Angle Recognition Cues using a new API dedicated to the VTPlayer Mouse

Benoît Martin, Isabelle Pecci, Thomas Pietrzak, *LITA, Université Paul Verlaine-Metz*

***Abstract*—The VTPlayer mouse is a tactile device that provides two matrices of pins that can be raised or lowered. We aim to use this device to make application more accessible for visually impaired children. To simplify the use of such device in classical applications, we propose a new API (Application Programming Interface) that manages the concept of icon independently of the feedback of the icon. The programmer can easily define icons and propose multimodal feedback (visual, tactile …). In the second part of this paper, we test this API through experiment to recognize angle with tactile icons. The choice of a set of icons suitable for angle discrimination is studied and a possible good one is proposed. User tests are conducted with this set of icons and a statistical study demonstrates its efficiency.**

***Index Terms*—Haptic, tactile display, icons, visually impaired users**

I. Introduction

THE Micole european project [1] aims to create a multimodal and collaborative learning environment for visually impaired children. The modalities used are auditory cues (sound and speech), and haptic cues (tactile and force feedback). Our team only works on haptic cues. Haptic is the sense of touch that is divided in two main categories: tactile feedback (temperature, roughness, pain ...) and kinaesthetic feedback (or force feedback: pressure, inertia, shape ...). Our goal is to use haptic devices to give the user some information like guidance for example. We present in this paper an API (Application Programming Interface) developped for the VTPlayer mouse and a study about angular cues recognition with this device.

The VTPlayer [2] is a tactile mouse with two 4 × 4 pin matrixes (figure 1). Each pin can be computer-controlled so that we can display all the patterns we want.

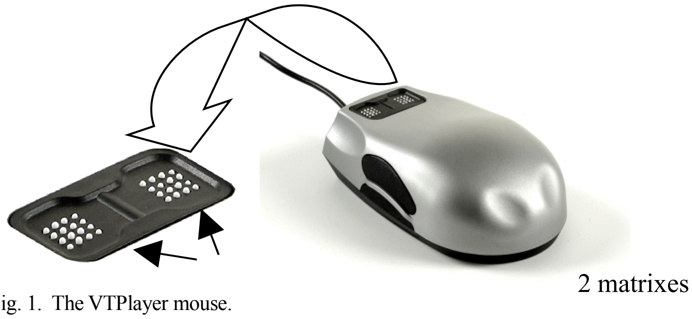


Fig. 1. The VTPlayer mouse.

There are two ways to give information on tactile devices: the first one is to translate visual information into tactile information, and the other one is to create icon-like information. Jansson has tested a geographical map exploration with a VTPlayer mouse [3]. The software used is one of those given with the VTPlayer. The user can feel the borders and some texture while moving onto the map. The results show that in this study the tactile output was not more helpful than audio output.

Wall and Brewster did a study about graph recognition [4]. Some users had to explore some lines with a VTPlayer, a force feedback mouse and raised paper. The goal was to compare the three techniques and to establish the user's efficiency to recognize whether a line goes up or down. The results show that the VTPlayer recognition threshold is ±4.7°, against ±3.25° for the force feedback mouse and ±2.42° for the raised paper. The authors explain the superiority of the raised paper by the combination of tactile and kinaesthetic sense that does not exist in the two other models. Another explanation is that with raised paper the user has the possibility to explore the scene with two hands. This study and the previous one use visual to tactile translation, which is claimed to be not enough for good recognition [5]. Moreover these tests only treat about angle recognition relatively to the horizontal line. Our goal is to recognize absolute angles.

In a previous work, we have designed directional icons [6]. These icons use a 4×4 tactile matrix display and represent eight directions. Several sets of icons have been tested, some of them are static and other ones are dynamic (the pattern changes to feel an animation). The results have shown that users rather prefer static icons because they can palpate the cell to recognize them. These icons have been used by Crossan and Brewster [7] to guide a user through a virtual maze. Their study confirmed that static icons are better recognized, and are sufficient to guide a user.

