A New Communication Network Model for Chat Agents in Virtual Space

Jong-Woo Kim[1], Seong-Hyun Ji[2], Seon-Yeong Kim[2] and Hwan-Gue Cho[2]
[1] Center for U-Port IT Research and Education, Pusan National University
Republic of Korea
[e-mail: jwkim73@pusan.ac.kr]
[2] Dept. of Computer Science, Pusan National University
Busan, 609-735, Republic of Korea
[e-mail: shji@pusan.ac.kr, s.y.kim@pusan.ac.kr, hgcho@pusan.ac.kr]
*Corresponding author: Hwan-Gue Cho

Received November 23, 2010; revised January 1, 2011; accepted January 26, 2011; published February 28, 2011

Abstract

Internet chat programs and instant messaging services are becoming increasingly popular among Internet users. One of the crucial issues with Internet chat is how to manage the corresponding pairs of questions and answers in a sequence of conversations. Although many novel methodologies have been introduced to cope with this problem, most are poor in managing interruptions, organizing turn-taking, and conveying comprehension. The Internet environment is recently evolving into a 3D environment, but the problems with managing chat dialogues with the standard 2D text-based chat have remained. Therefore, we propose a more realistic communication model for chat agents in 3D virtual space in this paper. First, we propose a new method to measure the capacity of communication between chat agents and a novel visualization method to depict the hierarchical structure of chat dialogues. In addition, we are concerned with communication networks for virtual people (avatars) living in virtual worlds. In this paper we consider a microscopic aspect of a social network in a relatively short period of time. Our experiments show that our model is highly effective in a virtual chat environment, and the communication network based on our model greatly facilitates investigation of a very large and complicated communication network.

Keywords: Virtual Reality, chat program, communication, social networks

A preliminary version of this paper appeared in ACM SAC 2010, March 22-26, Sierre, Switzerland. This version includes a communication network model in virtual world and experiments. This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2010-371-B00008).

DOI: 10.3837/tiis.2011.02.003

1. Introduction

Chat programs and instant messaging services are becoming increasingly popular among Internet users. So a number of chat systems have been released that integrate 2D and 3D graphical representations with standard chat [1]. These include commercial on-line service providers such as 'Second Life'[1], and non-commercial messaging services such as 'MSN', which provide a myriad of chat rooms that are used daily by millions of people [1]. In the beginning, most chat systems were text-based, but real conversation is generally face-to-face. Text chats lack nonverbal cues that facilitate face-to-face conversations, such as gestures, physical distance and direction of gaze [2]. However, informal chat-room conversations are intrinsically different, because of the many noises, dynamic expressions, and the interleaving nature of the discussions. So, it is crucial to combine different chat topics in a social network via chat data mining [3].

In recent years, many 2D graphical chat systems such as 'Comic Chat' [4] and 3D systems such as 'Second Life' have been developed. These 3D communication systems utilize avatars to convey social presence and identity. The avatar is a picture of a person or an animal which represents a person, on a computer screen, especially in a computer game or chat room. In this paper, an avatar means a 3D image which represents a person in a 3D chat system. However, a crucial issue in Internet chat is how to manage the corresponding pairs of questions and answers in the complicated sequence of conversation texts. Text-based chat systems did not have a management method for chat dialogues. This tends to result in confusing exchanges of short messages that are ambiguously ordered, which makes chat a poor decision-making tool and reduces its value for meetings involving presentations of detailed ideas, such as brainstorming [1].

Chat is not truly synchronous, because it has a sporadic rhythm in which fully formed turns pop out in a single moment, instead of in real time [1]. So in the current chat system, a reply delay is inevitable. This is a result of typing difficulties or of someone leaving the chat room. This chat delay could be misinterpreted as an unfavorable response. A typical problem with text-based chat is shown in **Table 1**. In these dialogues, we cannot identify which answer corresponds to a given question. Thus there are two possible answers ("Yes", "I don't know") to a given question. So each chat user is required to comprehend the semantic meaning of all chat dialogues and needs to scroll up to find the prior turn of a candidate. This is a common drawback in all current 2D chat systems.

Table 1. A typical example of delay chat. We cannot distinguish pairs of questions and answers.

Agent	Q/A	Dialogue Text	Time
A	Q1	1 + 1 =2?	t1
B	Q2	Chicken is plant?	t2
C	A1	Yes.	t3
D	A2	I don't know	t4

In this paper, we propose a new 3D virtual communication system to provide a more realistic environment which is similar to a face-to-face communication in the real word. We focus on

[1] http://secondlife.com/

communication networks for virtual people (avatars) living in virtual worlds. The contributions of our work are as follows:

- We propose a new method to measure the capacity of communication among chat agents. Our approach considers the loudness of chatting text (the size of the word balloon) and 3D spatial information in virtual spaces in a 3D viewing volume. For this purpose, we also newly define the Degree of Conversation Strength (DCS) as a measure to quantify the capacity of communication.
- We propose an elegant algorithm to cluster a set of text dialogues by considering the spatial information about virtual chat agents. The basic idea is to find the minimum edge cut of the Chat Flow Graph (CFG) that enables the explicit designation of conversation dialogue pairs based on the spatial information of virtual space. We also propose a novel visualization method to depict the hierarchical structure of chat dialogues by mapping the partitions of the CFG graph into the nested rectangle structures. This method enables us to search a small set of highly related, conversations (chat topics).
- We propose a new communication network model to provide a clear relationship between a pair of agents. Our model considers the microscopic aspect of a social network in a relatively short period of time by using the DCS values in constructing a social network.

This paper is organized as follows. Section 2 introduces related work on communication models in virtual space. Section 3 explains the basic principles of our model with new definitions. We describe how to make a conversation pairing using the spatial information of 3D avatars. Based on this step we can construct a novel chat flow graph model and its visualization in order to achieve a semantic clustering of conversation dialogues. Section 4 presents our experimental results, and we provide conclusion and plans for future work in Section 5.

2. Related Work

Many researchers have tried to resolve the delay chat problem to manage corresponding pairs of questions and answers. Viegas and Donath proposed the novel model Chat Circles[5]. Chat Circles presents each user as a colored circle that expands with an increase in the amount of text entered by the user. The circles then slowly shrink as the text fades. The timing of turns is thus visible, and turns-in-progress are presented as and expanding circle. However, this view of the conversation lacks a historical component, as turns evaporate over time. As a result, the application has an alternative historical view, which visualizes the conversation along a vertical time-line cross marked with lines indicating the timing and size of each user's turn. M. Smith [1] newly introduced an elegant chat model to cope with the delay chat problem with 2D chat. In their model, they proposed an efficient data structure and a novel interface for resolving the ambiguity among turns and replies pairing. This helps us to find a conversation history easily, which is the context chat users want. All of the small disadvantages of their system arise from manual interruption to designate the appropriate position of tree. This was clearly exposed in ample experiments. **Fig. 1** shows a snapshot of Smith's threaded chat program. This type of research indicates one solution to these problems [1].

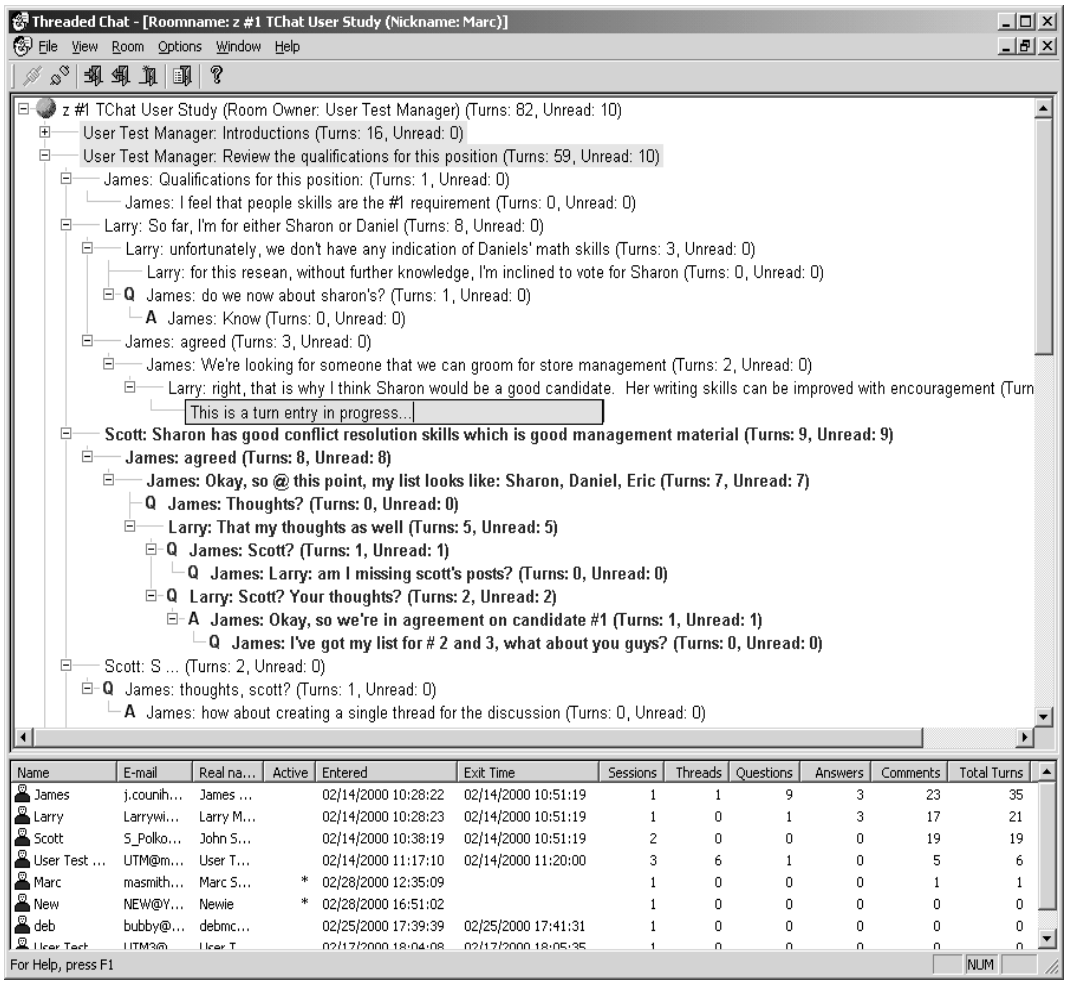


Fig. 1. Snapshot of threadedchat [1].

