A Hierarchical Annotation Database and a Dynamic Priority Control Technique of Annotation Information for a Networked Wearable Augmented Reality System

Daisuke Takada*, Takefumi Ogawa**, Kiyoshi Kiyokawa*,+, and Haruo Takemura*,+
*Graduate School of Information Science and Technology, Osaka University,
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
**Information Technology Center, The University of Tokyo,
2-11-16 Yayoi, Bunkyo, Tokyo 113-0032, Japan
+Cybermedia Center, Osaka University,
1-32 Machikaneyama, Toyonaka, Osaka 560-0043, Japan
takada@lab.ime.cmc.osaka-u.ac.jp, ogawa@nc.u-tokyo.ac.jp, {kiyo,takemura}@ime.cmc.osaka-u.ac.jp

Abstract

Filtering annotations is very important in networked wearable AR systems in order for the server to efficiently deliver the information that each user needs to his/her wearable computer. In this paper, we propose a hierarchical data structure associated with the real environment, a dynamic priority control technique for filtering annotations, and divided/weighted transfer of annotations. Our dynamic priority control technique is able to give higher priority to more important annotations according to the user's position and angular velocity of viewing direction. We also show the results of simulation experiments regarding the performance evaluation of the proposed technique.

1. Introduction

A number of studies on augmented reality (AR) have been conducted in recent years. The AR technique is familar in wearable computing because it can provide information that is not visible directly to the user, and navigation systems using wearabe AR technology have been developed[1, 2].

When a navigation system stores and manages a huge amount of data, the computational cost to look for annotation information according to the user's position and viewing angle and to display the information at the correct position will be very high, and this data is difficult for a wearable PC to process. In addition, in multi-user environment which annotations are refered to and updated by multiple users, the cost to guarantee the consistency of data in all user's wearable PCs will be huge. Therefore, a technique to manage annotation information on a specialized server in a network and to distribute adequate annotations on demand would be useful.

Sharing information between many users have been studied for a distributed virtual environment, and some of the results of these studies can be applied to a wearable AR system. However, in order to allow users to explore the real environment as they like and provide adequate annotations according to their position, a number of problems specific to a wearable AR system must be solved(i.e. the technique to consider narrow and unreliable network environment used by wearable PCs and distribute annotations efficiently).

We propose two techniques for a wearable AR system. The first is a management technique of annotation information. This technique focuses on the relation between annotation information and real objects, and considers the spatial structure of the real environment in order to determine the data structure. The second is an efficient distribution technique of annotation information from a server to clients. This technique determines the sending priority for each annotation dynamically and distributes data according to priority.

After a discussion of related research in Section 2, details about the hierarchical data structure of annotations and dynamic priority control technique for annotation transfer are presented in Section 3. Experiments for the evaluation of the proposed method are described in Section 4. Finally, in Section 5, a summary and future directions are presented.

2. Related Research

A number of studies have been conducted on distribution and management technique of annotation information shared with multiple users. DIVE[3] and SIMNET[4] distribute all of information that the server have to clients. Af-

ICAT 2008
Dec. 1-3, Yokohama, Japan
ISSN: 1345-1278

ter that, techniques of filtering virtual objects for distribution and checking whether an object is within view of the user were investigated.

2.1. Filtering objects

In RING[5], the server calculates the visibility of objects from each user according to user's position and orientation and distributes object information to users that can see the object. The system of Hosseini et al. uses the results of visibility calculation in the rendering pipeline in client machines[6]. Other techniques define the space around each user as the Area of Interest (AOI) and calculates whether each object is within each user's AOI. MASSIVE[7] uses an AOI called aura. VELVET[8] changes the size of the AOI according to the number of users and network bandwidth dynamically. The method of Han et al. [9] groups multiple users having overlapping AOIs as an interest group and determines the AOI for each group in order to handle an increased number of users. Although these techniques can use limited bandwidth effectively and provide adequate information required by users, they do not consider prefetching of information and the network bandwidth is not used effectively after the filtering results are tranferred. Therefore, priority evaluation of each object is considered to transferring the results.