These sets of icon can give sufficient directional information with their eight directions. But in some case we want to indicate

Manuscript received July, 2006. This work was supported by the European Community under the project IST-2003-511592.

B. Martin is with Laboratoire d'Informatique Théorique et Appliquée –LITA of the Université Paul Verlaine-Metz, FRANCE (phone:+33 3 87 54 73 17; fax: +33 3 87 31 53 09 ; e-mail: benot.martin@univ-metz.fr).

I. Pecci is with Laboratoire d'Informatique Théorique et Appliquée –LITA of the Université Paul Verlaine-Metz, FRANCE (phone:+33 3 87 31 54 42; fax: +33 3 87 31 53 09 ; e-mail: isabelle.pecci@univ-metz.fr).

T. Pietrzak is with Laboratoire d'Informatique Théorique et Appliquée –LITA of the Université Paul Verlaine-Metz, FRANCE (phone:+33 3 87 31 54 43; fax: +33 3 87 31 53 09 ; e-mail: thomas.pietrzak@univ-metz.fr).

the user a precise angle, then eight icons are not sufficient: it makes only a precision of 45°. So we have designed angular icons.

II.THE VTPLAYER API

The VTPlayer's official driver can be programmed through a dll, but only on Windows systems. On the opposite, a GPL (General Public License) Linux driver has been developed by Jacquard [8] and it only works on linux systems. We thus decided to develop a cross platform API, which is based on C++ and libusb. The API is based on MVC architecture (Model View Controller).

A.Model

The model is the basis of the API. It corresponds to what we want to display, that is the icons. The model must include static icons and dynamic ones. A static icon corresponds to a constant display of the matrix of pins during the time. A dynamic icon presents a matrix that varies along time. We call "step" each new view of the matrix. As icons can be complex by what they mean and by their use, four main classes are proposed:

1) VTPMatrix class. It's the semantic specification of an icon. The structure is quite simple because it only includes an array of characters representing the value of the matrix.
2) VTPIcon class. It's an abstract class that contains all annex data about icon that we consider as metadata (author's name …).
3) VTPStaticIcon class. This class aims to specify static icons. It inherits from VTPIcon. It contains the matrix (VTPMatrix).
4) VTPDynamicIcon class. This class represents the dynamic icons. It inherits from VTPIcon and contains a list of VTPStaticIcon. Each static icon corresponds to one step of the dynamic icon. The VTPDynamicIcon also includes all parameters required to define the dynamicity: time of the animation, delay of display for each step and iteration of the animation (to repeat it or not).

With this set of class, complex icon can be designed. As example, half dynamic icons can easily be defined with VTPDynamicIcon class. The list of VTPStaticIcon must contain successive icon where only a part change. Then the animation will produce icon where only a part blink.

A.View

The view is all the kind of output we want, that is for example the VTPlayer matrix display, or a visual window for debugging without device purpose. A view is the presentation for the user and defines the main actions allowed with this view: it includes the two matrixes of pins and it manages useful events that occur on the view. This first version of the API includes two specific views:

1) VTPView class. It's an abstract class that defines all common methods for any view.
2) VTPMouse class. It manages the VTPlayer mouse device. The data stored are similar to the data sent by USB port. The view manages the initialisation of libusb and the events of the mouse.
3) VTPScreen class. It aims to represent the screen. The user could then see a display of icons in a window by using the library SDL. This view can be useful for different purposes. The developer does not need a VT Player mouse to test icons; he can also have an easier way to debug interaction with icon. This view is convenient for user too during a learning session or for collaborative works between sighted and visually impaired people. The figure 2 gives an example of the screen view.

