ICAT 2005
December 5-8, Christchurch, New Zealand

Semantic 3D Object Manipulation using Object Ontology in Multimodal Interaction Framework

Sylvia Irawati[a,b], **Daniela Calderón**[a,b], **Heedong Ko**[a,b]
[a] Department of Human Computer Interaction, University of Science and Technology
[b] Imaging Media Research Center, Korea Institute of Science and Technology
39-1 Hawolgok-dong, Seoul, 136-791, South Korea
{sylvi, dcalderon, ko}@imrc.kist.re.kr

Abstract

This paper presents a multimodal interaction framework for semantic 3D object manipulation in the virtual reality. In our framework, interaction devices such as keyboard, mouse, joystick, tracker, can be combined with speech utterance to give a command to the system. We define an object ontology based on common sense knowledge which defines relationships between virtual objects. By taking into account the current user context and the object ontology, semantic integration component integrates the interpretation result from input manager, and then sends the result to the interaction manager. That result will be mapped into a proper object manipulation. Thus, the system can understand the user intention and assist him for achieving his goal in the handling process, instead of relying entirely on the user's control upon the interaction device and the object, avoiding nonsensical manipulations.

Key words: multimodal interaction, 3D interaction, object manipulation

1. Introduction

Object manipulation is an important task in the virtual world. Manipulation means modifying physical properties of objects such as its position, orientation, scale, etc. Many 3D interaction devices and techniques have been developed to improve object manipulation in virtual reality (VR) [1]. However, controlling the devices for the manipulation of virtual objects with physical metaphor in VR still is not an easy task in the absence of the force feedback, little coordination between hand and eye, and the noticeable physical fatigue.

Recent approaches to provide users with more natural interaction methods in virtual environment applications have shown that more than one mode of input may be beneficial and intuitive between humans and computer applications like human to human and human to environment interactions. A multimodal system supports communication through different modalities such as linguistic, visual, audio, gesture, and spatial [2]. There have been many works in improving 3D object manipulation using multimodal interaction. However the management of ambiguous constraints across modalities remains a difficult problem.

In this paper we present M^{3}I framework, multimodal manipulation for 3D Interaction framework which uses object ontology to solve ambiguity of object manipulation in the virtual work. The object ontology stores the information about constraints how an object may interact with other objects. It also describes the spatial characteristics of the object related to other things in the environment. Most of the objects in the real world are not placed arbitrarily in space. They are restricted by physics law, such as gravity force that is universal; in addition, they may follow human conventions, sometimes called common sense, for example, ceiling lamps are almost never placed permanently on the floor, and the pictures are always hung on the wall. Their location is relatives to the other object. Since the objects are resting on a plane, the objects have a maximum of three degrees of freedom in practice. The object ontology can describes those kinds of relationships using common sense. It can be used to assist the user in placing and manipulation objects in virtual environment as it will be in a real one. For example, a table must stand on the floor at all times. When a user interacts with the table by translating or rotating it in the scene, it never leaves the floor.

M^{3}I framework provides multimodal interface to interact with virtual world. By taking into account the current user context and the object ontology, it combines several modes of interaction to create an integrated interpretation to make object manipulation in 3D virtual world more intuitive.

2. Related Work

A number of researchers have addressed the issues of object manipulation in 3D environments. Bukowski and Sequin [3] enhanced the object manipulation by combining almost realistic-looking pseudo-physical behavior and idealized goal-oriented properties, called object associations. However, they utilize only a little knowledge about where an object is placed naturally. Smith and Stuerzlinger [4] enhanced the system by

attaching the semantic information to objects in the form of labels “binding areas” and “offer areas”. If the labels of two objects are compatible, objects are placed together by connecting binding areas to offer areas. Additionally, Xu [5] combined automatically–generated placement constraints, pseudo–physics, and a semantic database to guide the object placement.

Many initial steps were taken that motivated building immersive multimodal system. Multimodal 3D interaction that includes speech dates back at least to Bolt’s pioneering Put-That-There system [6], in which speech was integrated with 3D magnetic tracking of a user’s arm in order to manipulate a 2D world. Althoff et al. [7] present a multimodal interface for navigating in arbitrary virtual VRML worlds. The system provides intuitive input by command and natural speech utterances as well as dynamic head and hand gestures. Kaiser et al. [8] describe an approach to 3D multimodal interaction in immersive augmented and virtual reality environments that accounts for the uncertain nature of the information sources. The resulting multimodal system fuses symbolic and statistical information from a set of 3D gesture, spoken language, and referential agents.

Although there have been many works in improving 3D object manipulation using multimodal interaction, the management of ambiguities across modalities is still become a problem. In this paper, we discuss a framework that maintains the user context and the object ontology to solve ambiguity across modalities in 3D object manipulation.