2.2. Evaluation of sending priority

Many evaluation methods of objects' sending priority have been also investigated. The method of Beeharee et al.[10] visually assigns higher priority to objects that are more attractive using the color, contrast, orientation, speed, and other characteristics of the objects. The method of Park et al.[11] evaluates the sending priority by using each user's interest rate for objects and the total interest rate of all users for objects simultaneously. The method of Chan et al.[12] and the method of Li et al.[13] assume the user's environment to be a desktop-form virtual environment. They predict the user's movement in virtual space using the user's position in virtual space and the movement of the user's mouse on the desk and evaluate priority using the results of the prediction. However, these methods are focused on the virtual environment. It is thought that, unlike [10], we should not simply assign higher priority to objects that are more attractive visually because our research target is an AR system and the data handled by this system are annotations to the real environment. Furthermore, the movement of the user's viewing direction may be a networked wearable AR system specific problem. In a networked wearable AR system, the movement of the user's head directly affects the user's viewing direction and the movement will often be fast, especially when the user looks around, as compared to the movement in the desktop type distributed virtual environment, which is controlled by the user's mouse. Although [12] and [13] use the user's position and orientation to evaluate the priority and the weight for each parameter is given as a constant, the weight must be changed dynamically in order to adapt to the situation in a networked wearable AR system. Julier et al.[14] proposed a filtering method for mobile AR environment. Their method considers the task which the user is performing to evaluate the priority of objects. However, this method has some problems. At first, the expert for each task must choose the elements which should be considered for the task and the second is parameters used to evaluate priority must be set manually.

2.3. Hierarchical management of annotations

NPSNET[15] divides geometry data into grids of fixed size and manages the data in a quad-tree to achieve efficient data management in virtual environment. Although there are many techniques to construct database hierarchically, several of these methods simply divide the data into grids of fixed size and do not consider the relation between the data and the real environment, which should be considered in AR systems. Kolbe et al.[16] proposed hierarchical data structure to express city environment. In this data structure, each object can have its semantic information such as "building", "bridge", and "monument". However, filtering with semantic information is not considered.

3. Proposed Methods

3.1. A networked wearable AR system

Figure 1 shows an overview of a networked wearable AR system. A networked wearable AR system consists of a server that stores and manages all of the annotation information in a database and wearable PCs that are worn by users and work as clients. The annotation database on a server can be updated by a system administrator of the system and users of this system with wearable PCs. The wearable PC measures the user's position and orientation, sends this information to the server, receives appropriate annotation data from the server according to the user's situation, and then renders the received data.

3.2. Hierarchical data structure considering real environment

This section describes the method of managing annotations in the server. The areas covered by the server are subdivided into small areas hierarchically, and each area corresponds to a space in the real environment. Annotations belong to one of these areas. Figure 2 shows a hierarchical structure of areas and annotations.

Each area has information about the real space corresponding with the area. According to the definition of an area, a category for the area (e.g., "Room," "Floor," "Build-

