AUTOMATIC PLAYBACK OF 3D BLOG ENTRIES BASED ON 3D VIEWPOINT SIMILARITY

Rieko Kadobayashi
National Institute of Information and Communications Technology, 3-5 Hikaridai, Seika-cho, Kyoto 619-0289 Japan

Keywords: 3D Blog, 3D viewpoint information, Community support, Blog search, TrackBack, Viewpoint similarity.

Abstract: The 3D Blog system allows a user to interact with a 3D model, annotate any spot on the model, and publish the annotations in a blog so that the user can easily communicate with other people and share interests, idea, and questions about 3D models of interest to the user. The main features of the system is that each annotation is saved as a blog entry with 3D viewpoint information, i.e., the user's viewpoint and view direction as well as 3D information on annotated points. With this 3D viewpoint information, readers of a 3D Blog can reconstruct the exact same 3D scene the blogger saw when reading the blog entry, and, therefore, the readers may understand it better. The system also uses the 3D viewpoint information to detect similar entries and sends trackback links automatically to enable users to easily find people who may have similar interests. Users can easily and intuitively compare the similar entries using the "playback" function which renders the 3D scene in the similar entries. The 3D Blog system is expected to promote emergence of communities of people who share interests and support their communication.

1 INTRODUCTION

The number of 3D Computer Graphics (CG) models available on the Internet is rapidly increasing due both to strong demand for 3D models in such domains as online catalogues and cultural heritage sites and to the advancement of digitizing and CG technology. For example, in the cultural heritage sector, there is an increasing demand to be able to record and visualize cultural heritage items in 3D CG models because these models have many advantages over traditional recording and presentation media such as 2D drawings. In online catalogues, 3D models can provide consumers with more detailed information and interactivity than photographs can. This means that users can change points of view as freely as if they were looking at an object in a real space, and change the texture or color of a 3D model as many times as they want to test how its appearance changes, which may actually be more difficult in real space.

The number of blog users (bloggers) is also rapidly increasing. Blogs are now a very important and widespread medium for people to express and exchange ideas. One of the key features of a blog is that it allows readers to comment on blog entries, meaning that the owner of a blog site and its readers can easily communicate with each other. This bi-directional communication is facilitated by "TrackBack" mechanism which some blog systems use.

However, the use of 3D models is still restricted. Many 3D model applications provide users with some interactivity but do not provide users with a communication function for using the models. Assume that you are browsing an online catalogue and you like the design of a particular object. You may need to write a long sentence that explains why you think the design is excellent or take a snapshot of your computer display of the 3D model of the object if you want to recommend it to your friend. If the online catalogue supports communication using 3D models, the task is much easier, and you may only need to send a link to a 3D view of the object in a particular orientation and a comment to your friend.

Another example in the cultural heritage sector arises when people visit a museum. Visitors can usually share the experience with their family, friends,

Kadobayashi R. (2007).
AUTOMATIC PLAYBACK OF 3D BLOG ENTRIES BASED ON 3D VIEWPOINT SIMILARITY.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Web Interfaces and Applications*, pages 18-25
DOI: 10.5220/0001264100180025

or schoolmates, watching how they interact with exhibits. This observational or social learning is thought to be very important (Falk and Dierking, 2000). Sharing knowledge, interests, impressions, and experiences can deepen each person's understanding and appreciation of the exhibits. In addition, there are a lot of hands-on exhibits in real museums that enable visitors to interact, which facilitates their understanding.

However, existing 3D applications support only limited sharing and interactivity. This is especially true of applications for the general public. Experts can interact with 3D models using proprietary or expensive commercial software, but the general public is allowed only to view them passively, e.g. to see still images or animation in most cases. The experts may convey their interpretation of cultural heritage via 3D visualization, but the general public does not have any such opportunity.

Of course, there are 3D applications targeting the general public such as virtual 3D museums. For example *virtual Leonardo* provides on-line chat functions so that users can communicate with each other (Paolini et al., 1999). The main problem in virtual 3D museums, from the viewpoint of communication and collaboration, is that users need to be in a virtual museum at the same time. Practically, this may be an overly restrictive requirement, and other means of collaboration should be provided (Barbieri and Paolini, 2000).