The current Internet environment is evolving into a de-facto cyberspace or virtual space facilitating communication, business, and entertainment on a global scale [6]. Owing to this change, the chat environment is also evolving into a 3D platform. Many 3D virtual spaces provide communication services, such as Second Life and IMVU[2]. **Fig. 2** shows a current 3D virtual communication system. Such a chat service provides splendid graphics and interesting contents. But, despite the fact that current 3D chat systems have improved features such as rendering of avatars and 3D spaces, in terms of the management and visualization techniques for chat dialogues they are similar to previous plain text-based chat systems. Innovations in chat have mostly ignored the problem of dialogue management. 3D communication systems provide 3D space. Smith [2] analyzed the behavior of chat users in 3D space using V-Chat [7], and others proposed a method for sharing knowledge in 3D virtual space among 3D avatar agents [8]. To the best of our knowledge, most current 2D and 3D chat communication systems do not exploit the truly valuable 3D spatial information, which includes the avatars' physical locations and the physical distances between them, and the avatars' directions of gaze. We insist that this spatial information is crucial for solving all of the problems with previous chat systems. In a real conversation we can avoid confusion, since we can determine the

[2] http://www.imvu.com/

corresponding answer to a given question by perceiving the direction of gaze, the distances between the participants, and the conversation semantics.

(a) IMUV **(b)** Second Life

Fig. 2. Examples of current 3D virtual communication systems

Another problem of virtual communication systems is supporting social networking. Recently, with increasing social activities on the Internet, it becomes important to provide social network services. Social networking is built on the idea that there is a determinable structure to how people know each other, whether directly or indirectly [9]. Currently, a social network is constructed by adding various social objects. A social object is a connection between two people. For example, Flickr[3]'s social object is a picture and Amazon[4]'s is a book. One of the best social objects is communication. Previously, many social network construction methods that have been released involve communication [3][10][11]. Khan [12] proposed a generalized framework to understand a social network (community). This is a sound method for a social network in a community site. Orgnet.com[5] provides software for social network analysis (SNA) involving communication. **Fig. 3** shows Orgnet's software for social network analysis. Dana [13] proposed SNA in virtual environments. They construct a social network using a chat log in Multi-User Virtual Environments (MUVEs) and analyzed it. However, these systems constructed a social network using only a chat log. But most social network services require explicit user interaction. Some social network services automatically construct networks. But such construction methods provide uncertain relationships between users. In this paper, we present a social network used by people who are really acquainted (communication), and extended social networks such as affiliation networks, in our system.

3. Our Communication Network Model

In this paper, we propose a more realistic 3D virtual communication network model with data structures and algorithms that support solutions to the problems with the previous 2D and 3D conversation systems, by exploiting the 3D spatial information. **Fig. 4** shows snapshots of

[3] http://www.flickr.com/
[4] http://www.amazon.com/
[5] http://www.orgnet.com

Project Wonderland which included our 3D communication methods and algorithms. The summary in **Table 2** enables a comparison of our model with the other chat models.

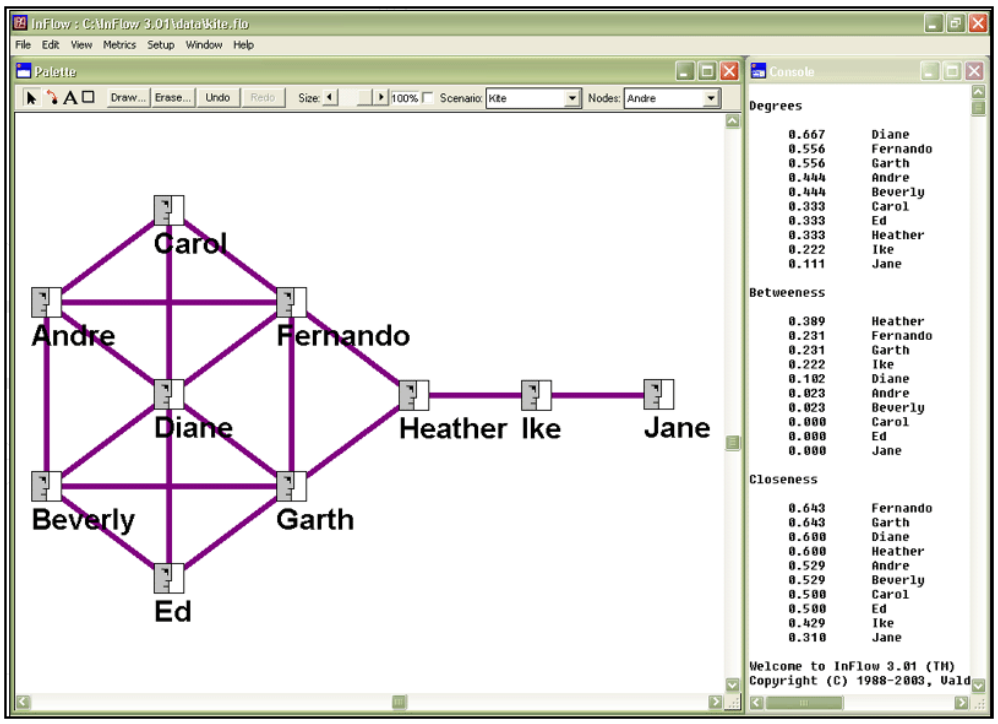


Fig. 3. Snapshot of software for social network analysis. Nodes and edges denote people and determined chat logs, respectively.

Fig. 4. A snapshot of our 3D-Chat System. The word balloon of agent at a long distance is too small to read, as it is hard to hear the sound of a person at a long distance.

Table 2. Supporting function for each chat tool. X = No support. Δ = Partial support . O = Full Support.

	MSN	Second Life	Wonderland (our added model)
System base	Plain text	3D	3D
avatar	X	O	O
emotion	Δ	O	O
voice	X	Δ	O
spatial info	X	Δ	O
manage dialogue	X	X	O
social interaction	X	O	O

3.1 Chat Model with 3D Information

To provide a more realistic communication model, the chat system needs to express the relationship between the conversations of agents. However, most 3D chat systems such as Second Life and IMVU have not considered the relationship between agents since they mainly use 2D-stylized word balloons of the same size and manage chat dialogues regardless of all 3D spatial information. Therefore, we propose a method to obtain the relationship between the conversations of agents in this paper. The problem to obtain the relationship is to find how to measure the capacity of communication between chat agents. We believe 3D virtual space systems can obtain the communication capacity from the information of the distance between two agents (avatars) and their gaze as in real-life chatting. For example, if an agent is near another agent with eye contact, the possibility of communication between two agents is very high. In this section, we present our two approaches to obtain communication capacity – *Virtual Chat Bandwidth (VCB)* and *Degree of Conversation Strength (DCS)*. The basic principle of our approaches is exploiting the 3D spatial information in virtual spaces.

Our first approach to obtain the communication capacity between chatting avatars is *VCB* which is defined in our previous work [14]. **Fig. 5-(a)** shows the concept and examples of *VCB*, and Definition 1 shows how to compute *VCB*. *VCB* is related to $dist(A_a,A_b)$ and $(\sin\theta_a \cdot \sin\theta_b)$. The $dist(A_a,A_b)$ is the distance between agent A_a and agent A_b. $(\sin\theta_a \cdot \sin\theta_b)$ is the scalar product of the visible regions for A_a to A_b and A_b to A_a. If $\theta_a, \theta_b = \pi/2$ as shown in **Fig. 5-(b)**, $(\sin\theta_a \cdot \sin\theta_b)=1$, and when two agents are precisely face to face, as shown in **Fig. 5-(c)**, $\sin\theta_a \cdot \sin\theta_b$ is maximized ($\sin\theta_a \cdot \sin\theta_b=1$).

