POSTER: A Visualization and Management System for Chat Dialogues on 3D Virtual Avatars

Seung-Hyun Ji, Soo-Hyun Park, Dong-Sung Ryu and Hwan-Gue Cho
Dept. of Computer Engineering
Pusan National University, Republic of Korea
{shji,shpark07,dsryu99,hgcho}@pusan.ac.kr

ABSTRACT

Chat is one of basic features in current virtual world communication. So tracking of chat dialogue is an important issue. However most chat programs only provide textual history transcripts based on the temporal sequence, without any explicit relationship tags. In this poster, we propose a novel data structure, Chatting Flow Graph (CFG)[6] which keeps track of the all chatting dialogues among virtual agents. On this CFG we newly propose one visualization method for managing Chat Dialog efficiently and hierarchically.

Keywords: Virtual World, Chat, Communication, Graph Visualization.

1 ISSUES IN VIRTUAL CHAT

Internet chatting is one of common amusements for people in Web Space. Due to rapidly evolving communication technologies, the framework for Internet chatting is evolving. Traditional chatting with plain texts is evolving into systems supporting multimedia. Especially current Internet technologies such as 'Second Life' and 'IMVU' enable us to chat in virtual space. Chatting among avatar agents in virtual space is quite different from those who chat on the plain text, since avatars can move around the virtual space and talk to different agents with different time. So, it is not easy to reconstruct the corresponding pairs of question/answer from a set of all dialogue texts appeared in the virtual space. In the previous chatting system, they should prepare another communicating session layers to make a pairwise chatting. or additional manual intervention such as clicking to designate talking objects is required as was done in 'Yahoo' and Microsoft MSN instant messenger. Since the current chat systems provide unstructured dialog sets and the distribution of chat history data are so widespread, there are many tasks for user to browse their chat history.

We know Figure1 (a) is a typical snapshot representing virtual chat system and Figure1 (b) shows typical plain-textual chat system. In Figure1 (b) we can not easily match the corresponding pair of question and answer. So we need another ways of representing and managing chat dialogues, especially in 3D virtual space.

(a) (b)

Figure 1: Internet Chat System. (a) 2.5-D Chat in Second Life. (b) Textual history transcripts(NATEON tool supporting Korean) with 8 chat agents.

2 CHAT DIALOGUE

As far as we know, the prevailing chatting tools only keep textual data according to the temporal sequence. In this case, if we do not find the corresponding answer for a query question. So we proposed Chatting Flow Graph (CFG)[6] for managing chat dialogues among multiple agents in virtual space. This helps us to explicitly designate the talking dialogue pairs with considering the geometric information of virtual space(the location of avatars and their facing direction and distance in-between.) Let us define Virtual Chat Bandwidth(VCB)[5] for creating CFG.

Definition 2.1 *For two chat agents A_a and A_b, **VCB(Virtual Chat Bandwidth)** is defined the communication capacity bandwidth as follows. If $(\overrightarrow{N_a} \cdot \overrightarrow{N_b} \geq 0)$, then $VCB(A_a, A_b) = 0$. Otherwise let,*

$$VCB(A_a, A_b) = C_1 \cdot \frac{(\sin\theta_a \cdot \sin\theta_b)^{k_1}}{(dist(A_a, A_b) + C_2)^{k_2}}$$

 ISBN 978-80-86943-95-4

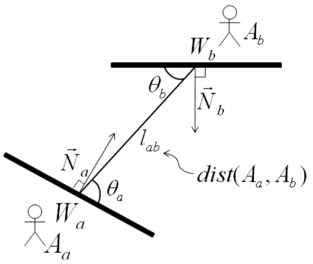

Figure 2: Avatars A_a and A_b are chatting. The solid line segment denotes the width of word balloons W_a and W_b appearing on each head to show chat texts.

, where C_1, C_2,k_1 and k_2 are control constants. [6]

Figure3 shows our Chat Flow Graph. Each node denotes the message with timestamp and agent. Edges are determined through timestamp Δt and VCB by considering the spatial information.

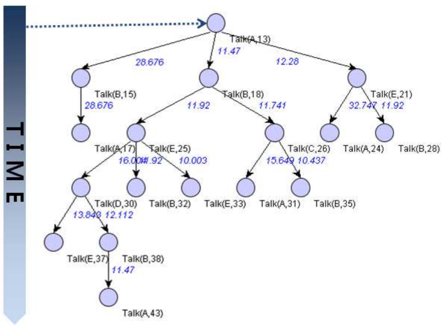

Figure 3: Chat Flow Graph.

