ICAT 2002
December 4-6, Tokyo, JAPAN

Multimodal Text Input in an Immersive Environment

Noritaka OSAWA[†*] and Yuji Y. SUGIMOTO[†]

[†]National Institute of Multimedia Education
[*]The Graduate University for Advanced Studies
2-12 Wakaba, Mihama, Chiba, 261-0014 JAPAN
{osawa,yuji}@nime.ac.jp

Abstract

We developed multimodal 3D widgets that enable a user to input textual annotations in an immersive environment. One widget utilizes both speech recognition and direct manipulation by hand. The user can input text by voice and choose a phrase, words or characters from the recognized candidates. The widget utilizes a 3D space. It displays not only the recognized candidates of one utterance vertically and horizontally, but also candidates of utterances in the depth direction. The user edits the inputted text by direct manipulation. The other widget is a virtual keyboard which has virtual 3D key buttons. The user can input a text by using the virtual keyboard.

Key words: textual annotation, speech recognition, direct manipulation, multimodal interface, 3D depth

1. Introduction

Cooperative work in a distributed immersive environment usually needs a mechanism that allows users to put annotations in the virtual world. Certain 3D symbols or icons can be placed only by direct manipulation. They can be used to give spatial marks in a virtual world. However, these symbolical annotations or concrete symbols are insufficient in that complex abstract annotations cannot be represented by them. Therefore there is a need for textual annotations in cooperative work.

Text input is also needed by applications such as immersive programming systems [1] and immersive 3D presentation systems. Immersive programming systems need textual input to give a name to a routine having an abstract functionality, since the abstract functionality cannot be represented by graphical icons very well. Immersive 3D presentation systems need text to explain the contents of a presentation.

It is possible to use an ordinary keyboard to input texts in an immersive environment but it is not easy to use the keyboard while wearing a sensor glove. It is also possible to use a usual keyboard to input texts on a desktop outside an immersive environment and to perform only spatial operations in an immersive environment; however, if one wants to add textual annotations in a virtual space, it is tedious and time-consuming to come and go between the two environments.

Recently, speech recognition technologies have been improved to the extent that it has become practical to use them in many applications. However, the accuracy of the speech recognition is not sufficient without post-editing because the recognized results are not necessarily what the user speaks. Therefore the use of only speech recognition is not practical for textual annotations in an immersive environment.

Hence we have developed new input widgets for textual annotations. One widget utilizes both speech recognition and direct manipulation. The widget recognizes voice-input and shows the candidates it has recognized in a 3D space. A user can choose a phrase, words, or characters from the recognized candidates by direct manipulation. It will be called *"the speech and hand-editing widget"* in this paper. A snapshot of the widget being used is shown in Figure 1. The other widget is a virtual keyboard that has virtual 3D key buttons.

We conducted an experiment to measure the efficiency of textual input using the developed widgets. Preliminary experiment results suggest that a combination of speech recognition and direct manipulation can improve the efficiency of textual input in an immersive virtual environment.

Figure 1: Multimodal text input

2. Related work

Multimodal interfaces have been actively studied. Speech and gesture interfaces have been studied to mainly let the user give a command to the system, in other words, for command inputs.

The *Put-that-there* system [2] integrates voice input with hand gestures in a 3D space in order to give a command for manipulation of virtual objects. The QuickSet system [3] is a 2D map application that uses speech input with pen-stroke gestures. The speech/gesture user interface for MDScope [4] allows a user to control a VMD (Visual Molecular Dynamics) system for structural biology by using vision-based hand gesture recognition. A multimodal control system for a whole earth 3D visualization system [5] also gives the user a control using a vision-based gesture pendant.

In command input systems like the ones mentioned above, the vocabulary of speech recognition is limited, because the systems can only recognize predefined words of commands, whereas the vocabulary for textual annotation should not be limited because the user needs to input text freely. Our widgets let the user input any word.

If the vocabulary for textual annotation is limited, the user may get frustrated or be forced to revise wording to express his/her intent successfully. In the first place, the user must know the allowed vocabulary; otherwise the user may consume time inputting unrecognizable words. Even if the user knows the vocabulary, the process of devising a word combination that expresses the user's intent may take much time. This impedes efficient textual annotation. In other words, limiting the vocabulary is not an answer to free and textual annotation. This is one of the differences between our widgets and existing speech/gesture systems for commands.