To overcome these problems, the 3D Blog system has been proposed. This system allows a user to interact with a 3D model, annotate any spot on the model, and publish the annotations in a blog so that the user can easily communicate with other people and share interests, ideas, and questions about 3D models of interest to the user (Kadobayashi, 2005). One of the main features of a 3D blog is that each annotation is saved as a blog entry with 3D information on the user's viewpoint. Readers of a 3D Blog can reconstruct the exact same 3D scene the blogger saw when reading the blog entry, perhaps helping the readers to understand it better.

In this paper, we propose a new mechanism for viewing 3D Blog entries that enables users to find similar entries easily and that intuitively calls their attention to the similarities among entries by using 3D models. It also enables them to find people who may have similar interests and to promote emergence of communities of people who share interests.

The rest of this paper is organized as follows. Section 2 introduces the 3D Blog system and describes 3D viewpoint information. In Section 3, the algorithm for finding similar entries and the mechanism that links them are described. In Section 4 the algorithm for playing back similar entries is explained. Section 5 discusses the features of the 3D Blog system that make it a very useful communication tool. Section 6 concludes the paper.

2 THE 3D BLOG SYSTEM

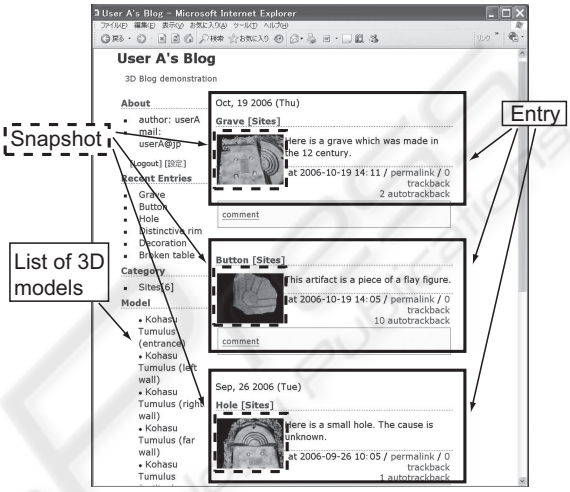


Figure 1: Top page of a 3D Blog. Annotations to 3D models are listed as blog entries in reverse chronological order. Each entry has a snapshot of the 3D scene the user saw when annotating.

2.1 Features

3D Blog enables users to interact with a 3D model, annotate any spot on the 3D model, and publish the annotations as a blog. It does not require any expensive proprietary software to make annotations or read blog entries. It only requires a 3D model viewer that runs as a browser plug-in for interacting with 3D models. The current 3D Blog system uses lily (lily, 2005) as a blogging tool and Cortona VRML Client (ParallelGraphics, 2006) as a viewer for the 3D models. Lily is a Ruby implementation of blosxom (Blosxom, 2006) written in Perl. Users need to download and install the plug-in, i.e., Cortona VRML Client, to their web browser (we are currently using Internet Explorer only) before they start 3D blogging.

The main feature of 3D Blog is that it uses the 3D viewpoint information of the 3D scene that the user (blogger) sees when adding an annotation. The 3D viewpoint information is composed of viewpoint and view direction, which are expressed as 3D coordinates, and a 3D vector. The 3D viewpoint information, as well as the position of an annotated spot,

is automatically added to the annotation by the system and stored in the database. It is then used to reconstruct the same 3D scene the blogger saw when a reader reads entries. This is thought to greatly help readers to understand the content of the entries.

The viewpoint is the point from which a user looks at the point being annotated. The view vector is a vector from the viewpoint to the point being annotated. The view direction is derived by normalizing the view vector. In implementation, these items are altered by the parameters specific to a computer graphics software. For example, in VRML, which we used in our system, the viewpoint and the view direction are expressed as parameters "translation" and "rotation" of a camera.

This system is based on the assumption that impressions may change if the user's point of view changes, even if the user looks at the same thing. The resulting design may enable users to convey their feelings more accurately if they can show the object as it looked when they saw it.

The 3D viewpoint information is also useful for searching (Kadobayashi and Tanaka, 2005; Fujita and Arikawa, 2005) and in the 3D Blog system is used to find similar entries. When a new entry is posted, the system detects similar entries based on a threshold set by users and sends links from the new entry to similar entries. We call this mechanism "automatic TrackBack." It helps users find other users with similar interests and create communities and facilitates communication among users.