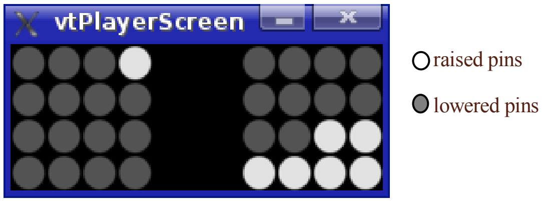


Fig. 2 Example of rendering for the VTPScreen view

B.Controller

The controller has to translate the model data to a data usable by the view. It is an interface between the model and a view that converts data contained in derived VTPIcon class into data matching derived VTPView class. To manage dynamic icons or events that occurred on icons, the controller may communicate with the model and the views at regular time intervals. VPTController class is an abstract class used to define the global behaviour of the controller. Many other controllers are implemented according the views used. In our first version of the API, we developed two controllers linked to the two specific views (CTPMouse and VTPScreen): the VTPMouseController class and the VTPScreenController class. They inherit from VTPController. They do not be so quite different from VTPController but they contain specific methods for the view they are attached. For example, as the VTPScreen view do not send events, its VTPScreeController controller does not need to manage threads of events.

III.Angle recognition experiment

A.Presentation

Based on our new API, we implement an application to test angle recognition with tactile feedback. The VTPlayer mouse is used to send tactile angle to user. Each angle will be associated to an icon displayed on one matrix of the VTPlayer. The set of angles to recognize are located in the first quadrant from 0° to 90° and the step is fixed to 15°. We restrict the study to this quadrant because we assume that symmetry could be applied to the set of tactile icons to obtain all the other angles (from 90° to 360°).

In the experiment, we want to test if users can recognize a random set of angles with tactile icons and eventually find the worst ones. We analyse:

1) if there is a session effect i.e. if the rate of success increases along the sessions;
2) if some icons are more difficult to recognize;
3) if the rate of failure for each icon is near zero.

B. Set of icons

The strategy used to find a good set of icons is based on a constraint: we want to define only icons for the first quadrant and deduce the other quadrants by applying symmetry on icons of the first one. As the user will learn icons by quadrant, each set of icons has to help him know which quadrant is used. Sjöström [13] recommends putting some reference points so that the user could anytime refer his position to it. Each icon includes reference points to help user locate the quadrant.

Many sets are candidate but two strategies appear: supplying icons that maintain strong semantic links with angles they represent or supplying only discrete icons. The first set proposed tries to keep semantic link with angle definition. The figure 3 shows examples of tactile angles. X axis and Y axis are included in each icon as reference points and angles make lines toward X-axis or Y-axis. This set is not appropriate because if we apply symmetries to obtain the other quadrant, ambiguities appear with angles 45°, 135°, 295° and 315°.

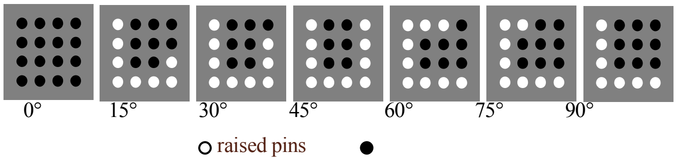


Fig. 3. Tactile angles: first set of icons

We assume that reference points and lines to feature angles in this first set are too big. A second set is proposed based on the same semantic approach. The angles are defined with only one point and one reference point is included: the intersection of X axis and Y axis. The figure 4 illustrates this second set of icons. If we apply symmetry on this new set, similar problem of ambiguity occurs: as example, between angle 45° and 225°.

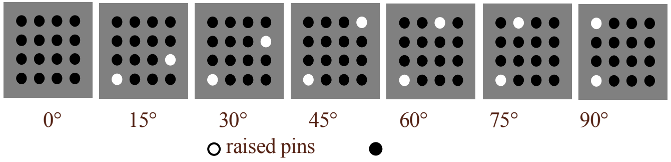


Fig. 4. Tactile angles: second set of icons

The main difficulties are to distinguish the reference point between each angle and to distinguish close angles. In the two cases, they are too similar. This observation leads us to delete semantic link with angle. The third set of icons aims to supply distinct icons. The same reference point as in the second set is maintained to help user know the quadrant and each angle is represented by a set of points different from close angles. The figure 5 shows such a set of icons. With this set of icons, the three other quadrants could be obtained by symmetry without ambiguities.