Voice and video recordings such as in Virtue[6] can be used for annotations. It does not need much time to record voice and video. However, it sometimes takes much time to replay the voice and video recordings and understand them. In other words, it is difficult to understand multiple annotations in a short period. We think that simple textual annotations would be better than the voice and video recordings to understand the annotations at sight. Therefore we investigate the input methods of textual annotations.

3. Speech and hand-editing widget

In this section, we describe the functionalities of the speech and hand-editing widget.

Figure 2 shows a screenshot of an experiment, just after the speech and hand-editing widget has been initialized. The upper part includes a message area for the experiment. The message area on the top with the white background shows a text task to input. The other area on top (confirmation area) displays the confirmed input text for the experiment. The concave box below the confirmation area is an editing box. The result of the input will be stored in this box. The candidates recognized by the speech recognition engine will be placed in the empty space in the center. A dust bin is used to discard unwanted elements. The control panel has buttons for selecting a set of candidates and controlling the editing box, for example, to show the next set of candidate or to clear the editing box. The temporary buffer box can contain elements selected from the recognized candidates. Those elements can be used for future input.

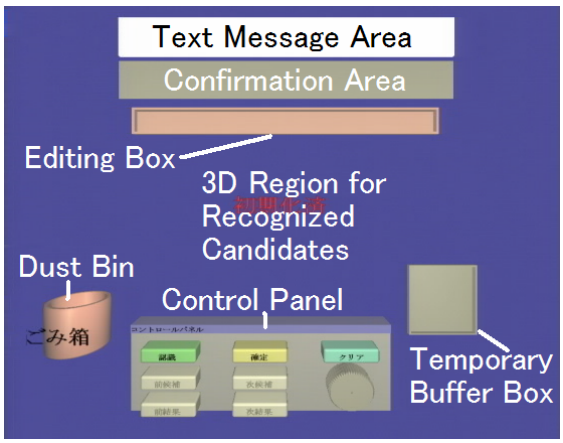

Figure 2: Speech and hand-editing widget after initialization.

3.1. Speech recognition

After initialization, utterances are recognized by the speech recognition engine. While an utterance is being recognized, a message in Japanese, "recognizing" is displayed as shown in Figure 3. Even if the recognition is in progress, another utterance can be inputted.

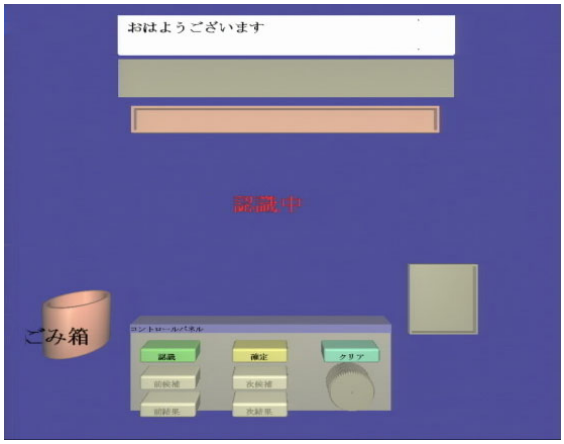

Figure 3: A screenshot of when an utterance is being recognized.

3.2. Candidates

The speech recognition engine's candidates for one utterance are displayed on a plane as shown in Figure 4. The candidates are placed vertically. In the experiment, four candidates were simultaneously displayed on a plane. The number of candidates to be displayed can be changed. Another set of candidates can be displayed by pushing a button on the control panel.

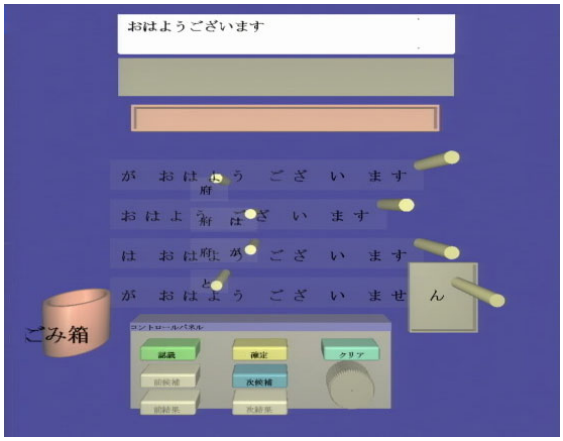

Figure 4: A screenshot displaying the recognized candidates.