To summarize, by embedding 3D viewpoint information into blog entries, the system enables users to easily understand the idea of blog entries by playing back a viewpoint of the annotated point, finding similar entries, and communicating with similar users, which facilitates forming communities.

2.2 Writing 3d Blog Entries

When a 3D Blog user logs in, the top page of the user's blogsite will open and show recent entries in reverse chronological order, as in ordinary blogsites (Figure 1). By selecting a 3D model from the list on the left-hand side of the window and clicking on the model's name, the 3D model viewer will appear at the top of the list of entries enabling the user to browse and interact with the model in the walk-through, fly-through, or study mode and to add notes about the model (Figure 2).

When the user clicks on a spot that he/she wants to annotate, a small ball, i.e. a marker, appears and covers the spot in the 3D model viewer. The user types an appropriate title and a comment in the dialogue box

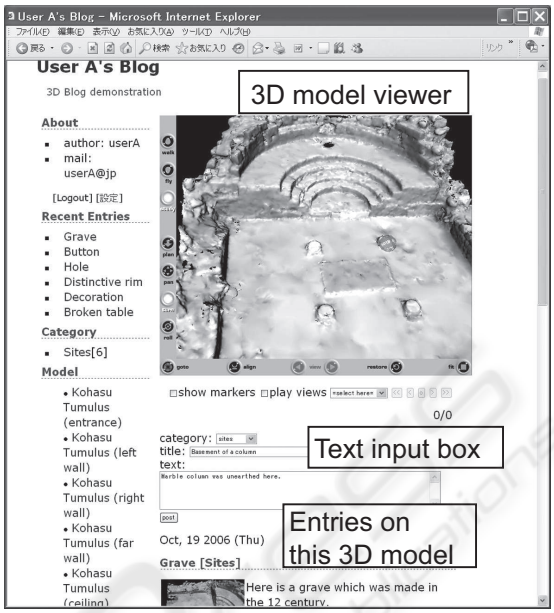


Figure 2: Typical page layout for interacting with and annotating 3D models. 3D model viewer is displayed on top of entry list.

below the 3D model viewer and then clicks the 'post' button.

Information, such as the position of the annotated area, the viewpoint of the virtual camera which is treated as that of the user, the date and time, the snapshot of the 3D model with the marker, and the annotation (i.e., the title and comments) are converted into a blog entry and stored in the blog database. The annotation is then added to the top of the entries of the user's blog site as the latest entry.

Users can annotate a 3D model as often as they desire. All annotations are automatically added to the blog site as separate entries. The user can change 3D models by clicking on a model's name and continuing to blog on a different model.

2.3 Reading 3d Blog

At first glance, the top page of a 3D blogsite looks similar to the top pages of an ordinary blog, but there is a big difference. Every entry in the 3D blog includes information about the 3D viewpoint, i.e., the viewpoint from which the 3D model was annotated and about the spot to which the annotation was added.

The snapshot included in each entry is an anchor to the 3D scene and, once it is clicked by a reader, the 3D model viewer pops up to render the 3D scene using the 3D viewpoint information while at the same time enabling the reader to read the entry (Figure 3). When rendering the 3D scene, the system first

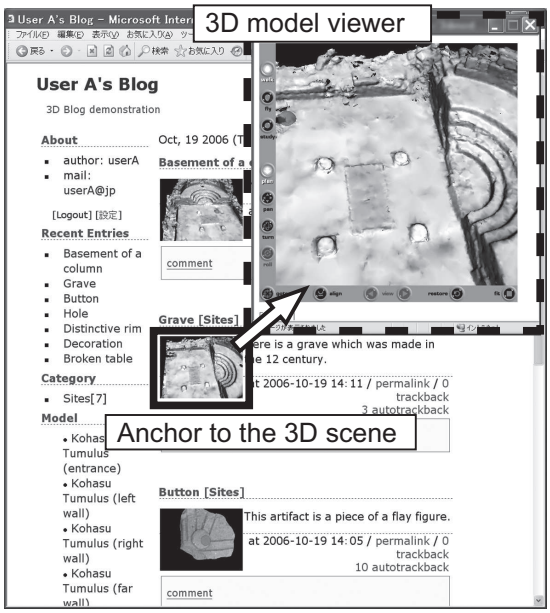


Figure 3: 3D model viewer pops up when a snapshot of an entry is clicked since a javascript is embedded as an anchor to the 3D scene. The user can read the entry and confirm how the 3D model looks.