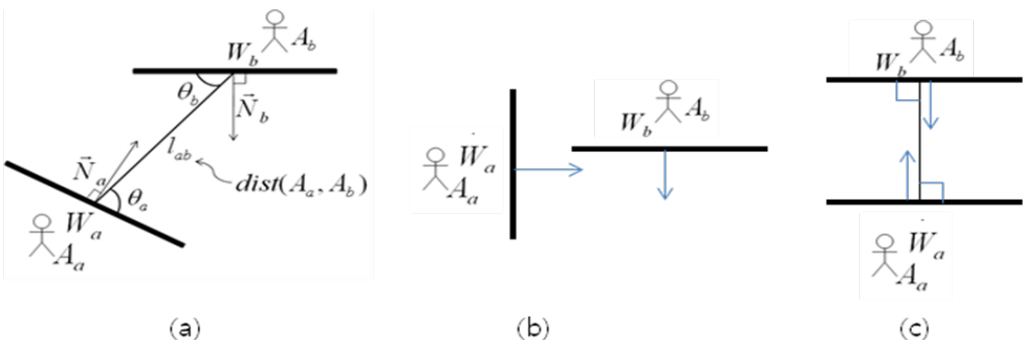


Fig. 5. Avatars A_a and A_b are chatting. The solid line segment denotes the width of the word balloons W_a and W_b appearing on each head to show the chat texts.

Definition 1 *For two chat agents A_a and A_b, the VCB is the communication capacity bandwidth, which is defined as follows. If* $\overrightarrow{N_a} \cdot \overrightarrow{N_b} \geq 0$, *then* $VCB(A_a,A_b) = 0$. *Otherwise let*

$$VCB(A_a, A_b) = \frac{(\sin\theta_a \cdot \sin\theta_b)^{k1}}{(dist(A_a, A_b) + C2)^{k2}} \qquad (0 < \theta_a < \pi, 0 < \theta_b < \pi)$$

, where C_1, C_2, k_1, and k_2 are control constants.

Determining control constants is important when simulating a real situation. In this paper, we found these constants empirically with several experiments. Since the communication ability of each chat agent differs, it is not easy to give a general rule to determine the control constants. In this paper, when all agents are distributed uniformly in the square region, we determined the control constants to make sure that the number of chat pairs does not exceed N edges, where N is the number of agents. If they are N community edges, then the number of chat pairs is about O(N/2) groups, which is an average real-world situation.

However, *VCB* is not sufficient to provide communication capacity. In reality, the audible distance for a chat agent is dependent on the loudness of the speaker. So, if the speaker shouts, his or her voice will be heard by more people than if he/she whispers to a few closer people. To measure audible distance, we defined the *Loudness of Chatting Text (LCT)* in our previous work [14]. It measures the emphasis of dialog d_i as follows. If $length(d_i) = 0$, then $LCT(d_i) = 0$. Otherwise we have

$$LCT(d_i) = \frac{No.of\ Exclamation(d_i) + No.of\ CapitalWord(d_i)}{Total\ No.of\ Words(d_i)} \tag{1}$$

, where d_i is the *i-th* dialogue text. So the *LCT(a)* can be regarded as the loudness of an agent's dialogue *d*.

LCT is determined by the number of exclamation marks, $Exclamation(d_i)$ and the number of capitalized words, $Capital(d_i)$. *LCT* is applied in order to zoom in the scale of the word balloon's text. **Fig. 6** shows different size balloons according to the Loudness of Chat Text *LCT(d)* for chat dialogue *d*.

Fig. 6. The size of each word balloon is depends on the *LCT*.
(a) *LCT = 0*, **(b)** *LCT = 0.22*, **(c)** *LCT = 1.0*.

In this paper, we propose a new chat model applying the *LCT* and *DCS* that is an approach to obtain the communication capacity with *LCT*. The main principle of our model is considering the size of the word balloon in the frustum of the agent. **Fig. 7** shows the concept of our model. If three same word balloons (A_a, A_b, and A_c) exist in the view volume of an avatar in 3D space as shown in **Fig. 7**, the sizes of the word balloons are different from each other. **Fig. 8** shows the size of word balloons according to distance between an avatar and his/her conversation partner. In **Fig. 9**, let *wb(a, t)* denote the word balloon of agent *a* at time *t*. Let avatar agent *b*

gaze in direction *direct(b, t)* at time t., where *direct(b, t)* is a vector. So we construct a 3D viewing volume for *b* at time *t*, *viewvol(b, t)*. Therefore, let *pyramid(b,t)* be defined as the viewing volume to face the physical location of *b* in the direction of the gaze vector *direct(b,t)* at time *t*. Suppose that avatar *a* chats to avatar *b* at time *t*. Then *wb(a, t)* is partially included in *pyramid(b, t)*. The spatial properties of each *pyramid(b, t)* can differ according the avatar's characteristics, such as a wide viewing gaze, a narrow view or a short-sighted one etc. However, we simply assumed that the projection degree of the pyramid is 76 degrees in this paper. So, all of the pyramids in this experiment are equivalent to the quadrangular pyramid. In this case, we say that *b* hears *a*. In practice *wb(a, t)* is clipped by *pyramid(b, t)*. Let *clip(a,b,t)* be defined as the projection of the clipped polygon of *wb(a,t)* by *pyramid(b,t)* at time. Let us define *cutpoint(a, b, t)* as the intersection point between the physical position of the chat avatar *a* and the *pyramid(b, t)*. And, *cutplane(a, b, t)* is defined as the cutting plane which passes *cutpoint(a, b, t)* and is perpendicular to *direct(b, t)*.

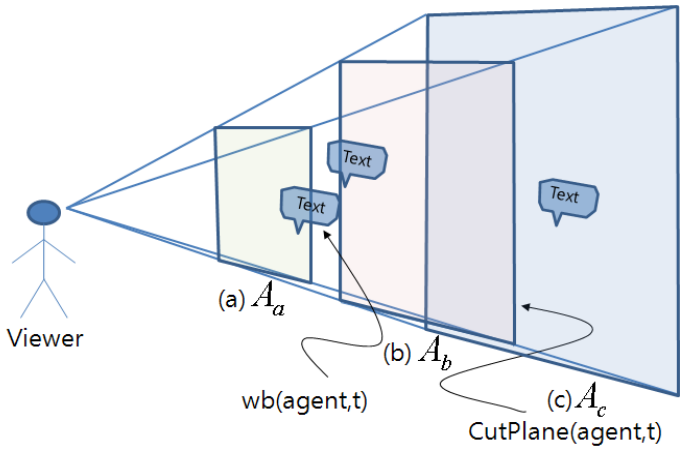


Fig. 7. Three chat agents (*Aa*, *Ab*, *Ac*). Aa is the closest to the listener.

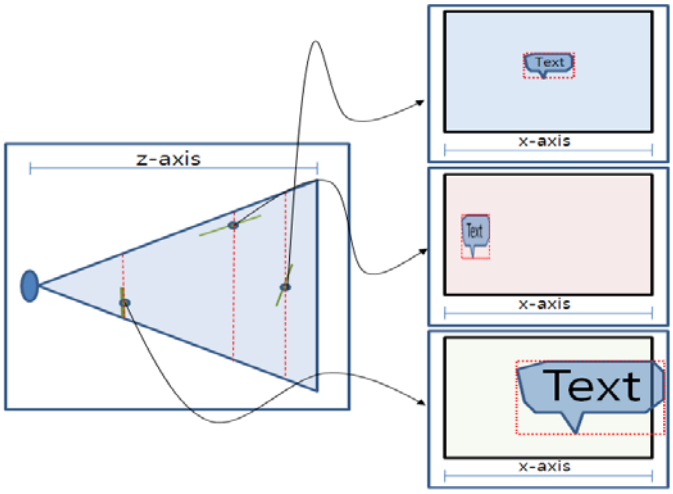


Fig. 8. Size of each word balloon when seen from above the z-axis.

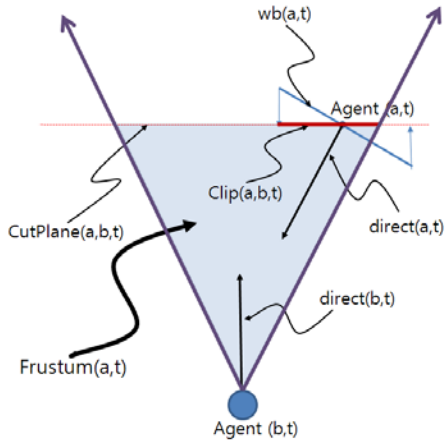


Fig. 9. Avatar *b* watches avatar *a* talking to him. The word balloon of avatar *a* is partially visible to avatar *b*.

Let *DCS(a, b, t)* be the degree of conversation strength for avatar *a* and *b* at time *t*. *DCS(a, b, t)* is the ratio of the area of *clip(a, b, t)* to that of *cutplane(a, b, t)*. Thus, nearer the chat agent *a* is to *b*, the larger the *DCS(a, b, t)*, since the ratio of that of *clip(a, b, t)* to the area of *cutplane(a, b, t)* increases. And, the larger the scalar product of *direct(b, t)* and *direct(a, t)*, the larger the *DCS(a, b, t)*. In other words, *DCS(a, b, t)* will be 1 (optimal), if two very close chat agents engage directly in face-to-face chat. Here we give the formal definition of *DCS(a, b, t)*. Let two avatars, *a* and *b*, chat during time interval *t* (a short period of 10 to 20 seconds).

$$DCS(a,b,t) = \frac{area\ of\ \ clip(a,b,t)}{area\ of\ \ cutplane(a,b,t)} \tag{2}$$

Fig. 10 shows the layout of three different locations of agents with the corresponding *DCS* values. In reality, we cannot chat with an agent who is far away from us, nor can we chat with a close agent in a very quiet voice. Thus if *DCS(a, b, t)* is smaller than a threshold value c_0,, then we can say that *a* has not chatted with *b*. c_0 can be determined empirically.