This directed acyclic graph,CFG, could be more flexible to the way of keeping chat linearly.

(a) (b)

Figure 4: (a) Our 3D-Chat System. (b) Chat Flow Graph for the conversation between two avatars.

3 VISUALIZATION OF CHAT DIALOGUE

In previous work, we have shown CFG could be a good solution for managing a set of chat dialogues [6]. But according to the features of the graph, if there are multiple nodes then CFG is difficult to understand. So, we need to make a reasonable visualization system that makes easier for people to understand the chat history.

Suppose that there are 100 agents chatting in a virtual space in the form a small(5-6) talking group. If we want to cluster a set of all dialogues into a meaningful chat group(a talk in a specific topic), then subgraph clustering is needed to show a group of chat subgroup. Thus we give a new hierarchical graph decomposition method where a group of topic chat is represented into a rectangle. So whole CFG is successively partitioned into sub-rectangles recursively. Algorithm 1 explains how to apply graph-cut on Chatting Flow Graph(CFG).

Algorithm 1 Graph-cut Algorithm

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

procedure $CFGRAPHCUT(Gs, min-depth)$
  if $checkHcut(Gs) = TRUE$
    $HGs \leftarrow$ **do** $Hcut(G)$;
    $CFGRAPHCUT(HGs, min-depth)$;
  else if $checkVcut(G, min-depth) = TRUE$
    $VGs \leftarrow$ **do** $Vcut(Gs, depth)$;
    $CFGRAPHCUT(VGs, min-depth)$;
  else return
end procedure

The basic idea of CFG cut is the minimum edge cut to find a more dense sub-graph partitioned. And $min-depth$ is a control constant and $checkHcut()$ and $checkVcut()$ are responsible to find the vertical/horizontal cut. If we find the minimal graph cut, then we separate them by the function $Hcut()$(horizontal-cut) and $Vcut()$(vertical-cut). The graph decomposition works till the size of subgraph divided is greater than a threshold value which is determined empirically. In our procedure we prefer the vertical cut to the horizontal cut.

Now we explain how to map these graph partition into a nested rectangle structures. Whenever we get two partitioned subgraph, then we assign then into a rectangle. The size of rectangle is dependent on the number of nodes(= the number of dialogues of chat avatar). Next we perform this procedure recursively till we get a small enough graph cluster(a topic chat).

Figure 5 shows an example of Chat Flow Graph(CFG). And Figure 6 shows the visualization result from Figure 5. In Figure6, the color of the recursively divided rectangles help us to distinguish each other. Also we provide an nice interactive user interface to review the a set of highly related chat text our of all dialogues by a simple mouse click. If you point one of divided unit sub-rectangle, then it immediately represents the chat dialogues in the form of directed acyclic graph as was done in CFG. So we you can read the a set of related dialogues assigned in a rectangle. Also if you control the viewing scale of rectangle visualization, we can view the more wider chat dialogues.

4 EXPERIMENT

We show some experimental results obtained from a movie, 'Dead Poets Society'. In one movie scene, one group of five men is talking together in a room. We calculated the VCB virtually considering the the movie scene to construct and visualize the CFG corresponding. In this example we adopted the the actual dialogues obtained the movie script and checked the time stamps of each dialogue by manual work in realtime.

Table 1: Textual script for agents in Figure 7.

Time	Agent	Message
00	B	"*Keen. Meeks, door closed.*"
02	E	"*Yes sir.*"
04	A	"*Gentlemen, what are the four pillars?*"
07	every	"*Travesty, horror, decadence, excrement.*"
09	B	"*Okay, good study group. Meeks aced Latin. I didn't quite flunk English. So, if you want, we got our study group.*"
13	A	"*Sure, Cameron asked me, too. Anyone mind including him.*"
15	B	"*What's his specialty, bootlicking?*"
17	A	"*He's your roommate.*"
18	B	"*That's not my fault.*"
21	E	"*I'm sorry. My name is Stephen Meeks.*"
24	A	"*This is Todd Anderson.*"
25	E	"*Nice to meet you.*"
26	C	"*Nice to meet you too.*"
28	B	"*Charlie Dalton.*"
30	D	"*Knox Overstreet.*"
31	A	"*Todd's brother was Jeffrey Anderson.*"
32	B	"*Oh, yeah. Valedictorian. National Merit Scholar.*"
33	E	"*Oh, well. Welcome to Hell-Ton.*"
35	B	"*It's every bit as tough as they say, unless you're a genius like Meeks.*"
37	E	"*He flatters me. That's why I help him with his Latin.*"
38	B	"*And English, and Trig.*"
43	A	"*It's open. Father, I thought you'd gone.*"