3.3. 3D depth

The widget utilizes the depth dimension to show the recognized candidates of multiple utterances. If the recognized candidates of only one utterance are displayed, certain noises may start a new speech recognition and erase the correctly recognized result. If the new utterance lets the system accept the wrongly recognized result as an annotation, the annotation caused by noises is needed to be corrected. This degrades the efficiency of the text input and increases the frustration of the user. The developed 3D widget places previous candidates on a plane at the back, as shown in Figure 4. The user can choose the result of the previous utterance as well as the current result. The user can understand the order of the recognized utterances on the basis of the depth of the candidates.

3.4. Nested structure

Candidates are represented by nested structures of concave boxes. A recognized candidate is decomposed into nested elements, or regions: a phrase, words (or word-like character groups), and characters. A phrase includes words or its similar units. A word includes more than one character.

The user can choose a phrase, words, or characters from the recognized candidates by direct manipulation (Figure 5). Elements of the recognized candidates can be used for textual input. Speech recognition results sometimes contain incorrect candidates and unnecessary prefix or postfix words caused by noise. The developed widget allows the user to choose correct and useful parts of the recognized results.

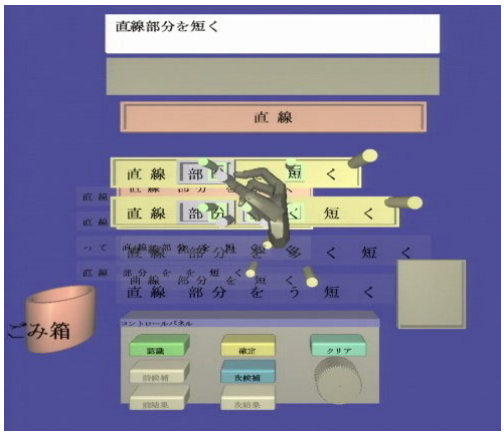

Figure 5: Nested structure of recognized candidates

3.5. Hand editing

As can be seen in Figure 5, each region has its handle, although it is not always displayed. This is because handles may limit the visibility within a structure or become visual clutter. Therefore, the handles are generally displayed on demand. When the user places his/her hand near the candidates, nested regions near the hand will display their structures and handles as shown in the figure.

The user can copy an element by pinching its handle between his/her thumb and forefinger and moving it from the candidate to the target region. When an element is picked, the color of the element changes to light green, as shown in Figure 6.

When the picked element can be placed in the target region, the color of the target region becomes red, as shown in Figure 7. The picked element is copied using the target cursor of the target region by releasing the element when the target region is red.

If the user picks an element between his/her thumb and forefinger in the editing box, the element in the editing box is not copied, but moved. This convention of copying and moving that depends on the regions can be changed, but we think that this convention eases the editing operations.

An unnecessary element can be discarded by moving the element into the dust bin, as shown in Figure 8. By using this operation, one can easily delete the unnecessary prefix or postfix caused by noises in speech recognition.