(a) **(b)** **(c)**

Fig. 10. Example of three different *DCS*s of a given dialogue in our system. **(a)** DCS=36.175. **(b)** DCS=17.214. **(c)** DCS=5.419.

3.2 Chat Flow Graph and Its Visualization

As mentioned in Section 1, a typical problem of both 2D and 3D chat systems is the delay chat problem which makes it difficult for us to distinguish between pairs of questions and answers.

However, to the best of our knowledge, the prevailing chat tools only store text data based on a temporal sequence. In this case, we may not be able to find the corresponding answer to a query question. Therefore, we proposed the *Chat Flow Graph (CFG)* for managing chat dialogues among multiple agents in virtual space [14]. This graph enables explicit designation of conversation dialogue pairs based on the spatial information of virtual space, i.e., the avatars' location, their directions of gaze, and the distances between them. Although the *CFG* could be a good solution for managing a set of chat dialogues, according to the features of the graph, if there are multiple nodes then the *CFG* is difficult to understand. So we need a reasonable visualization system that makes it easier for users to understand the chat history [15][16][17]. In this paper, we propose a graph-based chat clustering and visualization approach to manage chat dialogues among multiple agents in virtual space.

To cluster chat dialogues, we propose a new hierarchical graph decomposition method for CFG. If we want to cluster the set of all dialogues into a meaningful chat group (a conversation about a specific topic), we need to decompose a *CFG* into sub-groups, since a meaningful chat group usually forms a sub-group in *CFG*. In *CFG*, a group of chat topics is represented by a rectangle. So the entire *CFG* is successively partitioned into sub-rectangles via recursion. Algorithm 1 explains how to apply a graph-cut for the *CFG*. The basic principle of the *CFG* cut is the minimum edge cut needed to find a denser sub-graph partition, where *mindepth* is a control constant, and *checkHcut()* and *checkVcut()* are responsible for finding the vertical/horizontal cut. After finding the minimal graph cut, they are separated via functions *Hcut()*(horizontal-cut) and *Vcut()*(vertical cut). The graph decomposition continues until the size of the divided sub-graph is greater than a threshold value, which is determined empirically. In our procedure we prefer the vertical cut to the horizontal one. In order to map these graph partitions to the nested rectangle structures, whenever we obtain two partitioned sub-graphs, we assign them to a rectangle, the size of which is dependent on the number of nodes, which means the number of dialogues of the chat avatars. Next this procedure is recursively applied until we have obtained a sufficiently small graph cluster (topic chat). **Fig. 12** shows an example of a *CFG* showing a conversation on planning a short trip at an elementary school.

Algorithm 1 Graph-cut Algorithm

Input: *CFG(Chat Flow Graph)*
Output: Set of divided Graph

```
procedure CFGRAPHCUT(Gs, mindepth)
    if checkHcut(Gs) = TRUE
        HGs ← do Hcut(Gs);
        CFGRAPHCUT(HGs, mindepth);
    else if checkVcut(Gs, mindepth) = TRUE
        VGs ← do Vcut(Gs, depth);
        CFGRAPHCUT(VGs, mindepth);
    else return
end procedure
```

Algorithm 1 does not perfectly guarantee that the final returned groups are topically coherent, but our approach provides a useful solution for topic-based clustering. The CFG could be well-structured in terms of topical coherence in practice, since the CFG is created based on the DCS that is a measure of conversation strength considering the loudness of chatting text in virtual space. So our approach can provide reasonable accuracy on topic-based clustering, though the algorithm analyzes dialogues only via structural analysis. As far as we know, it is very hard to analyze the dialogue topic with the semantic and syntactic analysis. So our approach could be a good and alternative model for decomposing the dialogues in terms of chatting topic. If a more concrete semantic and syntactic analysis (based on linguistic study) could be applied on our topic-based clustering, then the final result could be improved more.

That means our clustering module could be used a good preprocessing step in analyzing chat dialogues.

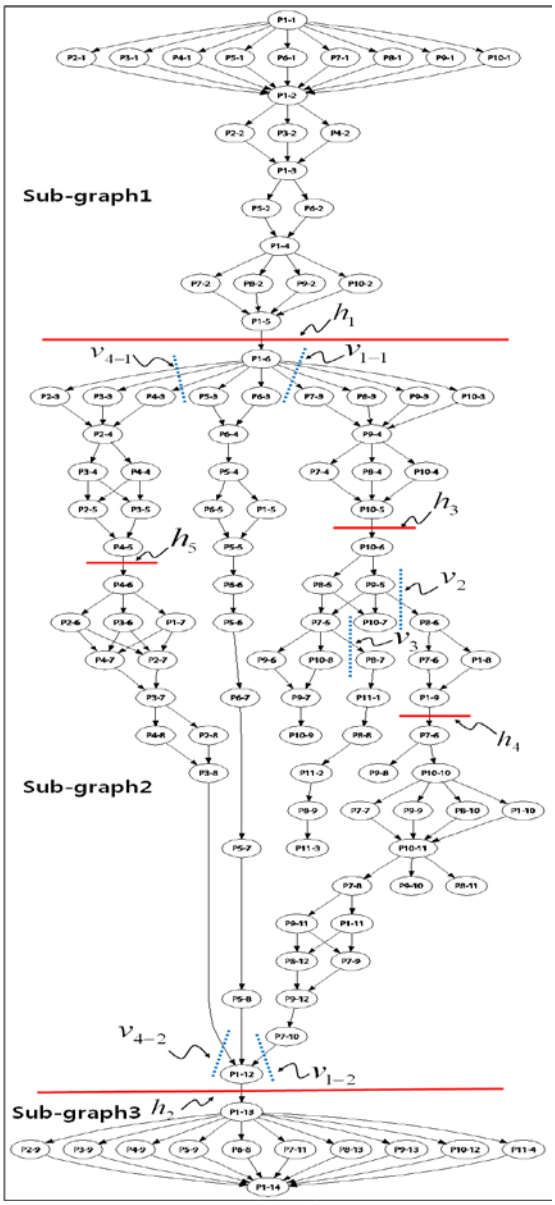


Fig. 12. *CFG* showing a conversation on planning a short trip at an elementary school. The conversation includes one *teacher(P1)* and ten *students(P2-P11)*. First, the teacher explains the trip plan (date and place) to all of the students. Then, the students are divided into three small groups and each group discusses its own picnic plan. Finally, the P1−14 node shows that all students agree on the picnic plan. Enlarge images of sub-graph1 and sub-graph3 is shown in **Fig. 13**, and sub-graph2 is shown in **Fig. 14**.

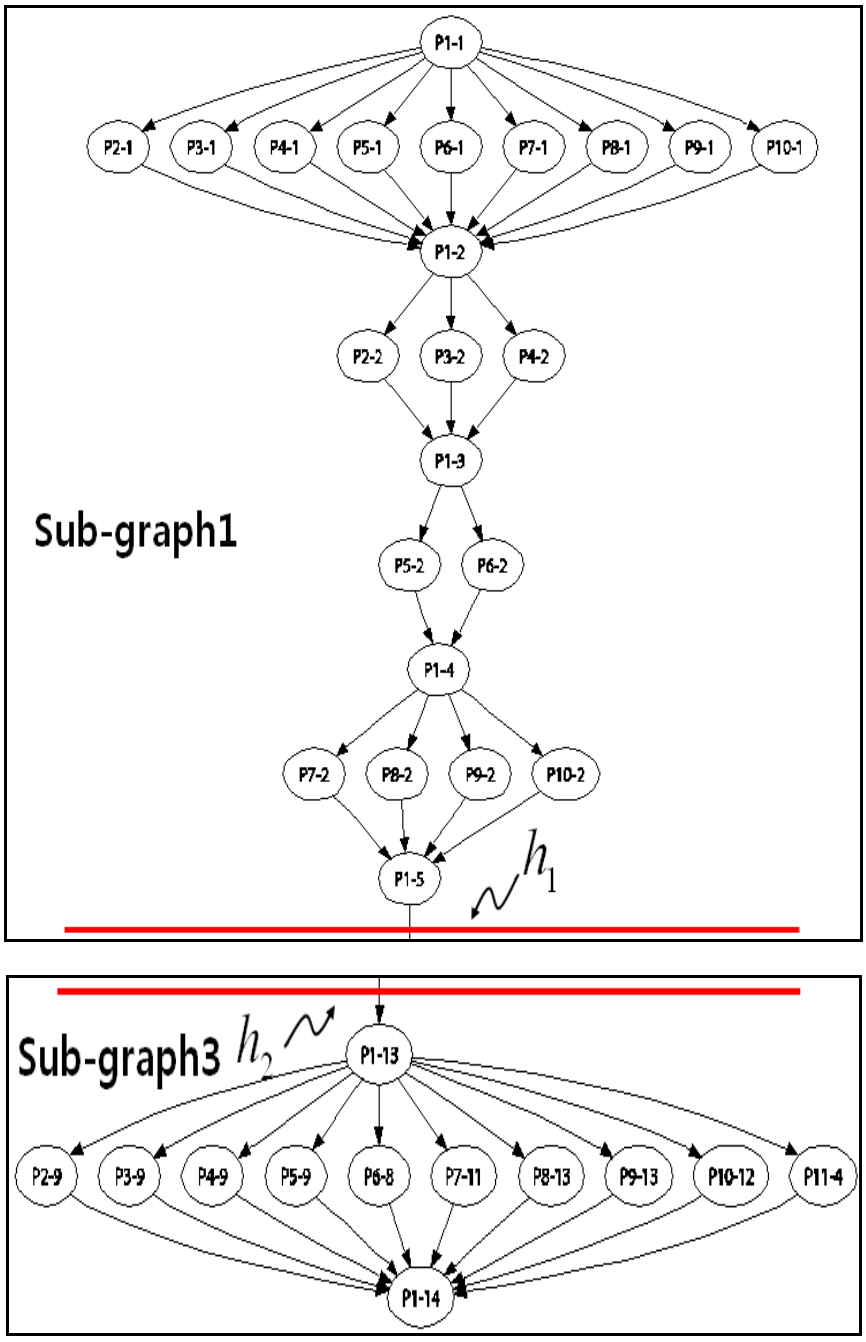


Fig. 13. The sub-graph1 and sub-graph3 of the *CFG* in **Fig. 12**.