3. Naver System Architecture

NAVERLib [9] is a microkernel architecture framework in the distributed network environment. It provides libraries for a variety of interactions, interface, and virtual contents that can be composed in the virtual reality environment. NAVERLib has a device manager to manage the peripheral interaction device. Device manager communicates with the device server in which interaction devices are connected. This communication is configured using a script file, making the device management more flexible. Each interaction device may have different update rate, dependent on its characteristic. The device manager gets the device values which is sent by the device server and then routes those values to the device interpreter. M^{3}I interprets those values based on the configuration file.

4. Multimodal Interaction Framework

A schematic diagram of multimodal interaction framework is shown in Figure 1. The framework, described below in detail, consists of four main components: recognition component, I/O manager component, semantic integration component, and interaction manager.

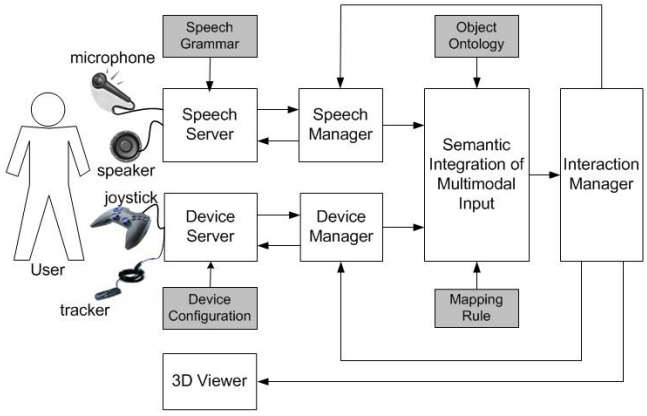

Figure 1. Multimodal Interaction Framework

4.1. Recognition Component

The recognition component is responsible for capturing the raw input from user. As shown in figure 1, it consists of speech server and device server.

Speech server has a speech engine which is used to recognize speech input and convert text to speech. It matches the input against the grammar to produce a literal text of the detected input. The result is sent to the speech manager for further processing. A part of a context-free grammar we are using is given below. It is described in Backus-Naur form (BNF).

```
<CMD>     := <SELECT>|<LOCATE>|<POS_ORI>
<LOCATE>  := <PM> <NOUN_PHRASE> <PREP_PHRASE> |
             <PM> <NOUN_PHRASE> <ADVERB>
<POS_ORI> := <TR> <NOUN_PHRASE> to the <LR> |
             <TR> <NOUN_PHRASE> <DIRECTION>
<SELECT>       := select
<PM>           := put | move
<TR>           := translate | rotate
<NOUN_PHRASE>  := <NOUN>|<PRONOUN>|
                  <NOUN> <PRONOUN>|
                  <ARTICLE> <NOUN>|
                  <ARTICLE> <ADJECTIVE> <NOUN> |
                  <PRONOUN> <ADJECTIVE> <NOUN> |
<PREP_PHRASE>  := <PREPOSIION> <NOUN_PHRASE>
<ADVERB>       := here | there
<LR>           := left | right
<DIRECTION>    := forward | backward |
                   downward | upward
<NOUN>         := table | lamp | ball | trash |
                  teapot |floor | wall | ...
<PRONOUN>      := it | this | that
<ARTICLE>      := a | the
<ADJECTIVE>    := small | big | ...
<PREPOSITION>  := on | near | ...
```

Device server is used to communicate with peripheral devices, such as, joystick, tracker, wand, etc. It can capture the device values and send the feedback values to the devices based on the configuration file. The device values are sent to the device manager for further processing.

4.2. I/O Manager

The I/O manager component consists of speech and device manager. Speech manager communicates with the

speech server to get the input sentence in the literal text and send the literal text to the speech server to be converted to speech. The device manager communicates with device server to get the device values and it is also responsible for sending the feedback values to device server. Both the speech manager and device manager route their result to the semantic integration component to be combined into a single interpretation.

4.3. Semantic Integration Component

The role of semantic integration component is integrating the input from various modalities to single interpretation. Given the literal text by the speech manager, it maps the text to the action command, selected object, new target location and the movement direction. It also maps the device values into meaningful values, such as the user gaze direction or user pointed hand direction, based on the mapping rule which is defined. The history of user interaction is also stored in the event log. It is used together with the user head pose and/or user hand pose to determine the user context. It needs to be maintained so that the system can understand the user intention properly.