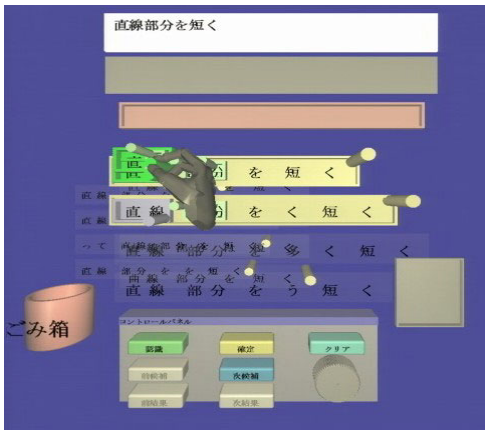

Figure 6: Picking a word

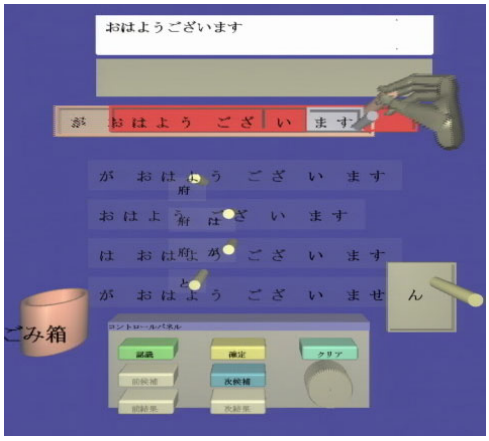

Figure 7: Moving a phrase into the editing box

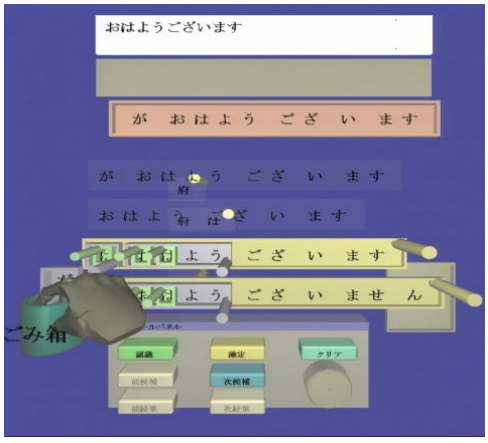

Figure 8: Discarding an element into the dust bin

4. Virtual keyboard widget

A virtual keyboard widget has Japanese kana-keys and QWERTY-based keys as shown in Figure 9 and Figure 10, respectively. They can be switched by pressing a mode key.

A usual Japanese text is composed of *kana* characters (phonograms) and *kanji* characters (Chinese ideograms). We will call it "mixed text" in this paper. A mixed text can be translated into a kana-only text, which can be understood by Japanese but may have some ambiguities of meanings.

The kana-kanji conversion system or kana-kanji input method engine is usually used to input a mixed text by using an ordinary kana keyboard or QWERTY keyboard. Although it takes a number of steps to convert a sequence of kana characters into a Kanji character, the mixed text is easy to read and most Japanese prefer the mixed text to the kana-only text.

The developed virtual keyboard widget does not currently support the kana-kanji conversion. Therefore people cannot input kanji characters using the virtual keyboard widget, although it usually takes less time to input a kana-only text because conversion operations are omitted. The speech recognition system does however produce a mixed text in Japanese.

Figure 9: Japanese kana keys

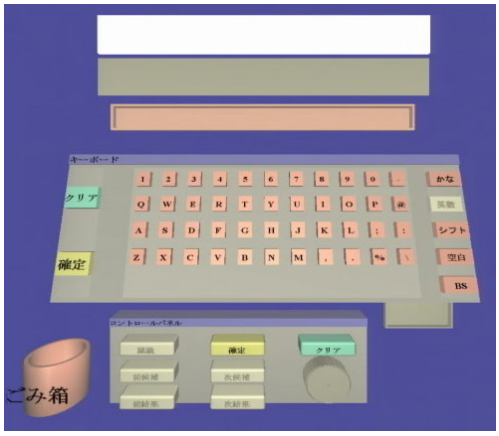

Figure 10: QWERTY-based keys

5. Prototype System

The hardware configuration of the prototype system is given in Figure 11. A textual input application for the experiment runs on a PC workstation (Dell Precision 530 with dual 2-GHz Pentium 4 Xeon processors and a 3DLabs Wildcat II 5110 graphics board supporting dual displays). A Six-DoF position tracker (Polhemus Fastrak) and sensor gloves (Virtual Technologies CyberGlove) are used to detect the position and motion of the user's body and hands. A notebook-type PC (IBM ThinkPad) is used for speech recognition. The PCs are connected through a LAN.

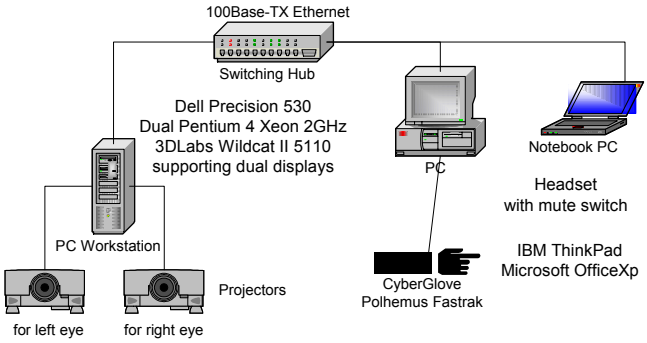

Figure 11: Hardware configuration of prototype system

5.1. TEELeX

The immersive virtual environment system called TEELeX (Tele-Existence Environment for Learning eXploration)[7] was used as an immersive projection display. TEELeX is a kind of surround display system such as the CAVE. Each screen of TEELeX measures 3 meters by 3 meters. The circular polarization method is used to give a stereoscopic view; that is, TEELeX uses a passive stereo system.