ICAT 2008
Dec. 1-3, Yokohama, Japan
ISSN: 1345-1278

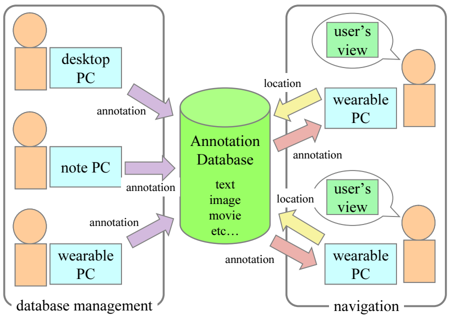


Figure 1. Overview of a networked wearable AR system

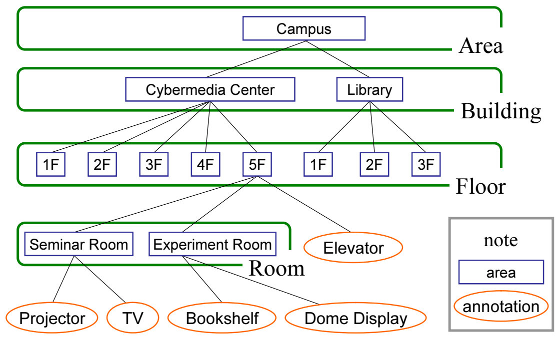


Figure 2. Hierarchical data structure

ing," "Area," "Town," "City," "State/Prefecture") can be set. These categories are used to find annotation information.

Each user can select a yes-or-no policy for each category. Eight different policies can be used:

1. When the user is in the area
   (a) add annotations in the area into the search result
   (b) traverse each child area for a search
   (c) traverse the parent area for a search
2. When the area is traversed from a child area
   (a) add annotations in the area into the search result
   (b) traverse each child area for a search
   (c) traverse the parent area for a search
3. When the area is traversed from the parent area
   (a) add annotations in the area into the search result
   (b) traverse each child area for a search

For example, when a user is in a room and needs annotations which are only in that room or in the buildings around the area, he the user can set "yes" for policies 1(a), 1(c), and 2(c) for categories "Room" and "Floor"; "yes" for 2(a), 2(c), 3(a) for "Building"; "yes" for 2(b) for "Area" to restrict the target area when searching for annotations.

3.3. Dynamic priority control technique

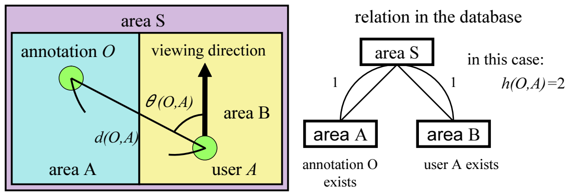


Figure 3. Parameters used for priority evaluation

The system receives a set of annotations that satisfy the search criteria. The system evaluates the transfer priority for each annotation and sends the annotations according to priority. Figure 3 shows the position of user A and annotation O. In this case, the priority of O for the user A, $p(O, A)$ is defined by the following expression:

$$\begin{aligned} p(O,A) &= (1-\frac{h(O,A)}{H(A)})\alpha(1-\frac{d(O,A)}{D(A)}) + \\ &\quad (1-\frac{h(O,A)}{H(A)})(1-\alpha)(1-\frac{\theta(O,A)}{\pi(A)}) \end{aligned} \tag{1}$$

where $d(O, A)$ is the distance between the annotation O and the user A, $\theta(O, A)$ is the angle between the user's viewing direction and the line between O and A, and $h(O, A)$ is the distance between O and A in the hierarchical structure of the database. α is a constant defined as $0 \leq \alpha \leq 1$. When α is small, annotations in front of the user are prioritized, and when α is large, annotations closer to the user are prioritized. In addition, when the distance between the area in which the user exists and the location of the annotation is further down in the tree structure, the priority becomes low because the area should correspond to the semantic space in a real environment as described above. In addition, $D(A)$, $\pi(A)$, $H(A)$ are constants used to normalize variables $d(O, A)$, $\theta(O, A)$, $h(O, A)$, and are given by the user when the connection is established.

When the user's viewing direction is moving quickly and spread over a large area, it is assumed that the user is looking around. When the user's viewing direction is stable, it is assumed that the user needs the information which is in the current direction. In this case, the priority depending on the user's behavior can be evaluated by relating the movement of the user's viewing direction with α. In this case, equation (1) is expanded as follows:

$$\begin{aligned} p(O,A) &= (1-\frac{h(O,A)}{H(A)})s(1-\frac{d(O,A)}{D(A)}) + \\ &\quad (1-\frac{h(O,A)}{H(A)})t(1-\frac{\theta(O,A)}{\pi(A)}) \end{aligned} \tag{2}$$

s and t are defined as:

$$
\begin{aligned}
s &= \frac{S(A)w(A)}{S(A)w(A) + \{1 - S(A)\}\{W(A) - w(A)\}} \\
t &= \frac{\{1 - S(A)\}\{W(A) - w(A)\}}{S(A)w(A) + \{1 - S(A)\}\{W(A) - w(A)\}}
\end{aligned}
$$

where $w(A)$ is the angular velocity that expresses the movement of the user's viewing direction. When $w(A)$ is large, annotations that are closer to the user's viewing direction are prioritized, and when $w(A)$ is small, annotations that are closer to the user are prioritized by equation (2). $W(A)$ is a constant for normalizing $w(A)$ and is given by the user when the connection is established, as with the other constants.

It might not be required to determine the weight for $d(O, A)$ and $\theta(O, A)$ automatically using the angular velocity. For such a case, there is a constant $S(A)$ in equation (2). $S(A)$ is defined as $0 < S(A) < 1$.

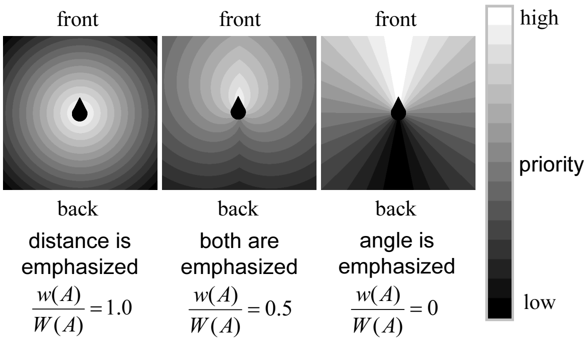


Figure 4. Difference in priority

Figure 4 shows the change in priority when the user's viewing direction is moving. When the angular velocity is large, annotations that are closer to the user are prioritized, and when the angular velocity is small, annotations that are closer to the user's viewing direction are prioritized. When the angular velocity is in the middle, annotations that are closer to the user's viewing direction are prioritized, but the smaller the distance between the annotation and the user is, the higher its priority becomes.

In this technique, it is important to set constant parameter appropriately. We investigate the probable maximum and minimum value of each parameter.

The maximum value of $D(A)$ should be 5,000 [m] because it is impossible to see real objects at distances greater than approximately 5 km, and the minimum value of $D(A)$ should be 10 [m], especially considering the case in which the user is in an indoor environment. $D(A)$ should be between 500 [m] and 1,000 [m].

The maximum value of $\pi(A)$ should be 180 [deg] because $\theta(O, A)$ is 180 [deg] when O is opposite A's viewing direction, and the minimum value of $\pi(A)$ should be 10 [deg] because the minumum horizontal field of view of current commercial HMDs is approximately 20 [deg].

For $W(A)$, the maximum horizontal angular velocity in the head bobbing motion of human is approximately 200 [deg/sec], and the angular velocity was less than 30 [deg/sec] during 97% of the experiment and was less than 120 [deg/sec] during 99.7% of the experiment in the research of Kitamura et al.[17]. When we measured human head bobbing motion for use in the experiment described in Section 4, the maximum horizontal angular velocity was approximately 200 [deg/sec], which was used as the maximum, and the actual value of $W(A)$ should be between 30 [deg/sec] and 120 [deg/sec].

For $H(A)$, the maximum height of the hierarchy in annotation database is 8, because the number of the categories to which areas can belong in the database is 8. Therefore the maximum distance in the database is 16, and it is considered that the maximum should be 16. The minimum should be 1 considering the situation in which the user is in indoors. $H(A)$ should be set according to the height of the actual database used in the system.

3.4. Divided and Weighted transmission of annotation information

Although the server sends annotations according to the priority acquired from the technique described above, some conditions specific to a networked wearable AR system must be considered in annotation transmission. The network bandwidth available in a networked wearable AR system is basically narrow, and the system cannot adapt to the change of priority when the server sends a huge amount of data at one time. Therefore, a technique that divides the list of annotations or annotation data into chunks of fixed size and sends chunks in series may be useful. When the server sends the list of annotations for an area, the total number of annotation in the area and the number of annotation in the chank are described concurrently, and when the server sends the data of annotation, the total data size, the chunk size, and the offset of the chunk from the top of data are described concurrently (Figure 5).