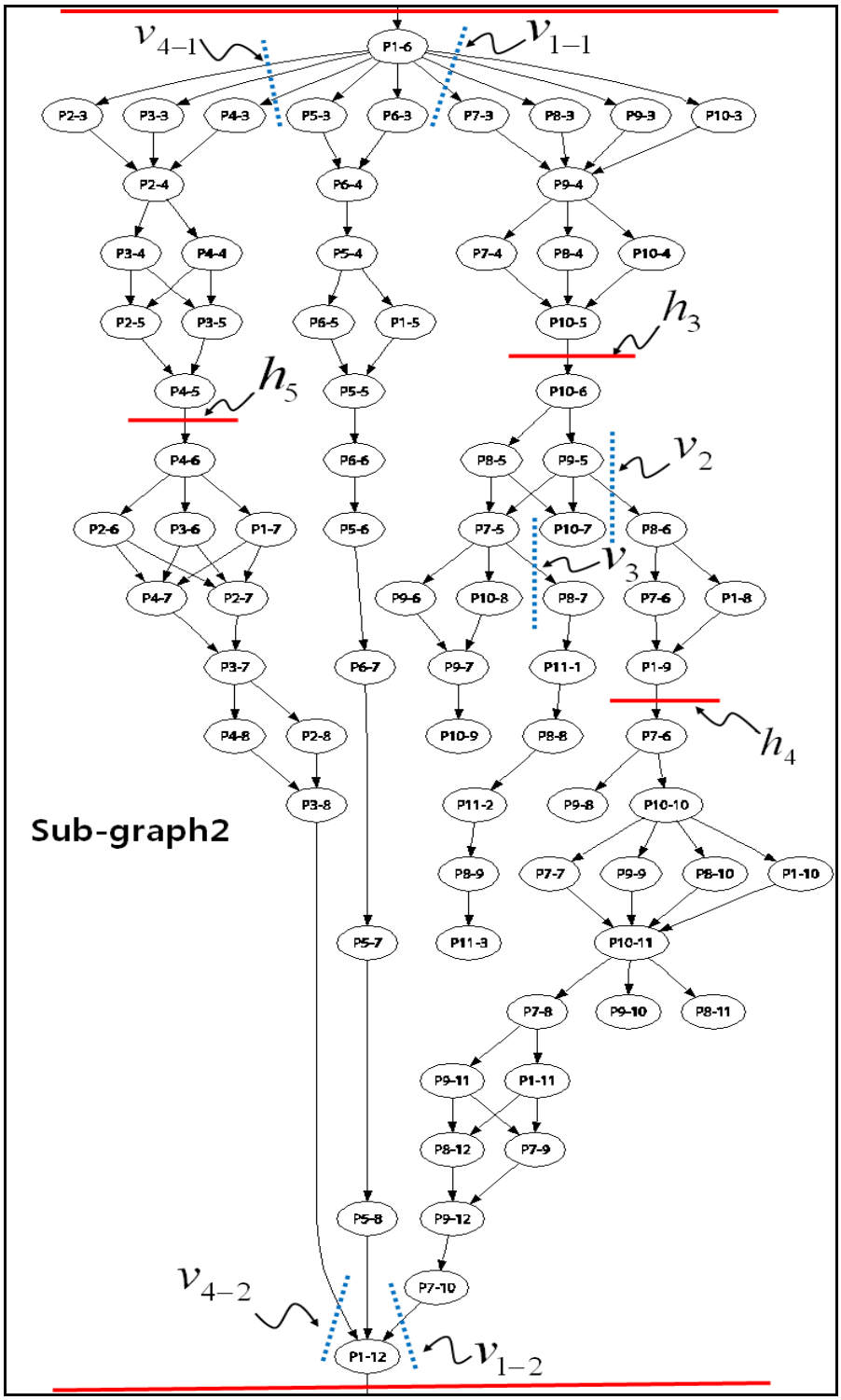


Fig. 14. The sub-graph2 of the *CFG* in Fig. 12.

Fig. 15 shows a new visualization method we propose in this paper. **Fig. 15-(a)** and **(b)** show the visualization result from **Fig. 12**. In **Fig. 15**, the colors of the recursively divided

rectangles enable us to distinguish between them. In **Fig. 15-(b)**, a bigger sub-rectangle means that it includes a lot of dialogue and it is more important to the whole conversation. Also we provide an interactive user interface that enables a review of the set of highly related chat texts of all dialogues via a simple mouse click. If we point to one of the divided unit sub-rectangles, then it immediately represents the chat dialogues as a directed acyclic graph, as with the *CFG*. So we can read the set of related dialogues assigned to a rectangle. Also, by controlling the viewing scale of rectangle visualization, we can see a wider view of the chat dialogues.

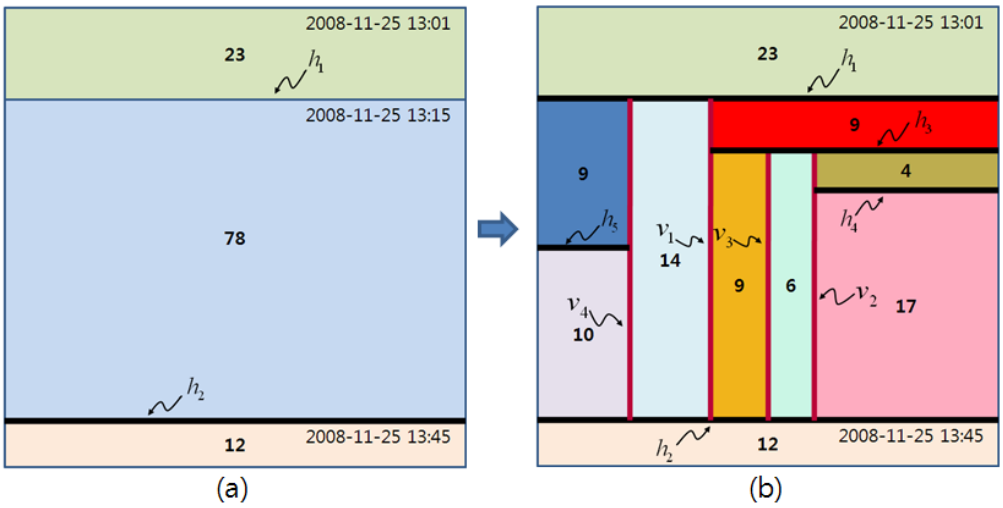


Fig. 15. **(a)** Visualization of CFG in Figure 12. The threshold parameter *mindepth = 1*. The number in each sub-rectangle denotes the number of dialogue nodes included.
(b) The threshold parameter *mindepth = 3*.

3.3 Communication Network Model

Social network construction methods have long been studied. In reality, communication-based social networks are composed via actual interaction (communication). However, current virtual communication systems create a social network of agents in the same chat room, regardless of actual interaction among agents. This results in uncertain relationships among agents and decreases reliability. In the chat system, it is necessary to provide the relationships among some participants in a short time. For example, sometimes we want to know the dynamic relationships among some participants in the same chat room. In this paper we propose a Communication Network Model considering a microscopic aspect of a global social network. The microscopic aspect means a sort of virtual interaction which depends on the facing degree and the talking distance during a relatively short period (a few minutes or hours). Our model uses DCS values presented in Section 3.1 to reflect the microscopic aspect. Our model also provides extension and simplification of a social network. Our model is extended by adding the relationships between the agents and objects such as places and things of interest, and simplified with internally complete sub-graphs. These characteristics of our model provide a clear relationship between a pair of agents that have actually talked, just as with a social network based on a realistic conversation.

First, we create an *Acquaintance Social Network (ASN)*. An Acquaintance Social Network is a social network constructed using DCS. **Fig. 16** shows an example of an *ASN*. Nodes indicate agents, and edges are determined via the *DCS*s of the conversations between them. The *edgeweight* denotes the sum of the *DCS*s in all dialogues between a pair of agents, and if it

is above the threshold parameter we connect the pair of nodes. The thickness of the edges indicates the strength of the *edgeweight*.