Using the proposed 3D virtual system and CFG algorithm, we can construct the corresponding Chat Flow Graph(CFG) for a take in the movie scene(about 2.5 minutes). Figure 8 shows the corresponding CFG. Our visualization algorithm can depict this dialogue graph into a nested rectangle as was shown in Figure 9.

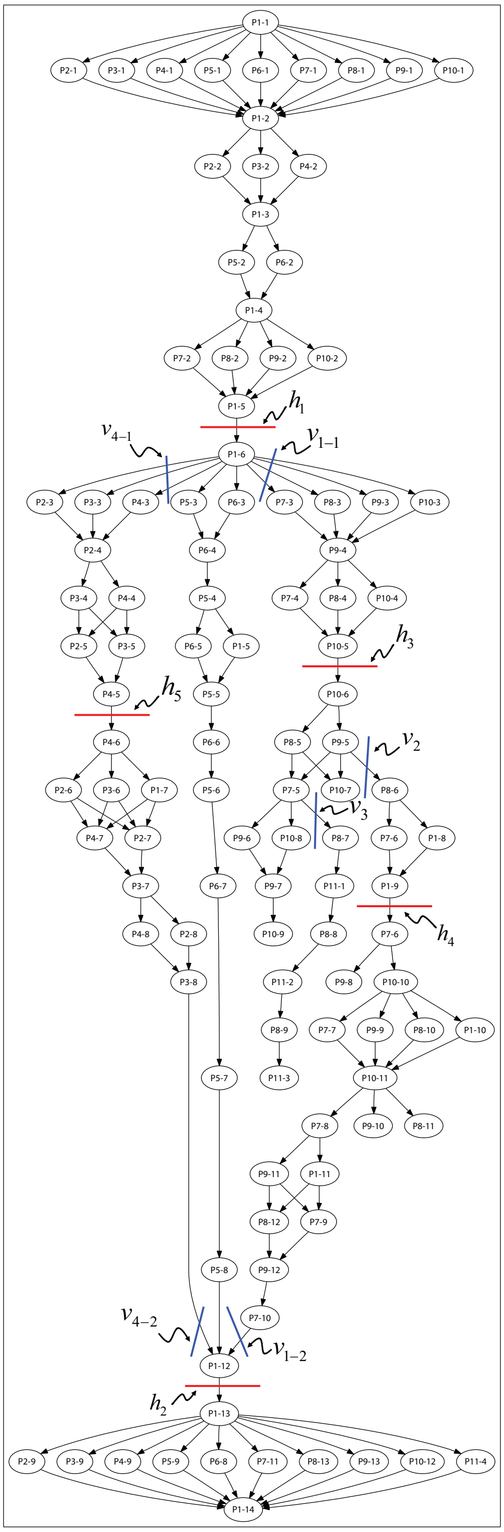

Figure 5: Chat Flow Graph(CFG) showing a conversation on the planning a short trip in an elementary school. Conversation includes one teacher(P1) and ten students(P2P11). In the conversation, first, a teacher is talking about trip plan(data and place) to all students. After this question, students are divided into a three small group to discuss this matter. Each group is talking about their picnic plan. Finally P_{1-14} node shows that all students agreed on the picnic plan.

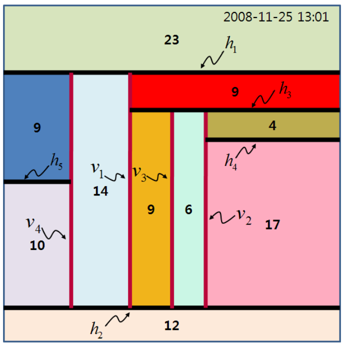

Figure 6: Visualization of CFG in Figure 5. We set threshold parameter min-depth = 3. The number in each sub-rectangle denotes the number of dialogue nodes included.