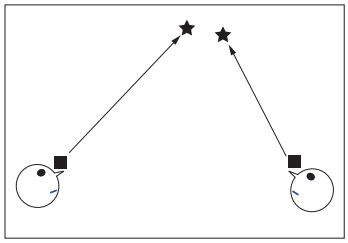
shows the 3D model in a default orientation and then changes the orientation of the model so that the 3D scene is exactly the same as the blogger saw. This animation brings to the reader's attention how the blogger changed viewpoints when annotating. This is a great help to readers in understanding the content of entries, because it enables them to visually confirm it.

Readers can add comments to an entry by clicking the 'comment' button at the bottom of the entry to open a dialogue box. Other basic functions such as viewing old entries and searching entries are also available.

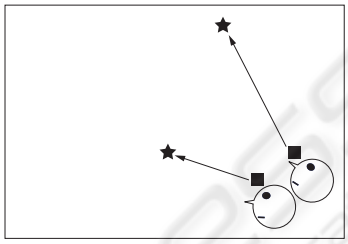
3 FINDING SIMILAR ENTRIES

The 3D Blog system uses the 3D viewpoint information to find similar entries and to render 3D scenes and calls user attention to these similar entries by automatically setting TrackBack links. There is a blog service that connects entries that share the same keywords (hatena, 2007), i.e., semantic similarities. In contrast, the 3D Blog system uses the physical perspective similarities.

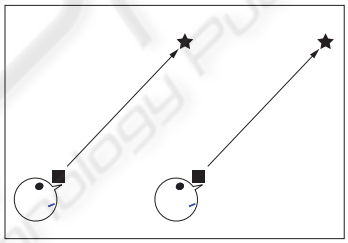
In the current implementation, four measures are used to detect similar entries: *target*, *viewpoint*, *direction*, and *gaze*. *Target* is the distance between annotated points while *viewpoint* is the distance between viewpoints. Similarity in *direction* is obtained by cosine similarity of view direction vectors. Since the value of cosine theta is not linear, it is not reasonable to expect users to set the threshold. As a result, the degree of the angle between two vectors is used as a measure and users set the threshold from 0 to 180 degrees. *Gaze* is a combined measure of *target* and *viewpoint*. If *target* is less than or equal to the threshold and *viewpoint* is less than or equal to the threshold, the entries are similar in terms of *gaze*. Figure 4 depicts examples that are similar for each measure.



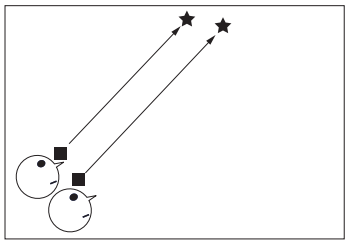
(1) Targets are similar.



(2) Viewpoints are similar.



(3) Directions are similar.



(4) Gaze is similar.

Figure 4: Similarity measures.

Usually the TrackBack link is used by a blogger who writes an entry referring to another blogger's en-

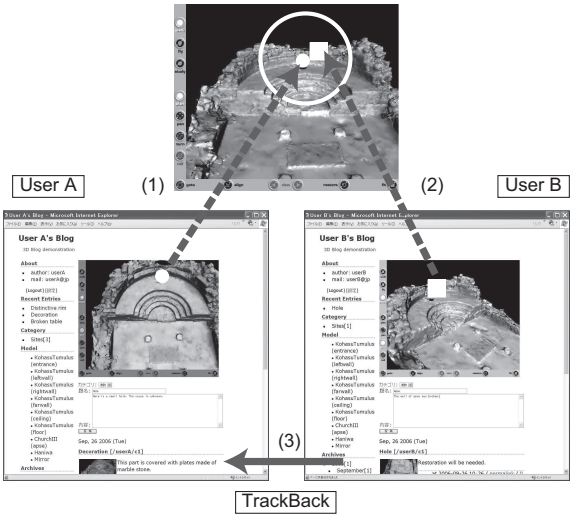


Figure 5: Automatic TrackBack is conducted by the system when similar entries are found. Users may easily find other users with similar interests.

try to explicitly notify the other blogger of the reference. In the 3D Blog system, the system also automatically sends TrackBack to entries in the database if it finds similar entries as new entries are written.