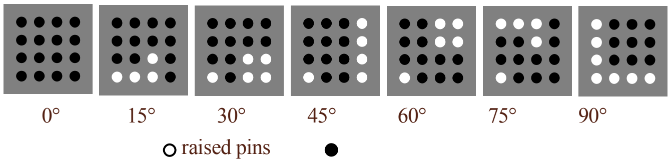


Fig. 5. Tactile angles: third set of icons

In our experiment, we choose this third set of icons. We assume that the loss of semantic link will be compensated by a better recognition of icons and by "effect session". Our experiment tends to prove it.

C. Experiment Protocol

The user begins with a training session, in which he can explore the icons as much as he wants. He just has to select an angle and the corresponding icon is displayed with the two views supplied by the API: on the screen and on the VTPlayer mouse (see figure 6). Then when he is ready he can start a 5 minutes session where a random block of icons are presented to be recognized. In a session, there is never successive icon similar to improve random distribution.

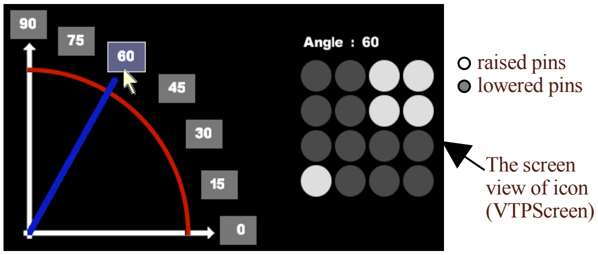
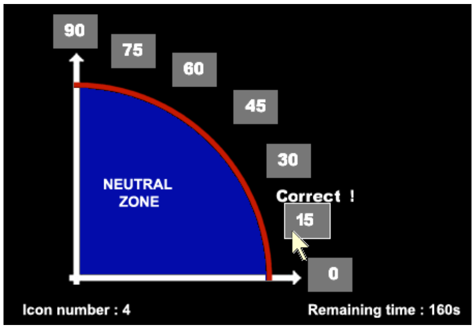


Fig. 6. Screenshot of the learning session

During the test, a quadrant is drawn on the screen, with one button for each angle available (see figure 7). The view on the screen provided during the learning session disappears for the test. The user has to click on the button corresponding to the angle represented by the tactile icon he feels on the VTPlayer mouse. The application directly shows the correct answer: the learning continues during the test session.



Icon number : 4

Remaining time : 160s

Fig. 7. Screenshot of a test session

Then the user has to go back into a neutral zone in order to get the next icon. This move is important for logs to have the same starting point for each icon and then to allow us to analyze the answer times.

A log file is managed when the user interacts. It contains data about the session: the date of the test, the user's name, the time planned and the real time for the session, the time spent to explore each icon during the learning session, the list of icons presented to user with the answer time, the angle to recognize, the answer of the user. At the end of the session, statistics are presented to user and stored in the log file:

1) the global rate of success;
2) the number of icons to recognize for each angle;
3) the number of good answer for each angle;
4) the number of wrong answer for each angle
5) the average delay of answer for each type of angle.

After each session, the user can see what he must improve for the next one. He can do new training sessions before the next test session.

D. Users

Users are sighted people but the hand they use to interact with the VTPlayer is hidden: they can't see tactile icons. Users can be right-handed or left-handed. To avoid difference between the grips and allow everyone to use its index, tactile icon is displayed on the left matrix for right-handed users and on the right matrix for the left-handed users.

Twelve users participate to the test and they have to do five sessions. Only eight did the five sessions, one did four sessions, two did two sessions and one did one session.