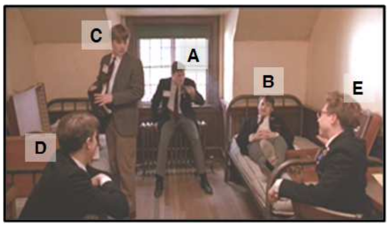

Figure 7: A scene taken from the movie 'Dead Poets Society'. Five men are discussing together.

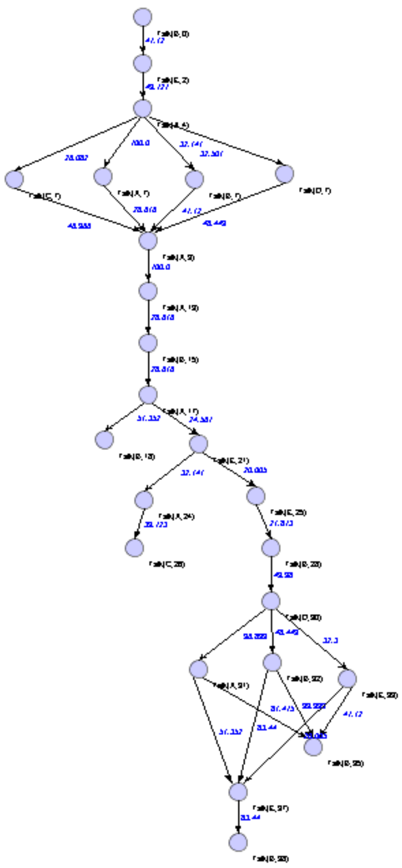

Figure 8: The corresponding Chat Flow Graph for the movie scene of Figure 7.

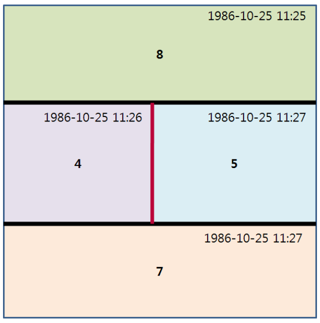

Figure 9: The rectangle visualization result from the CFG in Figure 8. We set the threshold parameter min-depth = 2.

5 CONCLUSION

As Internet chatting in virtual environments becomes prevalent, managing the communication information as a textual script is becoming important. The straightforward method used in the common text based chat protocols(e.g., instant messengers provided by MSN, Yahoo and NateOn) cannot be applied directly to tje virtual reality environments such as the SecondLife model, since they do not keep no semantic relationships in a linear text script. Also they do not utilize the geometric information of virtual avatars. In this paper we proposed a nice algorithm for clustering a set of text dialogue by considering the geometric information about virtual chatting agents. Also we newly proposed a novel visualization method to show the hierarchical structure of chatting dialogues to help us to search a small, but highly related dialogue set(a topic chat). Some experiments show that our approach works properly and efficiently.

REFERENCES

[1] James Abello, Frank van Ham, and Neeraj Krishnan. Askgraphview : A large scale graph visualization system. *IEEE TRANS. ON TVCG, 2006*, 2006.

[2] Renaud Blanch and Eric Lecolinet. Browsing zoomable treemaps: Structure-aware multi-scale navigation techniques. *IEEE TRANS. ON TVCG, 2007*, 2007.

[3] Soo-Hyun Park, Seung-Hyun Ji, Dong-Sung Ryu, and Hwan-Gue Cho. Avachat: A new comic-based chat system for virtual avatars. *15th ACM Symposium on VRST 2008*, pages 279–280, 2008.

[4] Soo-Hyun Park, Seung-Hyun Ji, Dong-Sung Ryu, and Hwan-Gue Cho. A new cognition-based chat system for avatar agents in virtual space. *The 5th ACM SIGGRAPH International Conference on VRACAI 2008, To appear*, 2008.

[5] Soo-Hyun Park, Seung-Hyun Ji, Dong-Sung Ryu, and Hwan-Gue Cho. A smart and realistic chatting interface for gaming agents in 3-d virtual space. *2008 International Conference on CyberGames 2008*, pages 32–38, 2008.

[6] Soo-Hyun Park, Seung-Hyun Ji, Dong-Sung Ryu, and Hwan-Gue Cho. A textual dialogue management system for internet chatting over virtual agents communication. *4th International Conference on NCM 2008*, pages 165–171, 2008.

[7] Ying Tu and Han-Wei Shen. Visualizing changes of hierarchical data using treemaps. *IEEE TRANS. ON TVCG, 2007*, 2007.