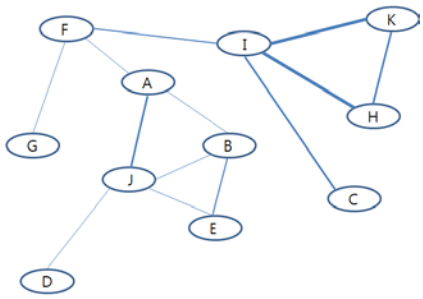


Fig. 16. Acquaintance Social Network. Nodes denote agents. Edges are determined via *DCS*s.

Next, we create a *General Social Network (GSN)* by adding the relationships between the agents and objects to an *ASN*. The major target of social networks is people, but we need to know the interest objects of the people in society in order to apply *Social Network Analysis* (*SNA*). To find the relationships between the objects and agents, a *GSN* is composed of the interactions between the agents and objects. If you touch a given object in a virtual world, you become related to it. Based on this type of interaction, we can create a *GSN*. **Fig. 17** shows an *agent-object GSN*. We then merge the Acquaintance Social Network and the General Social Network. If an edge overlaps during merging, we redraw the edge in bold to signify a stronger relationship between a pair of agents. But the merged graph is very complex, so we need to use a simplification method for easy identification. Let a complete society be a society in which all the members are related. If a small complete society exists in a large non-complete society then we consider it to be an internally complete society and we can make a single group of it. But in a general society, this is a difficult case, so we need to extend our social graph in order to make a complete relationship. If all of an agent's interest objects are the same as those of another one, we consider there to be a potential relationship between the pair of agents and so we extend the edges between this pair. There may be no obvious relationship between the objects, but we consider that if all of the members of a complete society have an interest in the given objects, then there is in fact a relationship between them, so we create augmented edges between the object pair. This is called an *Extended GSN*.

Definition 2 Let Gg(Vg,Eg) be a GSN with a set of vertices Vg and a set of edges Eg. The notations to be used in the definition are defined as follows:

- *A set of vertices, Vg = Va U Vo. Va is a set of agents, and Vo is a set of object.*
- *A set of edges, Eg = Ea U Ea,o U Eo. Ea is a set of relationship between agents, and Ea,o is a set of relationship between agents and objects. Eo which is a set of relationships between objects is ø, since GSN does not have relationship between objects.*

For example, in **Fig. 18**, $V_a = \{a_1, a_2, a_3, a_4\}$, $V_o = \{o_1, o_2, o_3, o_4, o_5\}$, $E_a = \{\{a_1, a_2\},\{a_2, a_3\},\{a_2, a_4\}\}$, $E_{a,o} = \{\{a_1, o_1\},\{a_1, o_2\},...\{a_4, o_5\}\}$.

Definition 3 *We define an Extended GSN for a GSN. For $G_e(V_e,E_e)$, $V_e \equiv V_g$ we augment the edges in E_a and $E_{a,o}$ according to its topological connections.*

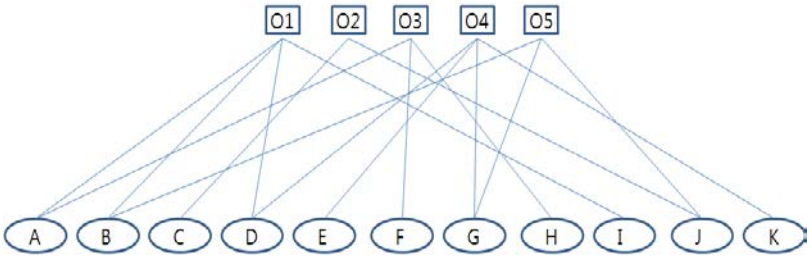


Fig. 17. General Social Network with 11 agents and 5 objects.

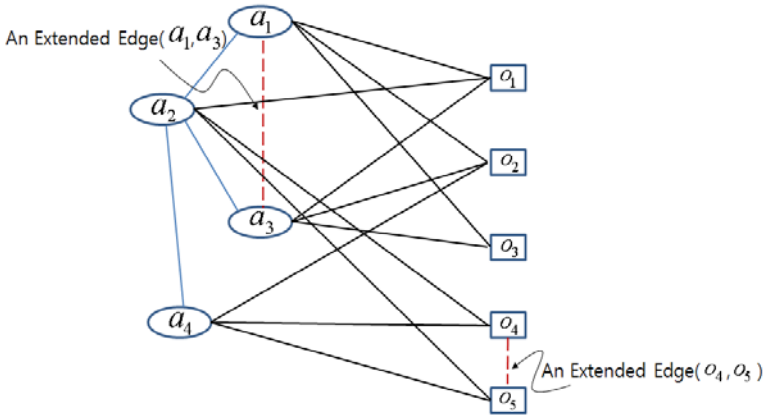


Fig. 18. Agent and Object augmented edges. If all of the interest objects of a pair of agents are the same, then we connect the edges. And if the agent graph is a complete graph and all of their interest objects are the same, then we connect the edges between the object pairs.

Definition 4 *Let $No(a_i)$ denote the set of objects o_j , which are adjacent to agent a_i. In a similar manner we can define $Na(o_i)$.*

For example, in **Fig. 18**, $No(a_1) = \{o_1, o_2, o_3\}$, $No(a_4) = \{ o_2, o_4, o_5\}$. Similarly $Na(o_1) = \{a_1, a_2, a_3\}$, $Na(o_4) = \{a_2, a_4\}$. For two avatars a_p and a_q, if $N_o(a_p) \equiv N_o(a_q)$, then we add an extra edge to E_g such that $E_e = E_g \cup (a_p, a_q)$, and we call this type of extra edge an avatar augmented edge. Similarly, we also add some edges to $E_{a,o}$ in the following manner; if $N_a(o_i) \equiv N_a(o_j)$, then we add edges such that $E_e = E_g \cup \{(o_i, o_j)\}$. **Table 3** shows the relations matrix of **Fig. 18**.

Table 3. Matrix of relationships between the agents and objects in **Fig. 15**. The x-axis and y-axis denotes agents and objects, respectively.

	a_1	a_2	a_3	a_4
o_1	✓	✓	✓	
o_2	✓		✓	✓
o_3	✓		✓	
o_4		✓		✓
o_5		✓		✓

Finally, we simplify the Extended GSN by finding an internally complete graph in the whole Communication Network. And then if there is a complex graph, we can make it into a single node. **Fig. 19** shows the simplified Communication Network.

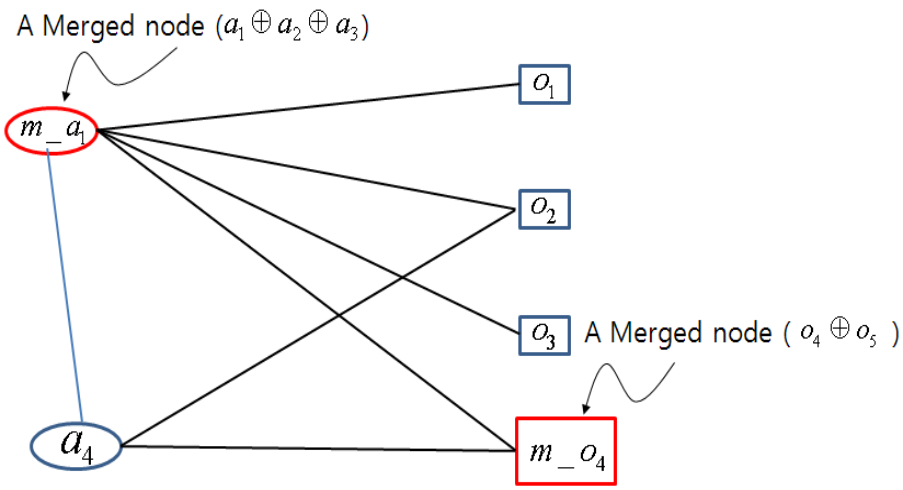


Fig. 19. Simplified Communication Network in **Fig. 17**.

In this paper we considered a microscopic aspect of a social network in a relatively short period. For example sometimes we want to know the dynamic relationships among some participants in a one-day conference or exhibition or a social party program. Rather than the general social network studied previously, these relationships are constructed in a short time (a few hours). Therefore, one contribution of this paper is that it is the first work dealing with the microscopic aspect of a social network in a virtual space. In addition, we can regard a general social network as a macroscopic social network, where the relationships are established over a long time such as several months or even multiple years (alumni). It is crucial to observe the interactions among all the virtual participants to characterize the microscopic aspect of a short-time social network. Since we considered virtual interaction as done in real space (facing degree, the talking distance and the indirect interaction among virtual agents via the virtual object), our model could be applied to track all the interactions precisely compared to other work in virtual communications, such as in plain text-based chat.

4. Experiment

In this section, we show the experimental results of *DCS* obtained by using our system. And we show the construction method of a Extended Social Network and a Communication Network. First, in order to show the usefulness of our system, we assumed a chat situation obtained from a study book for English conversation. Seven agents are moving around in a virtual space, chatting with each other, as shown in **Fig. 20**. **Table 4** shows a part of the chat situation including two topics. In **Table 4**, it is difficult to know what dialogue is the question on the dialogue at time t9. On the other hands, we easily know pairs of questions and answers in our system as shown in **Fig. 21**.

Fig. 20. Snapshots of our system. **(a)** A agent is getting close to locating six agents that take part in the converstation. **(b)** The agent join the conversation. **(c)** Seven agents are splited into two groups. **(d)** Seven get together again to make their farewells.