E. Results and discussion

At the end of the test experiment, all the log files are analysed to generate two files that are used for statistics: one stores the rate of success for each session and each user and the other stores the average time of answer and the rate of failure of each user for each angle during each session.

We use the variance analysis called Anova1 to study the influence of one qualitative parameter, a session, on a quantitative parameter, the rate of success. This analysis aims to determine if for every user there is a "session effect" on the rate of success i.e. a progress in recognition along the sessions.

The figure 8 shows the result of the Anova1 with a box and whisker diagram. This box plot reveals a "session effect" for the three first sessions. For the two last ones, this effect disappears. A test of Student on these sessions shows that their medians are similar.

Another variance analysis allows to study the average time to answer and the rate of failure functions of the icon, the session and the user. To illustrate these results, we show the boxes plots of figure 9 and 10. By looking at figure 9, icons for angle 0°, 45° and 90° are examined quickly while users take more time to give an answer for icons for angle 15°, 30°, 60° and 75°. Icons used for 0°, 45° and 90° are easier to feel because for 0° no pins are raised and for 45° and 90° icons is composed of raised lines. All the others icons take more time because pins raised design a drawing more complex.

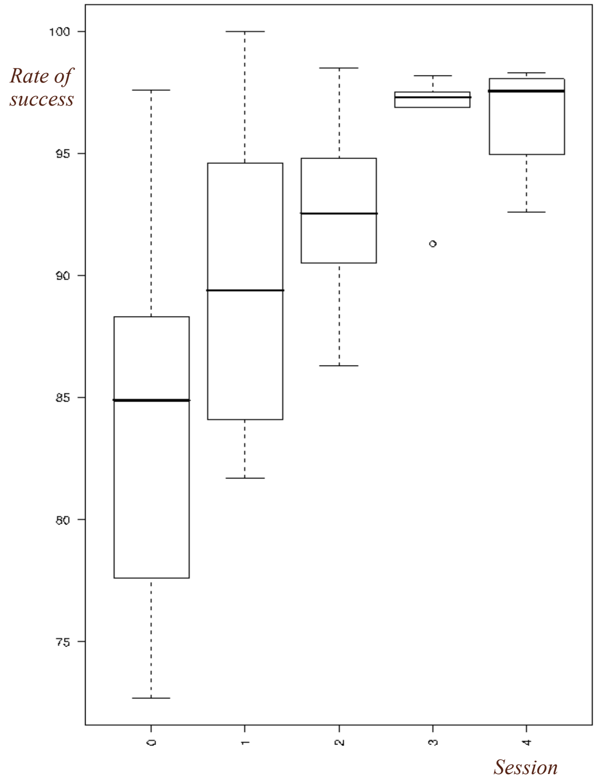


Fig. 8. Box plot: the "session effect".

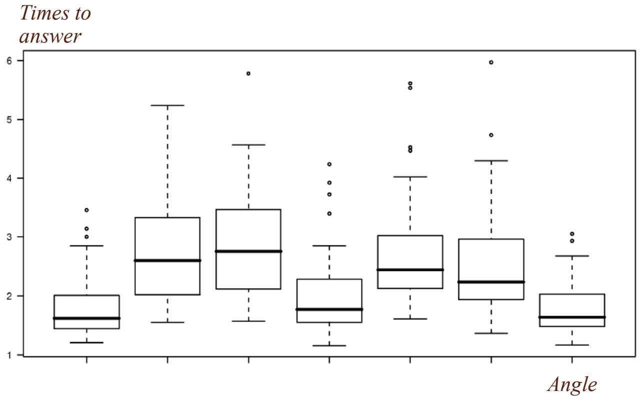


Fig. 9. Box plot: the "icon effect" on time recognition.