We define the object placement constraint in the object ontology. It determines the possible location of the object relative to other objects. The object ontology and the user context are used to find the object correlates for the semantic meaning of deictic terms, such as "this", "that", "here", and "there", in the user utterance. A part of an object ontology that we are using is given below:

```
<ontology>
  <Table>
    <location>on_the_floor</location>
  </Table>
  <Lamp>
    <location> on_the_table | on_the_floor
    </location>
  </Lamp>
  <Picture>
    <location> on_the_wall </location>
  </Picture>
  ...
</ontology>
```

Object ontology has information about virtual objects and spatial relationships between them. As shown above, the ontology may include common sense terms referring to objects and relationships to be encountered on the scene. Such as "location of" a "table" is "on the floor". This relation describes the "location of" objects in real-world environment. The object ontology can be implemented in such common reasoning engine, like OpenCyc [10], a general knowledge base and common sense reasoning engine. By adding the object ontology to the current database, it is possible to query directly to the object ontology database finding the possible relationship among virtual objects. That ontology provides the information which can be used to solve the ambiguity in 3D interaction.

We show how the semantic integration work in the following example, the user points to certain direction and gives the command "put the picture there". The integration component has the information about user pose and user hand pose, and then it finds the objects which are located in the user hand pointed area. Suppose there is a table between the user hand and the wall. M^{3}I will check the ontology for object "picture", since the relation between picture and table is not described in the object ontology, the system can understand that the user wants to put the picture on the wall, not on the table. It shows that the object ontology can be used to solve ambiguity in finding the correlated object with deictic term "there". This ambiguity refers to either the user is pointing the table or the wall.

The object ontology also can be used to restrict the object manipulation. For example, the user gives the command: "put the lamp on the wall". Since, the relationship between lamp and wall is not described in the ontology, the system rejects the user command (it does nothing and sends the notification to the user).

The result of the integration component is the action command, such as put, translate, rotate, move, the selected object, the new target location, and the movement direction.

4.4. Interaction Manager

Given the action command, the selected object, the new target location, and the movement direction, the interaction manager has to find the proper translation and rotation to fulfill the user command and then shows the result in 3D Viewer by translating or rotating the selected object using those values.

5. Implementation

We have implemented a prototype using M^{3}I framework for object manipulation in virtual reality. Our implementation is based on NAVERLib [9] which uses VRPN (Virtual Reality Peripheral Network)[11] to communicate with peripheral devices. For speech recognition, we use Microsoft Speech API 5.1. We defined the grammar in the XML file according to Microsoft SAPI 5.1 grammar format.

In order to evaluate our approach, we make a prototype which simulates a room with several objects inside. Each object has a name, a current pose, an object size, and an object parent, which refers to the object that becomes a based of the current object. The object parent owns the child meanwhile the child does not have ownership on the parent. The example is shown in the following example, suppose that the lamp is located on the table, the table is the parent of the lamp. If the user moves the lamp to another valid position, only the lamp will move. However, if the user moves the table, the lamp will move together with the table. Figure 2 shows the virtual room which consist of several objects. The corresponding object tree is shown in Figure 3.

The user can interact with the system by giving an input command and control the arrow representing the hand avatar using a joystick. The process of maintaining the user context in the M^{3}I is shown in the following example:

1. Given the speech command "Put the lamp on this table". M^{3}I tries to find the previous selected object that correlated with the "lamp" using an object manipulation log of the most recent events and objects manipulated. If it is found, M^{3}I understands the object "lamp" that the user means is the previous selected "lamp". If it is not found, M^{3}I finds the object "lamp" that is located in the user gaze direction area. If there is more than one lamp, it chooses the nearest lamp from the user position. After that, it finds the object "table" that located in the hand pointed area. If there is more than one table, it chooses the nearest table from the user position. This selection correlates with deictic term "this".

2. Given the speech command "Put the teapot here". M^{3}I finds the object "teapot" using the same way as described in the example 1. It finds the object that has relationship with the teapot in the object ontology, and then it checks the location for each object which is found. It selects the nearest object which is located in the hand pointed area. This selection is related with the deictic term "here".

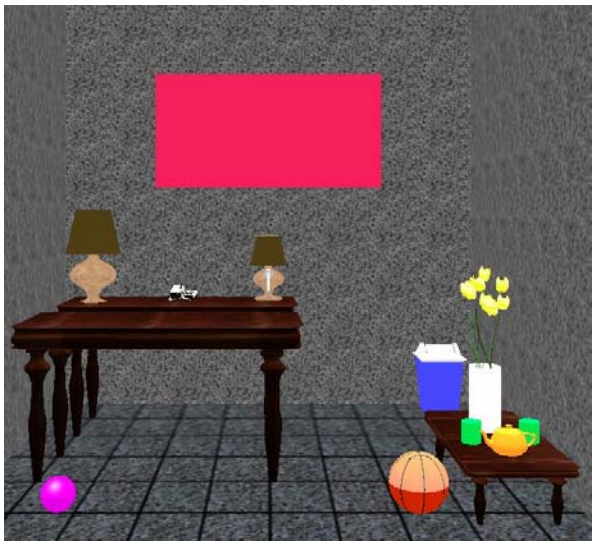

Figure 2. The virtual room

There are four actions that can be understood by the system, "select", "locate", "translate", and "rotate". As the initial result, the system can understand the simple command such as, "Put the lamp on this table", "Move the small table forward", "Rotate the table to the left", "Put the teapot here", "Put the picture there", etc. M^{3}I can understand whether the user utterance is related with the previous command using an event log of the objects that have been manipulated recently and it uses the object ontology to find the meaning of the deictic terms.