5.2. It3d class library

The speech and hand-editing widget and the virtual keyboard widget were developed using the *it3d* class library. *It3d*[1] is an interactive toolkit library for 3D applications utilizing artificial reality (AR) technologies [8]. It was implemented using the Java language and the Java 3D class library to enhance portability. *It3d* makes it easy to construct 3D applications that are portable and adaptable. It consists of three sub-libraries: an input/output library for distributed devices, a 3D widget library for multimodal interfaces, and an interaction recognition library.

The input/output library for distributed devices includes interfaces for CyberGlove, Polhemus Fastrak, and the speech recognition engine (Microsoft Speech API version 5). The 3D widget library for multimodal interfaces includes basic 3D widgets, such as 3D buttons and 3D panels. The interaction recognition library helps the user to pick an element by hand.

[1] *it3d* can be accessed through <http://www.nime.ac.jp/it3d> (in Japanese) or <http://www.nime.ac.jp/it3d/index-e.html> (in English).

6. Experiment

An experiment was carried out to compare multimodal input using the developed widgets with voice-only input. The speech recognition engine (Microsoft Japanese ASR Version 5 Engine, that is, MSASRJapanese) included in Microsoft Office XP was used. It is a state-of-the-art speech recognition engine. A participant wore a headset with mute switch and volume control (Logicool stereo headset A-002) in the experiment.

In order to evaluate performance of multimodal interface, the time that participants took to input a short phrase was measured.

6.1. Participants

Six participants took part in the experiment. All of them were Japanese students majoring in either engineering or language studies. They were between 19 and 23 years old, and had normal vision. They had some VR experience, but had not experienced speech recognition applications.

6.2. Procedures

Before the experiment, each participant enrolled his/her voice into the speech recognizer.

After the voice enrollment, a participant was given three tasks as practice for each input method. The tasks required them to input short Japanese phrases or sentences. They inputted the text using (A) speech recognition only, (B) the speech and hand-editing widget, and (C) the 3D virtual keyboard. The practice tasks are shown in Table 1.

For speech-recognition-only input, Microsoft Word 2002 and its speech input functions were used. In the speech-only task, a participant inputted a text by voice in voice-dictation mode and edited the inputted text in voice-command mode. The participant needed to switch between two modes by voice. The speech-recognition-only experiment was performed outside TEELeX, that is, in an office. A participant sat in a seat and used the speech recognition system. On the other hand, other experiments were preformed in TEELeX, where a participant stood up, and wore stereoscopic glasses and a sensor glove for hand motion detection.

As explained before, the current 3D virtual keyboard widget does not support kana-kanji conversion functions. Therefore a participant inputted the kana-only text equivalent to the mixed text. A kana-only text input needs fewer keystrokes than a mixed text input, and thus takes less time.

After practice, a participant was given ten test tasks. The test tasks are shown in Table 2. We assumed that the developed widgets would be used for cooperative work in a virtual 3D space. Therefore, we chose short phrases related to the construction and modification of a 3D scene.

Table 1: Practice tasks

#	Text in Japanese	Translation in English
1	おはようございます	Good morning
	今日もよい天気です	It is fine today
2	合計１８７０円です	total is 1870 yen
	２千円ちょうど	Just 2000 yen
3	降水確率50%です	the probability of rain is 50%
	XたすYはZ	X plus Y equals Z

Table 2: Test tasks

Type of task	Text in Japanese	Translation in English
Name	オブジェクト2	Object 2
Comment1	デザイン再検討	Review design
Comment2	綴りを訂正	Correct spelling
Attribute1	緑を濃く	Deepen green
Attribute2	直線部分を短く	Shorten a straight part
Structural direction 1	これらを連結する	Connect these
Structural direction 2	5個に分離する	Separate into 5 pieces
Structural direction 3	要素をグループ化	Group elements
Structural direction 4	上へ移動	Move upward
Structural direction 5	タイトルを中央揃え	Center the title

6.3. Experimental results

The time which it took for each participant to input a text was measured. The results of the experiment are shown in Table 3. Although the result of the experiment indicated no statistically significant difference between them, the average time in (B) was shorter than in (A). This suggests that the combination of speech recognition and hand direct manipulation, that is, the multimodal interface, is superior to the speech-recognition-only interface.