The box plot on figure 10 tends to confirm that icons for angle 0°, 45° and 90° are the best sets. Users do not have difficulties to recognize them while the other icons are less recognized. The icons for angle 15° and 30° are the worst, what is consistent with the time of answer: we can already notice this in the box plot on figure 9. If we look at the pins raised on the matrix for these icons we can notice that they are near the reference point and the feeling may be too similar. Moreover, the reference point in the icon for angle 15° is mixed with the other raised pins. We assume that this can disturb the user insomuch he loses this reference point.

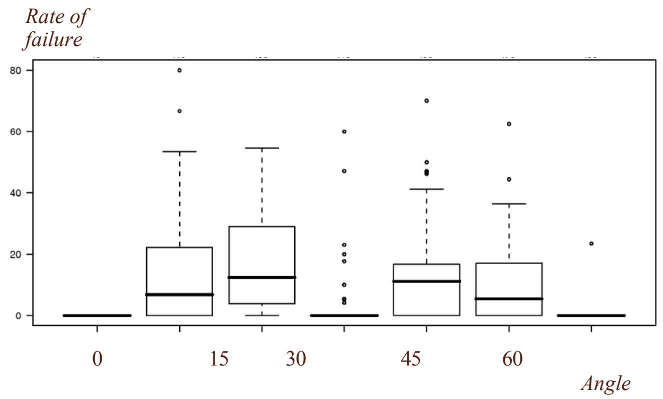


Fig. 10. Box plot: the rate of failure on icons.

We have to check the normality of the distribution and the equality of the variance. Thus, we decide to delete icons for angle 0°, 45° and 90° and perform again variance analysis (see figure 11).

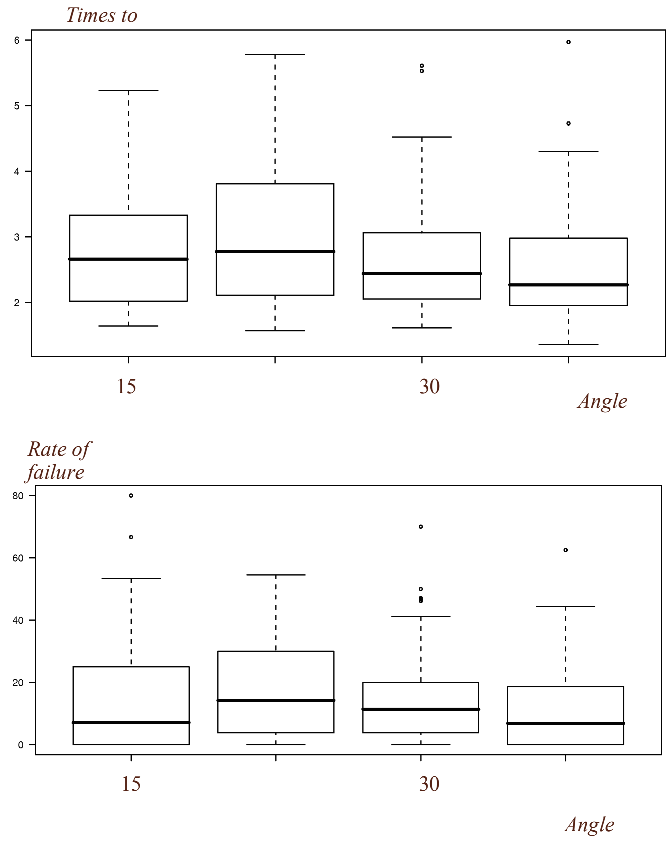


Fig. 11. “Icon effect” and rate of failure on reduced set

The second data file allows to perform a variance analysis based on the average time of answer and the rate of failure of each user for each angle during each session (figure 12).