The system checks the similarities between each new entry and the stored entries whenever a new entry is posted. Then it makes the TrackBack links from the new entry to the stored entries. This means the latest entry by any user of the 3D Blog system has no automatic TrackBack links. The 3D Blog system includes the information about the similarity measures, i.e., the similarity score, as well as such information as the title and URL of the blog, which is normally included in the TrackBack ping (SixApart, 2004). The blogger's own entries are excluded from the similarity check since it is unremarkable that entries by the same blogger tend to be similar.

This automatic TrackBack mechanism helps 3D Blog users find similar entries and others who may have similar interests or perspectives. Users can set the threshold according to their preference. For example, some people may want to find people who have similar interests and set a low threshold for the distance between annotated spots. Others may want to find people who have a similar (physical) perspective and set that distance threshold low.

3.1 Finding Similar Entries: A Scenario

Assume that user A has written an entry about a 3D model and the entry is stored in the database (see (1) in Figure 5). After user B writes an entry about the same 3D model (see (2) in Figure 5), the system

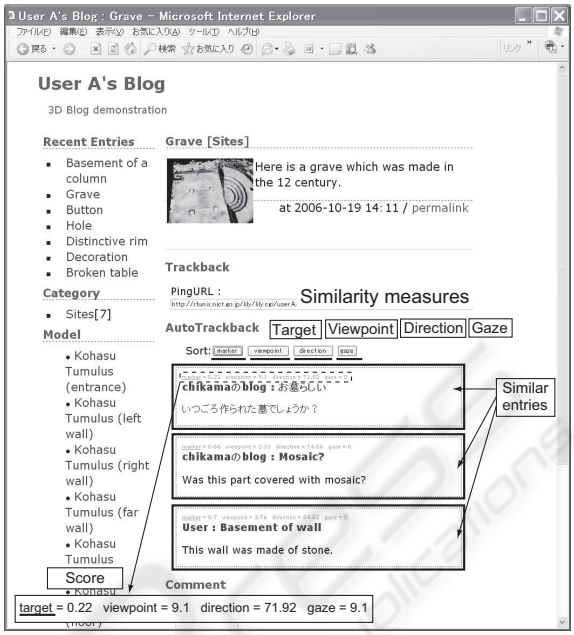


Figure 6: List display of similar entries. Entries are sorted in ascending order of the score of the measure selected by the user. In this case entries are sorted using the score of "target" measure.

checks for similar entries in the database. In this case, the system finds that an entry written by user A is similar to the entry that user B has just posted because the distance between the annotated spots of the two entries is very small and is less than the threshold specified by user A. It then sends TrackBack to user A's entry so that user A will notice user B's entry (see (3) in Figure 5).

4 AUTOMATIC PLAYBACK OF 3D BLOG ENTRIES

The 3D Blog system enables a user to see similar entries in playback mode as well as in list mode. The list mode, which is shown in Figure 6, looks similar to many blog pages. The user's entry is shown on top of the blog page and similar entries written by other bloggers are shown below the user's entry. Users can choose the sort key from the four measures. For example, if *target* is chosen, the entries are sorted in ascending order of the *target* score.

In the playback mode, by contrast, the user can playback the 3D scene in each similar entry so that she or he will notice which spots on a 3D model are of interest to other users and which of those spots are annotated by other users. First, the user chooses "user

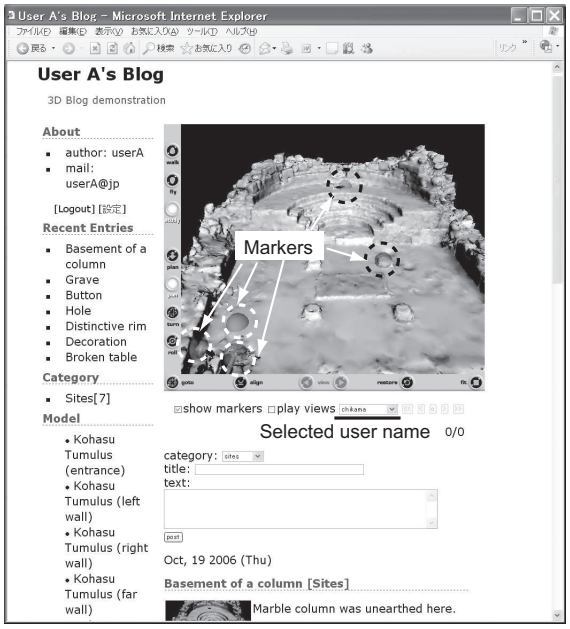


Figure 7: All markers placed by a selected user in similar entries are shown when "show markers" button is clicked. Users can easily grasp the distribution of annotated points.