The average time in (C) is shorter than in (A) and (B), but (C) allowed the user to input only kana characters (phonograms), whereas (A) and (B) enabled the user to input kana-and-kanji texts, i.e., mixed with phonograms and ideograms. Kana-and-kanji texts are usual in Japanese as explained.

The dispersion among the participants was larger in (A) and (B) than in (C). This shows that the virtual keyboard enables one to input a text steadily.

The speech-recognition-only input is fast in some cases although the total minimum of 10 tasks does not show that. If one utterance completes a task, it is fast. However, speech recognition engines sometimes could not recognize the spoken input properly. In some cases, it took a very long time to input the correct word.

Table 3: Experimental Result

Total of 10 tasks		*(A) Speech recognition only*	*(B) Speech and hand-editing widget*	*(C) Virtual keyboard (no kana-kanji conversion)*
Time (s)	Average	481.5	459.2	363.5
	Standard Dev.	267.7	127.5	71.1
	Max.	985	645	449
	Min.	299	282	269

6.4. Subjective evaluation

After completing their tasks, the participants were asked questions about the text input methods on the subjective evaluation questionnaire.

They rated the methods on a scale of 0 to 9 for the characteristics listed in Table 4. The result summary of the questionnaire is given in Figure 12.

Speech and hand-editing widget is better than other input methods except characteristics B. The participants answered that among three methods, the virtual keyboard is the best method for precise input.

Table 4: Characteristics rated in the questionnaire

A	Appropriateness of fast input, from 0 (slowest) to 9 (fastest)
B	Appropriateness of precise input, from 0 (most incorrect) to 9 (most correct)
C	Usability of text input, from 0 (worst) to 9 (best)
D	Fatigue, from 0 (most fatigued) to 9 (no fatigue)
E	Satisfaction, from 0 (completely unsatisfactory) to 9 (completely satisfactory)
F	Future use desire, from 0 (no desire at all) to 9 (very much desire)

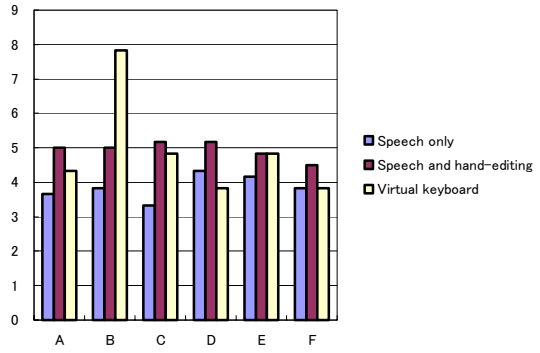

Figure 12: Average scores of the questionnaire

7. Discussion

7.1. Speech Recognition

Speech input is fast if recognition is precise. The short phrases used in the experimental tasks could be spoken in one or two seconds. However, most participants failed to input some tasks in one utterance trial. If they could not input a text in a task at the first time, it often took a lot of time to input a correct text or to edit the incorrectly recognized text.

7.2. Speech and hand-editing

Most participants reported that they felt little fatigue if their utterances were correctly recognized after one trail. This is because it took only a little time to input a phrase and they did not need to move their hands. However, most participants complained that they felt fatigued when their utterances were incorrectly recognized.

To stop noise from being input, the participants needed to turn off the microphone by using the mute switch of the head set. Some participants sometimes forgot to turn off the microphone, and certain noises subsequently caused unwanted recognitions, which often eliminated correctly recognized candidates.

Given the experimental setting, all participants could easily understand the manipulation of the nested structures and edit a text by hand.

7.3. Virtual keyboard

The virtual keyboard enabled the participants to input text steadily. This is shown by the standard deviation of the time to input texts. Some participants complained about hand fatigue, although one of them reported that despite the fatigue, it was fun to input text using the virtual keyboard.

7.4. Switching between modalities

In this experiment, the speech and hand gestures did not need to be coordinated. However, it seemed that it was difficult for some participants to switch between speech and hand-editing appropriately.