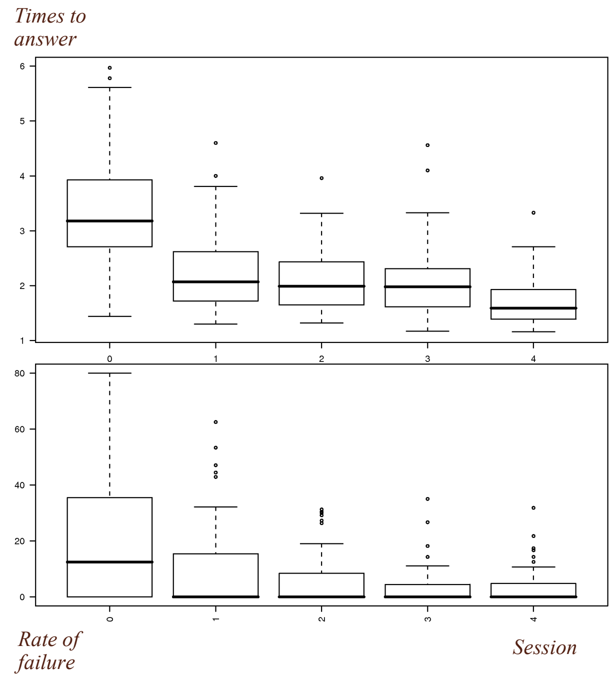


Fig. 12. Analysis by sessions

As the normality is not obtained, we delete the first session (figure 13). The box plot on the top of figure 13 shows that for sessions 1, 2 and 3 the medians are similar and the scattering of the values between session 2 and 3 is similar too. We can notice that, on the box plot on the bottom of figure 12, the rate of failure decreases gradually and the two last sessions (session 3 and 4) have a similar scattering.

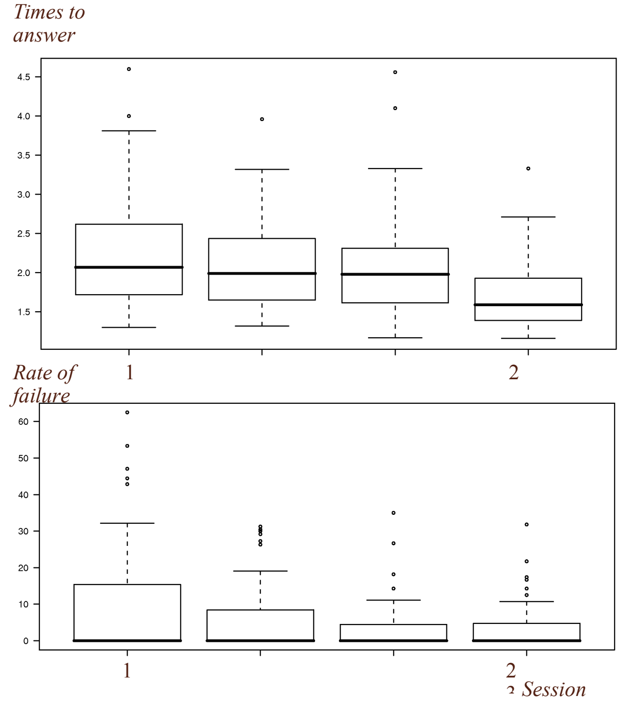


Fig. 13. Analysis by sessions without session 0

If we look globally at the figure 13, we can notice that whatever the icon is, the time of answer decreases with a rate of failure increasingly low. The users recognize better and better icons they had difficulties to recognize at the beginning. The recognition speed of these icons increased too. We can conclude that there is an effect of learning during the test.

If we now compare the errors between icons for angle 15°, 30°, 60° and 75° for all the users that did the five sessions, the more distinct icons are between 30° and 75°, 15° and 75°, 15° and 60° (see figure 14). For these sets of icons, main errors occurred during the first session. For all the other types of errors referred in the graph figure 14, errors are spread during all the session.