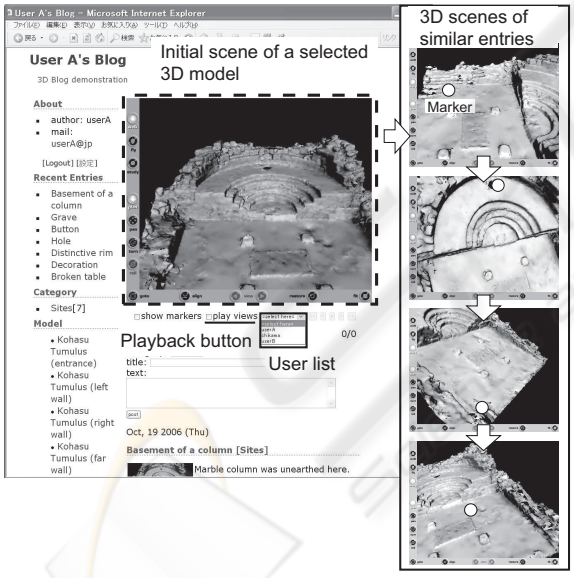


Figure 8: Playback of 3D Blog entries. Choose a user from pull down menu and then click the playback button. The system automatically shows the 3D scenes of the entries in sequence in the 3D model viewer.

name" from the pull down menu located just below the 3D model viewer. This menu shows the names of those who have made entries similar to the user's. Second, the user chooses a function by checking either "show markers" or "play views."

The "show markers" function shows all the markers from similar entries written by the selected user, as shown in Figure 7. Playback of the 3D Blog entries in "show markers" mode thus helps users to visually grasp the distribution of other user's interests. If the user selects the entries of another user from the pull down menu, the user can graphically see which spots she or he annotated.

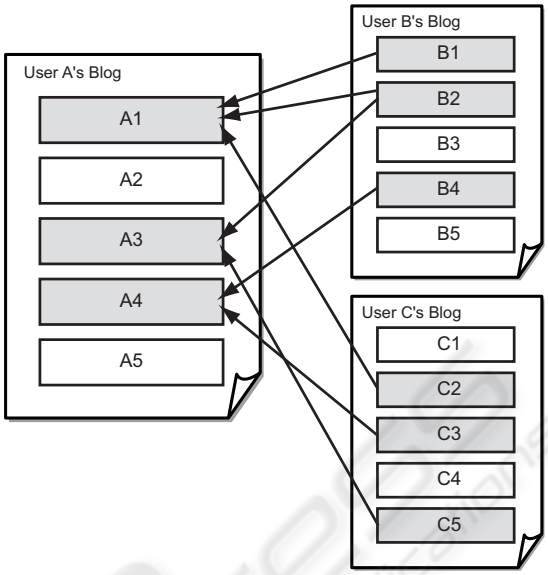


Figure 9: Relation of similar entries. For simplicity, automatic TrackBack links from user A's entries to other users' entries are not shown.

Alternatively, the "play views" function renders all the 3D scenes of similar entries one by one, smoothly changing the viewpoint. This helps users see which other users saw and annotated the 3D model and how the viewpoint changed spatially. The algorithm for "play views" is as follows: First the system collects all of a particular user's similar entries for a particular 3D model and then categorizes them by writer. Finally the system sorts the entries in reverse chronological order by writer.

Figure 9 shows an example of a relation among entries. User A wrote five entries about a particular 3D model, and three of them have automatic TrackBack links from users B and C's entries. Although entry $B2$ has two links to user A's blog, it is treated as one link. In this case, user A can choose user B or user C in addition to himself from the pull down menu, and playback user B's entries $B1, B2, B4$ or user C's entries $C2, C3, C5$ or his or her own entries $A1, A2, A3, A3, A5$ in the 3D model viewer.

5 DISCUSSION

The most essential feature of the 3D Blog system is that it uses 3D viewpoint information of a user. In other words, the system places importance on from where the user observes the 3D model as well as on to which the user has an interest.