It is possible to use elements of the inputted phrases by using hand-editing in the tasks. Previous candidates as well as current candidates are shown by the interface, but some participants tended to speak a task's text over again, even if the task could be completed by hand-editing the displayed candidates. This, we feel, was partly due to insufficient practice. A participant performed only three tasks in the practice, each of which included two short phrases. We should thus repeat the experiment after sufficient training, in order to know whether the proper use of modalities is difficult for some people or whether the sufficient training is required to use modalities properly.

Apart from our experiment, in a kana-kanji conversion using the ordinary keyboard, we think that some people tend to erase incorrectly recognized clauses in Japanese sentences and "re-input" shorter phrases even if the recognized clauses can be changed by using commands. This seems to be the case in the above situation. We thus need another investigation.

8. Future Work

We have a plan to add kana-kanji conversion functions to the 3D virtual keyboard widget. This will help us to input a natural Japanese text. Moreover, the comparison between methods using speech recognition and those using a virtual keyboard would be clearer.

We will also implement predictive and suggestive functions in the virtual keyboard widget that supports kana-kanji conversion. We think that those functions increase the efficiency and decrease the fatigue in an immersive environment.

The current widgets do not have tactile feedback. Therefore, a participant tends to want to see his/her fingers when he/she pushes a key. Otherwise it is difficult to know if the intended key has been pressed or not. We will therefore implement tactile feedback functions in the widgets.

9. Conclusion

We developed multimodal 3D widgets for textual annotation in an immersive environment. One widget allows one to use speech recognition and hand-editing. The other is a virtual keyboard widget.

The experiment suggests that the speech and hand-editing widget is a good method for text input, although the difference between the widget and others is not statistically significant. The virtual keyboard lets users input textual annotations steadily.

The developed multimodal 3D widgets can help us to input textual annotations in an immersive virtual environment.

Acknowledgments

This research was partially supported by a Grant-in-Aid for Scientific Research (14380090) in Japan.

The toolkit library "*it3d*" was developed with funding from the Support Program for Young Software Researchers in 2000, which was implemented by the Research Institute of Software Engineering (RISE) commissioned by the Information-technology Promotion Agency (IPA) in Japan.

References

[1] Noritaka Osawa, Kikuo Asai, Yuji Y. Sugimoto, and Fumihiko Saito: "A Dancing Programmer in an Immersive Virtual Environment," *Symposia on Human-Centric Computing Languages and Environments* (HCC2001), pp.348-349, 2001.

[2] R. A. Bolt: "Put-That-There: Voice and gesture at the graphics interface," *ACM SIGGRAPH Computer Graphics*, Vol. 14, No. 3, pp. 262-270, 1980.

[3] P.R. Cohen et al.: "QuickSet: Multimodal Interaction for Distributed Applications," *Proc. Fifth ACM Int'l Multimedia Conference*, pp. 31-40, New York, 1997.

[4] Rajeev Sharma, Michael Zeller, Vladimir I. Pavlovic, Thomas S. Huang, Zion Lo, Stephen Chu, Yunxin Zhao, James C. Phillips, Klaus Schulten: "Speech/Gesture Interface to a Visual-Computing Environment," *IEEE Computer Graphics and Applications*, pp.29-37, March/April 2000 (Vol. 20, No. 2) (2000).

[5] David M. Krum, Olugbenga Omoteso, William Ribarsky, Thad Starner, Larry F. Hodges: "Speech and gesture Multimodal Control of a Whole Earth 3D Visualization Environment," *IEEE TCVG Symp. on Visualization* (VisSym'02), Barcelona, Spain, (2002).

[6] Eric Shaffer, Daniel A. Reed, Shannon Whitmore, Benjamin Schaeffer: "Virtue: Performance Visualization of Parallel and Distributed Applications," *Computer*, Vol. 32, No. 12, pp. 44-51, December 1999.

[7] Kikuo Asai, Noritaka Osawa, and Yuji Y. Sugimoto: "Virtual Environment System on Distance Education," *Proc. of EUROMEDIA '99*, pp. 242-246, 1999.

[8] Noritaka Osawa, Kikuo Asai, and Fumihiko Saito: "An Interactive Toolkit Library for 3D Applications: it3d," *Eighth Eurographics Workshop on Virtual Environments* (EGVE2002), pp.149-157, May 2002.