Table 4. An example of conversation. We cannot distinguish pairs of questions and answers

Agent	Dialogue Text	Time
A	Do you know a way to Bexco?	t1
B	Most of the time, I drive.	t2
D	I have as many as 3 meetings today.	t3
C	Isn't traffic really bad during the rush hour, though?	t4
E	We're behind schedule.	t5
C	Why don't you take the subway instead?	t6
B	The subway is always packed with people during the rush hour.	t7
D	We are pressed for time.	t8
A	I really don't want to deal with it.	t9

Fig. 21. Snapshots of the situation **(c)** in **Fig. 20**. We eaily know pairs of questions and answers in our system through the size and shape of word balloons determined by DCS values.

We provide a simple example to easily demonstrate the usefulness of the *DCS*. **Fig. 22** shows snapshots of our system, and **Table 5** shows the corresponding *DCS* value.

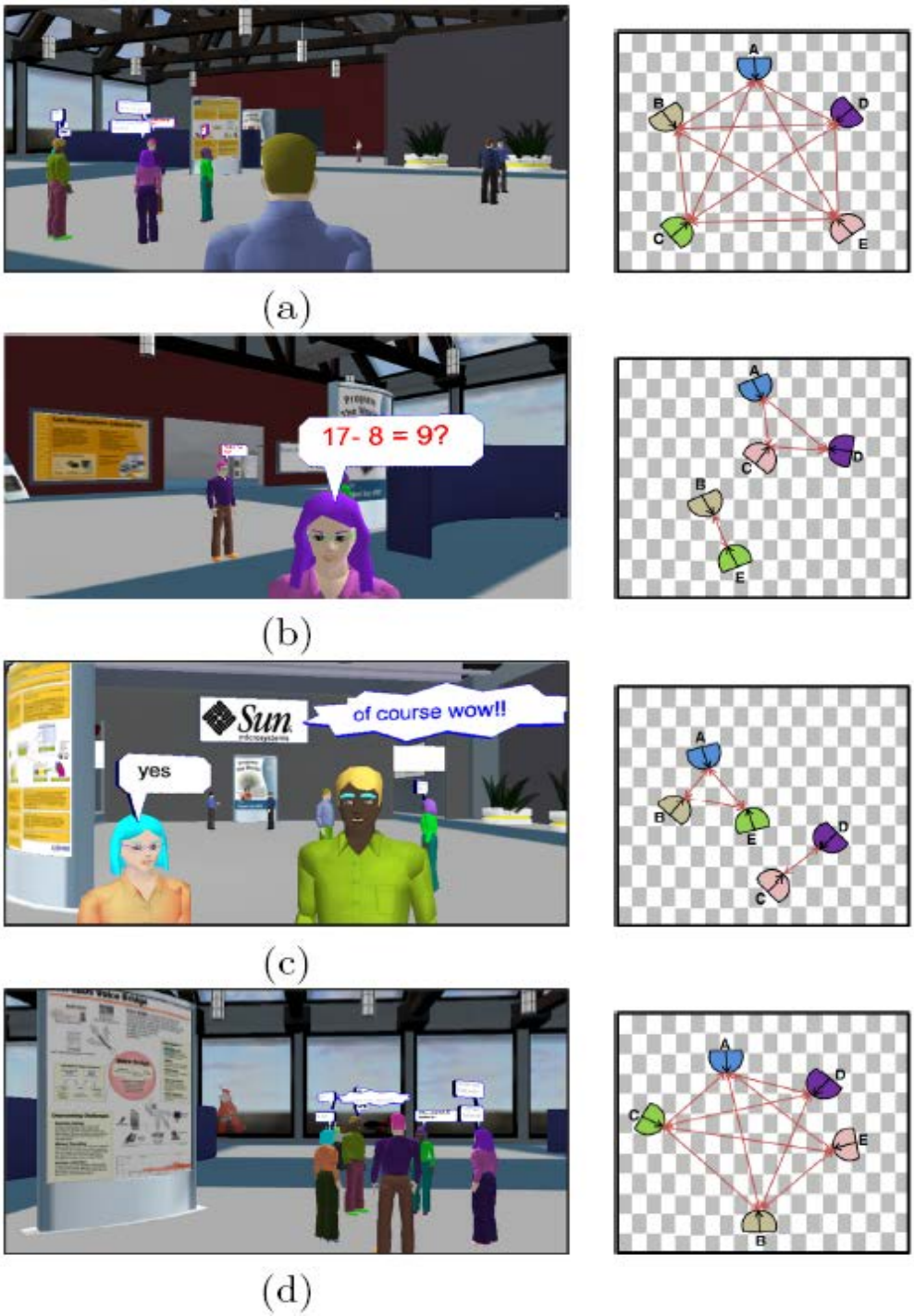


Fig. 22. Snapshot of our system. **(a)** Global view showing the layout of the chat group with five persons. **(b)** View of agent c. **(c)** View of agent a. **(d)** Global view.

Table 5. Four corresponding *DCS(*,*)* matrix of agents appearing in **Fig. 20-(a)**, **(b)**, **(c)**, and **(d)**.

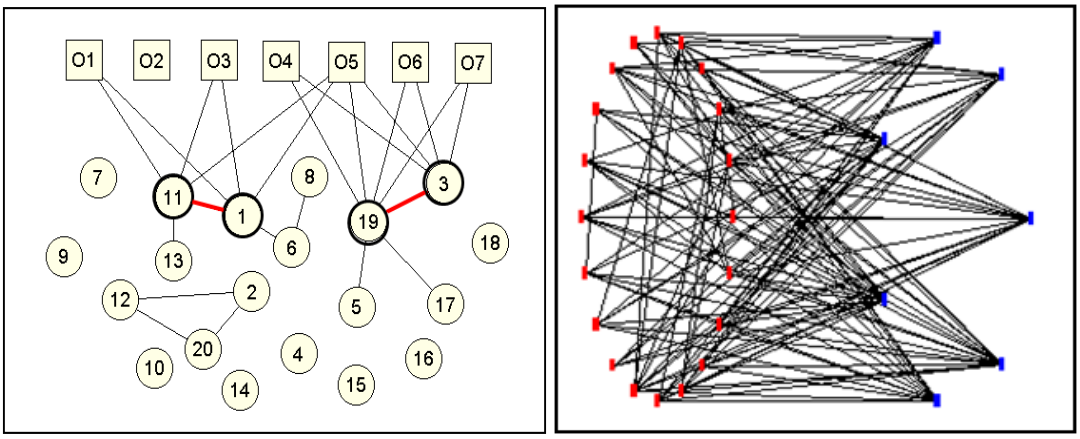
(a)	A	B	C	D	E
A	0.00	9.95	12.54	10.76	9.66
B	16.85	0.00	9.94	11.94	14.34
C	12.66	8.81	0.00	14.47	14.66
D	10.72	11.69	11.94	0.00	12.24
E	11.65	15.27	10.25	13.69	0.00

(b)	A	B	C	D	E
A	0.00	0.00	13.45	17.54	0.00
B	0.00	0.00	0.00	0.00	21.65
C	16.87	0.00	0.00	14.56	0.00
D	14.56	0.00	13.54	0.00	0.00
E	0.00	20.45	0.00	0.00	0.00

(c)	A	B	C	D	E
A	0.00	19.44	0.00	0.00	23.65
B	20.24	0.00	0.00	0.00	7.36
C	0.00	0.00	0.00	24.95	0.00
D	0.00	0.00	25.53	0.00	0.00
E	24.57	9.35	0.00	0.00	0.00

(d)	A	B	C	D	E
A	0.00	13.69	12.58	11.13	12.85
B	14.13	0.00	13.55	12.32	16.21
C	11.82	9.54	0.00	11.57	11.71
D	12.65	10.61	12.22	0.00	13.54
E	10.46	11.54	16.54	9.65	0.00

Second, we show our Communication Network. We made an artificial communication network by using a random graph, as shown in **Fig. 23-(a)**. *N* and *E* denote the number of nodes and edges, respectively. Next, we added the relations of the objects in the acquaintance communication network. **Fig. 23-(b)** shows the social network that includes the object relations. Because the number of edges is much higher it is complex.



(a) **(b)**

Fig. 23. **(a)** Acquaintance Communication Network. circle-node and rect-node denote agents and objects. The two red edges will be augmented by the indirect interactions between agents and objects, where, *N(a11),N(a1) = {O1,O3,O5}* and *N(a3),N(a19) = {O4,O5,O6,O7}*. **(b)** Communication Network that includes all the relationships between the agents and objects in **Fig. 19**. The dots on the left hand side (the red dots) denote the avatar agents, and the dots on the right hand side (the blue dots) denote the objects, where *N=27, E=115*.

The augmented communication network was constructed by using our augment method, as shown in **Fig. 24-(a)**. 17 edges were augmented in the initial acquaintance communication network. Finally, we found the cliques in the augmented communication network and then we set the threshold clique size=3 in this experiment. So, if we found that there were more than 3-cliques, we made a grouping into a single node. **Fig. 24-(b)** shows the results of simplification. Using the same method, the number of object nodes was also reduced. **Table 6**

shows the reduced rate of our communication network by using various random graphs. To summarize, first we created a social network that included the relations of the interest objects. We could reduce the nodes connected to each node. Also, we can show the hierarchical social structure via the clique size. The more agents and objects there are in the initial communication network, the smaller the size of the generalized social network obtained, as shown in **Table 7**.