Considering a way of asking in our daily life, it is quite often that we ask a question like "what is that?" when we do not know the name of what we are looking at. Those who are talking with you can understand what is of interest to you since they can share your viewpoint and follow your glance. It is, therefore, 3D viewpoint-based search is very natural way to find things of interest.

In contrast, information about what a user looking at in the 3D Blog system is similar to a keyword in keyword-based search. If a user uses keyword-based search system, the user needs to know keywords or choose keywords from a list to formulate appropriate queries. This requires the user to have knowledge a priori about what he/she will search while 3D viewpoint-based search does not require such assumption.

The automatic playback function enables users to take advantage of the 3D viewpoint information easily and intuitively. As the system automatically changes the 3D scene according to the viewpoint of entries, users do not need to struggle with the 3D model to recreate the 3D scene and therefore they can concentrate on understanding similarities among entries written by other users.

Although the 3D Blog system has many advantages, there are many topics that need further research and development to make it fully functional.

Development of an excellent user interface for 3D models, especially for annotating and comparing 3D models, is crucial. It is difficult, intuitively and otherwise, to select an arbitrary part of a 3D model, but this is essential when adding annotation. In addition, easy-to-use interfaces that compare 3D models in a single viewer should be provided since comparing objects is fundamental to research and discussion.

Functions for analysis are also important in terms of community support tools. The "show markers" function can provide an overview of a spatial distribution of annotated points. Users can know which areas are of interest to community members while they cannot know why the areas are of interest. That is functions to support semantic analysis as well as physical analysis are needed.

6 CONCLUSION

We discussed the automatic playback function for 3D Blog entries. This function is useful especially when viewing similar entries that the 3D Blog system automatically finds based on 3D viewpoint information since users can easily and intuitively understand the similarities among entries by viewing 3D scenes.

The 3D Blog system uses 3D viewpoint information to display blog entries and to find similar entries and users. With this tool, people can easily discuss a particular topic using 3D models. The automatic playback function expands this advantage.

Although further investigation is needed on additional functions of the system to support user communication and knowledge sharing via 3D blogging, we believe 3D blogging with the automatic playback function has great potential and can be applied to many domains.

REFERENCES

Barbieri, T. and Paolini, P. (2000). Cooperative visits for museum WWW sites a year later: Evaluating the effect. In Bearman, D. and Trant, J., editors, *Museums and the Web 2000*, pages 173–178. Archives & Museum Informatics.

Blosxom (2006). The zen of blogging. `http://www.blosxom.com/`. Retreived October 24, 2006.

Falk, J. H. and Dierking, L. D. (2000). *Learning from Museums: Visitor Experiences and the Making of Meaning*. AltaMira Press.

Fujita, H. and Arikawa, M. (2005). A ubiquitous photo mapping considering users' lines of sight. In *Proceedings of International Workshop on Ubiquitous Data Management (UDM2005)*, pages 71–77.

hatena (2007). Hatena diary keyword. `http://d.hatena.ne.jp/keyword/`. Retreived January 9, 2007.

Kadobayashi, R. (2005). Automatic 3D blogging to support the collaborative experience of 3D digital archives. In Fox, E. A., Neuhold, E. J., Premsmit, P., and Wuwongse, V., editors, *Digital Libraries: Implementing Stragegies and Sharing Experiences, Proceedings of the 8th International Conference on Asian Digital Libraries (ICADL 2005)*, number 3815 in LNCS, pages 109–118. Springer.

Kadobayashi, R. and Tanaka, K. (2005). 3D viewpoint-based photo search and information browsing. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 621–622. ACM SIGIR, ACM.

lily (2005). lily. `http://lily.sourceforge.jp/`. Retreived October 24, 2006.

Paolini, P., Barbieri, T., Loiudice, P., Alonzo, F., Gaia, G., and Zanti, M. (1999). Visiting a museum together:

How to share a visit to a virtual world. In Bearman, D. and Trant, J., editors, *Museums and the Web 1999 Selected papers from an international conference*, pages 27–35. Archives & Museum Informatics.

ParallelGraphics (2006). Cortona VRML Client. `http://www.parallelgraphics.com/products/cortona/`. Retreived October 24, 2006.

SixApart (2004). Trackback technical specification. `http://www.sixapart.com/pronet/docs/trackback_spec`. Retreived October 24, 2006.