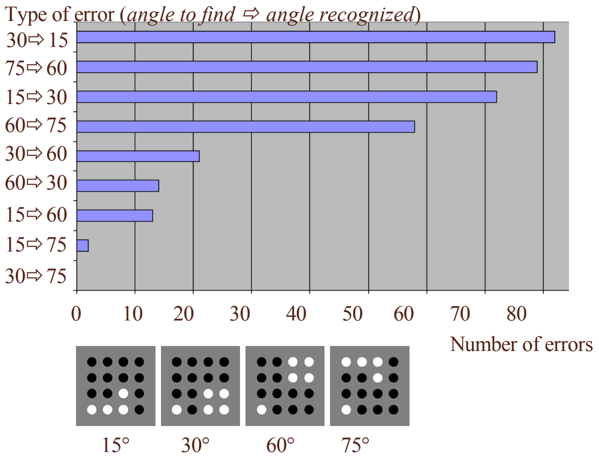


Fig. 14. Errors between icons for all sessions and all users

We assume that icons that seem to be confused for users are too similar in the tactile feeling. However, it is important to notice that even for these icons the result of recognition is better along the session.

Conclusion

In this paper, we present a new Application Programming Interface to manage the VTPlayer mouse. It is based on the MVC (Model-View-Controller) architecture and we can use it on various platforms (Windows, Mac, Unix). With this API, the programmer can easily develop applications based on icons cues dedicated to a large panel of users. The various views the API can manage allow the unique development of a common application for visually impaired users and sighted ones. In the second part of this paper, we test an application based on our API to experiment angle recognition with tactile icons managed by the VTPlayer mouse. We have chosen a set of tactile icons to represent the angles between 0° and 90° with a set of 15°. We assume that icons for the angle in the other quadrants can be obtained by symmetry. This should be tested in future work. Our icons have not a strong link with angle definition. The semantic link between icons and angles may disturb tactile recognition and it seems to be difficult to represent all angles with this constraint. Our hypothesis about a "session effect" is checked and we succeed in finding icons more difficult to recognize. Our last hypothesis about the low rate of failure for each icon is checked too: even for the worst icons to recognize, users tend to improve their learning and their results. Our test gives good results because even for icons difficult to recognize we show that each user improves his result along the session and increases his speed of recognition.

The preliminary tests presented have been done on sighted people. Before testing the angle recognition with visually impaired people to compare the previous results, we think that we can improve the set of icons to reduce ambiguity between icons that appears similar. A solution could be to use dynamic pins on the matrix i.e. to activate the reference point that was "hidden" in icon for angle 15°.

Acknowledgment

The authors would like to thank Pierre Collin, Cyril Hazotte, Timothée Seite and Jérôme Wax, students in Master degree, for their work on the API and the test experiment.

References

[1] European project Micole- IST-2003-511592, http://micole.cs.uta.fi.-

[2] Virtouch Ltd, http://www.virtouch2.com.

[3] G. Jansson and P. Pedersen, "Obtaining geographical information from a virtual map with a haptic mouse," in International Cartographic Conference, La Coruña, Spain, July 2005.

[4] S. A. Wall and S. A. Brewster, "Sensory substitution using tactile pin arrays: Human factors, technology and applications," Special Issue of Signal Processing on multimodal interfaces, 2006, to appear.

[5] Y. Eriksson, "How to make tactile pictures understandable to the blind reader," in *the Proceedings of the 65th IFLA Council and General Conference*, Bangkok, Thailand, August, 1999.

[6] T. Pietrzak, I. Pecci, and B. Martin, "Static and dynamic tactile directional cues experiments with vtplayer mouse," in *the Proceeding of the 6th EuroHaptics International Conference*, Paris, France, July 3-6 2006, pp. 63-68.

[7] A. Crossan and S. A. Brewster, "Two-handed navigation in a haptic virtual environment," in *CHI 2006: Proceedings of the SIGCHI conference on Human factors in computing systems*, Montréal, Québec, Canada, ACM Press, April, 2006, pp 676-681.

[8] C. Jacquet, Linux VTPlayer Driver (GPL), http://vtplayer.sourceforge.net/.

[9] Calle Sjoström. *Non-Visuel Haptic Interaction Design – guidelines and applications*, Thesis, 2002, 212 pages.