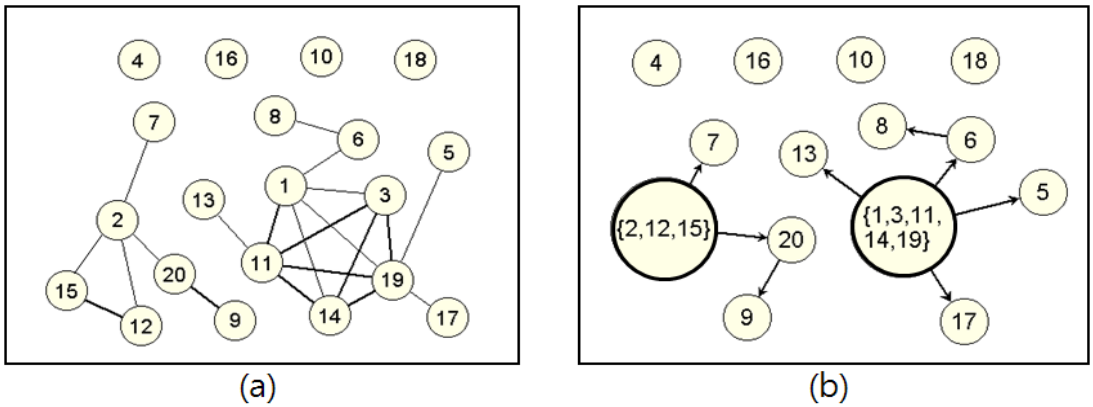


Fig. 24. **(a)** Augmented Communication Network in **Fig. 18**, where *N = 20* and *E = 21*. **(b)** The simplified Communication Network from **(a)**. Each circle node denotes an agent, where *N = 10*, *E = 8*.

Table 6. Five random testing graphs, *Gi*. |*A*| and |*O*| denotes the number of agents and objects, respectively. *Pa* denotes the edge probability between agent nodes and *Pa,o* denotes the edge probability between agent and object nodes.

	$\|A\|$	$\|O\|$	P_a	$P_{a,o}$
G_1	20	7	0.1	0.7
G_2	40	14	0.2	0.8
G_3	80	20	0.1	0.7
G_4	150	20	0.2	0.8
G_5	200	20	0.05	0.7

Table 7. The corresponding Generalized Social Network for the five input graph *Gi* in **Table 5**. *Gg* indicates the number of nodes and edges in the general social network including the objects. *Ge* indicates the number of nodes and edges in the Communication Network. The rate denotes the reduced rate of each node and edge, comparing *Gg* to *Ge*.

	$G_g(V_g, E_g)$		$G_e(V_e, E_e)$		$Rate$	
	$\|v\|$	$\|e\|$	$\|v\|$	$\|e\|$	$\|v\|$	$\|e\|$
G_1	27	115	17	60	0.32	0.48
G_2	54	507	23	106	0.57	0.89
G_3	100	1507	35	274	0.65	0.82
G_4	170	3365	29	227	0.83	0.98
G_5	220	3904	29	124	0.87	0.98

5. Conclusions and Future Work

In this paper we proposed a new communication network model for chat agents in a virtual space to provide a more realistic communication environment. First, we proposed a new method to measure the capacity of communication between chat agents by considering the spatial information. So by computing the DCS for two agents, we can obtain the corresponding pairs of chat turns and replies, even if there are numerous obstacles (replay delay etc.) in generic chat dialogue sequences. Second, we proposed an algorithm for clustering a set of text dialogues by using DCS and a novel visualization method for showing the hierarchical structure of chat dialogues. The hierarchical representation of chat dialogues by chat flow graphs is a useful method for depicting all of the chats, in terms of topical coherence. Finally, we proposed a new communication network model to reveal the microscopic aspect of a social network.

There are several issues for future work. First, it is necessary to experiment with a sufficiently large number of users. In this paper we only experimented with 7 users due to the constraints of our system specification. We are preparing an experimental environment to accommodate more than 30 users. We have to collect empirical chat history and study on more quantitative analysis methods. In addition, we need to improve our model to show human behavior more realistically. Also, we hope to associate our model with current social networks such as Twitter and Facebook.

Acknowledgement

All the authors would like to thank reviewers for helpful comments and suggestions on this paper.

References

[1] M. A. Smith, J. J. Cadiz and B. Burkhalter, “Conversation trees and threaded chats,” in *Proc. of the ACM Conf. on Computer supported cooperative work*, pp. 97–105, 2000. Article (CrossRef Link).

[2] M. A. Smith, S. D. Farnham and S. M. Drucker, “The social life of small graphical chat spaces,” in *Proc. of the SIGCHI Conf. on Human factors in computing systems*, pp. 462–469, 2000. Article (CrossRef Link).

[3] V. Tuulos, V. Tuulos, H. Tirri and H. Tirri, “Combining topic models and social networks for chat data mining,” in *Proc. of IEEE/WIC/ACM Int. Conf. on the Web Intelligence*, pp. 206–213, 2004. Article (CrossRef Link).

[4] D. Kurlander, T. Skelly and D. Salesin, “Comic chat,” in *Proc. of SIGGRAPH 1996*, pp. 225–236, 1996. Article (CrossRef Link).

[5] F. B. Vi´egas and J. S. Donath, “Chat circles,” in *Proc. of the SIGCHI Conf. on Human factors in computing systems*, pp. 9–16, 1999. Article (CrossRef Link).

[6] T. Alpcan, C. Bauckhage and E. Kotsovinos, “Towards 3d internet: Why, what, and how?” in *Proc. of the Int. Conf. on Cyberworlds*, pp. 95–99, Oct. 2007. Article (CrossRef Link).

[7] Microsoft Corp, “Microsoft Introduces V-Chat Communications for MSN,” *The Microsoft Network*, Microsoft Press Release. Redmond, WA 98052, 1995.

[8] J. R. Frisvad, P. Falster, G. L. Mo´ller and N. J. Christensen, “Knowledge exchange between agents in real-time environments,” in *Proc. of the Int. Conf. on Computer Animation and Social Agents*, 2005, pp. 127–132, 2005. Article (CrossRef Link)

[9] E. F. Churchill and C. A. Halverson, “Guest Editor’s Introduction: Social Networks and Social Networking,” *IEEE Internet Computing*, vol. 9, no. 5, pp. 14-19, 2005. Article (CrossRef Link).

[10] G. Kossinets, J. Kleinberg and D. Watts, “The structure of information pathways in a social

communication network," in *Proc. of the 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 435-443, 2008. Article (CrossRef Link).

[11] J.S. Saltz, S. R. Hiltz and M. Turoff, "Student social graphs: visualizing a student's online social network," in *Proc. of the ACM Conf. on Computer Supported Cooperative Work*, pp. 596-599, 2004. Article (CrossRef Link).

[12] J. I. Khan and S. S. Shaikh, "Computing in social networks with relationship algebra," *Journal of Network and Computer Applications*, vol. 31, no. 4, pp. 862 – 878, 2008. Article (CrossRef Link).

[13] D. Rosen and M. Corbit, "Social network analysis in virtual environments," in *Proc. of the 20th ACM Conf. on Hypertext and hypermedia*, pp. 317-3212, New York, NY, USA, 2009. Article (CrossRef Link).

[14] S.-H. Ji, J.-W. Kim, T.-J. Yoon and H.-G. Cho, "How to Manage Interactive Dialogues of Avatar Agents by Using 3D Spatial Information of Virtual World", in *Proc. of the 25th Symposium on Applied Computing*, pp. 1181-1187, 2010. Article (CrossRef Link).

[15] F. van Ham and N. Krishnan, "Ask-graphview: A large scale graph visualization system," *IEEE Trans. on Visualization and Computer Graphics*, vol. 12, no. 5, pp.669–676, 2006. Article (CrossRef Link).

[16] R. Blanch and E. Lecolinet, "Browsing Zoomable Treemaps: Structure-Aware Multi-Scale Navigation Techniques," *IEEE Trans. on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1248–1253, Nov.-Dec. 2007. Article (CrossRef Link).

[17] Y. Tu and H.-W. Shen, "Visualizing changes of hierarchical data using treemaps," *IEEE Trans. on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1286–1293, 2007. Article (CrossRef Link).

Jong-Woo Kim received his B.S., M.S. and Ph.D. from Pukyong National University, Korea, in 1998, 2003 and 2006, respectively. Since 2009 he has been a Postdoctoral researcher in Center for U-Port IT Research and Education, Pusan National University, Korea. His research interests include Human-Computer Interaction and computer graphics.

Seung-Hyun Ji recevied the B.S. degree in Computer Science from Pusan University of Foreign Studies (2008), Korea, the M.A. (2010) degree in Computer Science from Pusan National University (2010), Korea. Since 2010 he has been a researcher in Navis Automation System, Korea. His research interests include Path Finding Algorithm and 3D Graphics.

Seon-Yeong Kim received the B.S. degree in Computer Science and Engineering from Pusan National University (2010), Korea. She is currently a master student in Computer Science, Pusan National University, Korea. Her research interests include image processing and bioinformatics.

Hwan-Gue Cho received the B.S. degree from Seoul National University (1984), Korea, the M.S. (1986) and Ph.D. (1990) degrees from Korea Advanced Institute of Science and Technology, Korea. Since 1990 he has been a Professor in Pusan National University, Korea. His research interests include image processing and computational photography.