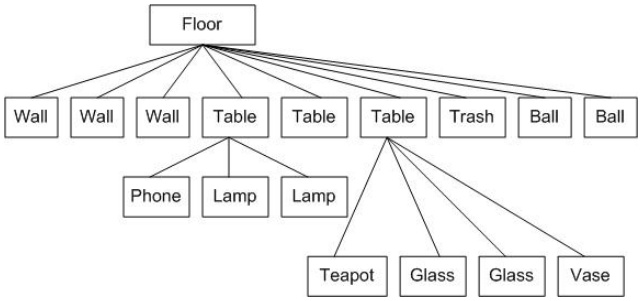

Figure 3. The object tree

6. Conclusion

We proposed M^{3}I framework for semantic 3D object manipulation in the virtual reality. The system can understand the simple natural language used to manipulate the object. We defined the object ontology which described the relation between objects to solve ambiguity in the user command using common sense knowledge. This ambiguity as mentioned before is related to where to put the things in spatial terms and for this we used the common sense knowledge. As the result, the system can understand the user intention and assist the user for doing the object manipulation. Thus, the object manipulation can be done more intuitively.

M^{3}I framework is suitable for applications which may have ambiguous input, like speech input. Since the system has to interpret the user input and do a cognitive process for solving ambiguity, this framework is applicable for applications which do not require very fast system reaction. It also can be used together with other 3D interaction techniques (e.g. ray casting, Go-Go interaction technique) to increase the effectiveness of system control tasks. The user can keep his attention focused on his main activity (object manipulation activity) while he controls the system (e.g. changing the interaction mode) so that it can decrease the user cognitive load. In the future, the object ontology can be extended to more complex object spatial relationship and M^{3}I framework can be applied to other domain application.

7. Acknowledgement

This research is supported in part by the Ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

8. References

[1] Doug A. Bowman, Ernst Kruijff, Joseph J. Laviola, JR, Ivan Poupyrev, "3D User Interfaces Theory and Practice", Addison-Wesley, Person Education, Inc., Boston, 2004.

[2] Schomaker, L., Nijstmans, J., Camurri, A., et al., "A Taxonomy of Multimodal Interaction in the Human Information Processing System", Esprit Basic Research Action 8579 Miami, 1995.

[3] Bukowski, R., and Sequin, C. "Object Associations", *ACM Symposium Interactive 3D Graphics*, pp. 131-138, 1995.

[4] G. Smith, W. Stuerzlinger, "Integration of Constraints into a VR Environment", *VRIC, Virtual Reality International Conference*, 2001.

[5] K. Xu, "Automatic Object Layout using 2D Constraints and Semantics", Master's thesis, University of Toronto, 2001.

[6] R. Bolt, " Put-that-there: Voice and gesture at the graphic interface," SIGGRAPH-*Computer Graphics*, 1980.

[7] Frank Althoff, Gregor McGlaun, Bj ¨ orn Schuller, Peter Morguet and Manfred Lang, "Using Multimodal Interaction to Navigate in Arbitrary Virtual VRML Worlds", *ACM Symposium on Perceptive user interfaces*, 2001.

[8] E. Kaiser, A. Olwal, D. McGee, H. Benko, A. Corradini, X. Li, P. Cohen, and S. Feiner, "Mutual Disambiguation of 3D Multimodal Interaction in Augmented and Virtual Reality", *ICMI –PUI*, pp. 12-19, 2003.

[9] C.H Park, H.D Ko, T. Kim, "NAVER: Networked and Augmented Virtual Environment aRchitecture; design and implementation of VR framework for Gyeongju VR Theater", *Computers & Graphics* 27, pp. 223-230, 2003

[10] OpenCyc, http://www.opencyc.org/

[11] Russell M. Taylor II, Thomas C. Hudson, Adam Seeger, Hans Weber, Jeffrey Juliano, Aron T. Helser, "VRPN: A Device-Independent, Network-Transparent VR Peripheral System", *ACM International Symposium on Virtual Reality Software and Technology (VRST 2001)*, Berkeley, USA, 2001.