototype system. As one solution, we are considering a way to represent the vertical direction relationship using frequency, such as a Doppler effect metaphor. However, when the tone of auditory icons is changed according to mouse cursor location, the problem of explaining the difference between application instances arises. For example, a user often uses more than two editors simultaneously. If the user can see, he can select the correct editor icon using visual information. However, will a blind user be able to select the correct editor icon under the Doppler effect? To solve this problem, we investigated the above result in more detail. Table 3 shows the number of correctly and incorrectly selected icons. In Table 3, the number of incorrectly selected icons is based on a misunderstanding of Cockadoodledo1 (fast speed) and Cockadoodledo2 (normal speed). The numbers in parenthesis show the other misunderstood data. This result suggests that under both conditions, users often mistake objects which have similar auditory characteristics. This reveals that a difference in frequency is not necessarily effective in representing auditory icon characteristics. Therefore, although we have not yet examined a Doppler effect metaphor, we can say that it may not be of much benefit. A Doppler metaphor would not necessarily explain the location of vertical relationships, especially when the same types of icons are on the screen.

As we are using a non instrumental auditory icon metaphor, we cannot use the Earcon’s approach [1], which differentiated icon instances by pitch. Therefore, other sound processing techniques are required. One idea that may be effective is Gaver’s sound synthesizing technique [4]. In Gaver’s approach, bouncing and breaking sounds are characterized by the temporal patterning of a series of impacts. For example, the difference in “flip flip” sounds (a sound depicting book pages being turned over continuously) of a note pad and a dictionary have different characteristics. These characteristics are not explained by differences in frequency.

Another problem regarding icon selection is that a subject’s mouse cursor often passes through the icon region. Fitt’s law is known to describe the relationship between mouse positioning time and the distance from the mouse cursor to

type	X	Y	time	label
E	98	456	22.060	Cockadoodledo1
E	1	450	24.283	Cockadoodledo1
S	14	453	26.455	Cockadoodledo1
E	1070	408	60.182	Cockadoodledo2
S	1069	419	61.780	Cockadoodledo2
E	559	21	75.576	SpeechVoice
E	530	31	79.131	SpeechVoice
F	570	155	79.146	SpeechVoice
E	572	119	80.087	SpeechVoice
F	652	90	80.857	SpeechVoice
E	625	106	85.208	SpeechVoice
S	607	110	86.759	SpeechVoice
E	623	889	94.490	FlushSound
E	623	885	95.625	FlushSound
F	598	896	97.989	FlushSound
E	560	894	102.754	FlushSound
E	528	890	105.018	FlushSound
S	563	886	105.708	FlushSound

Figure 3: Sample event history of subject A

the icon. Fitt's law can be written as follows:

$$T = I \log_2(D/S + 0.5)$$

Where, T is positioning time, D is distance from the mouse cursor to the icon, S is size of the icon and I is constant.

According to Fitt's law, as the mouse cursor gets close to an object, the speed of mouse movement slows down. When the mouse cursor is on the icon, the mouse stops. The event history for each subject was analyzed. Figure 3 shows the sample event history for subject A. In Figure 3, Type explains Enter(E), Success in selecting (S) or Fail in selecting(F). The X and Y represent the coordinate where the event is generated. Time means the time when the event is generated. Label is the icon label. From Figure 3 we can see that the subject often passed through the icon. This reveals that this result does not conform to Fitt's law. This is because the location of the mouse cursor cannot be precisely explained, though our system satisfies state change awareness. For example, can a sound effect explain that the mouse cursor is located on the lower right side of the icon now, and that it has almost left the icon? We do not have the answer to this question now. It must be considered in future research.

6 Conclusion

We designed an auditory user interface based on four characteristics, interaction, relation, state change awareness, and the cocktail party effect. Some auditory user interfaces were compared using these characteristics, and we made a simple auditory user interface prototype system and conducted experiments. The experimental results suggests that users can select icons using the cocktail party effect, but greater improvement of our prototype is needed.

We have adopted a traditional and informal approach to the construction of an interesting and unusual interface. Our formalization suggests the following results:

1. Interaction reduced average motion distance.
2. Relationship helped users to show direction of icons. However, it was difficult to explain the vertical direction relationship using frequency.

3. State change awareness was a important cue to click a mouse on a icon. It reduced the number of mouse click outside the icon.
4. Cocktail party effect was effective in selecting an appropriate auditory icon among auditory icons. However, we found that users have difficulty distinguishing between two icons of the same type, even though these sounds have different frequencies. This suggests that other sound process techniques that reconcile class information and characteristic information will have to be developed.

References

- [1] Brewster S. A., Wright P. C., and Edwards A. D. N. An Evaluation of Earcons for Use in Auditory Human-Computer Interfaces. In INTERCHI'93 Conference Proceedings,pages 222-227. Addison Wesley,1993.
- [2] Karshmer A. I. Navigating the Graphical User Interface (GUI) by the Visually Impaired Computer User. In Y.Anzai et al. editor, Proceedings of the Sixth International Conference on Human-Computer Interaction - HCI95, pages 149-154. Elsever,1995.
- [3] Ludwig L. F., Pincever N., and Cohen M. Extending the Notion of a Window System to Audio. In IEEE Computer, August, pages 66-72. 1990.
- [4] Gaver W. W. Synthesizing Auditory Icons. In INTERCHI'93 Conference Proceedings,pages 228-235. Addison Wesley,1993.