When the user is looking for his destination, rough information about several annotations will be more useful than detailed information about a few annotations. Therefore, the server sends data for multiple annotation that have the highest priority. The data size is weighted by the order of priority. Figure 6 shows the sequence of transferred data in weighted transmission. The number of annotations transferred at one time is n, and the size of buffer transferred at one time is x. When the priority $p(O, A)$ is the kth highest priority, the buffer size for the transmission of O is $\frac{1}{2^k}x$.

Figure 7 and 8 show the result of rendering of annotation images transferred with the divided and weighted transmission technique. Each image is in the 50KB progressive jpeg

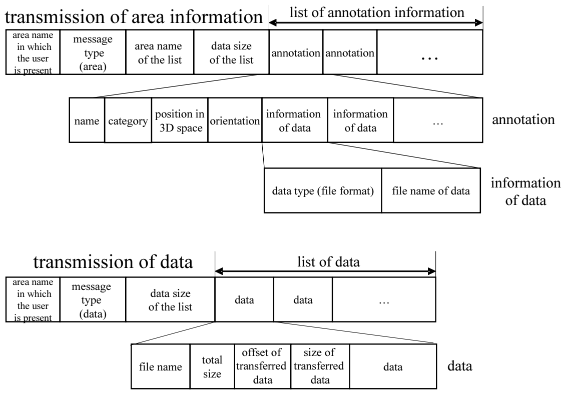


Figure 5. Message format for annotation transmission

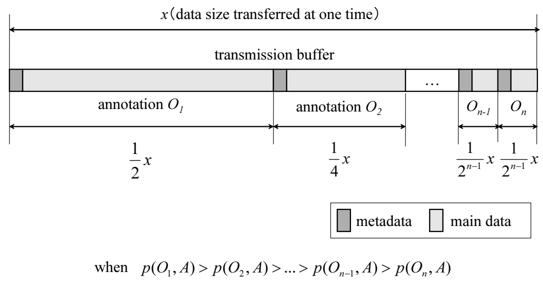


Figure 6. Overview of weighted transmission

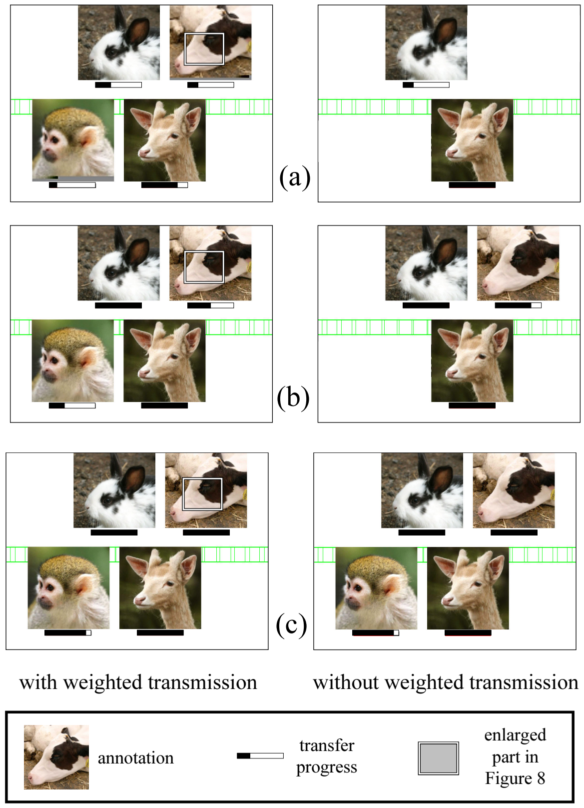


Figure 7. Weighted transmission 1

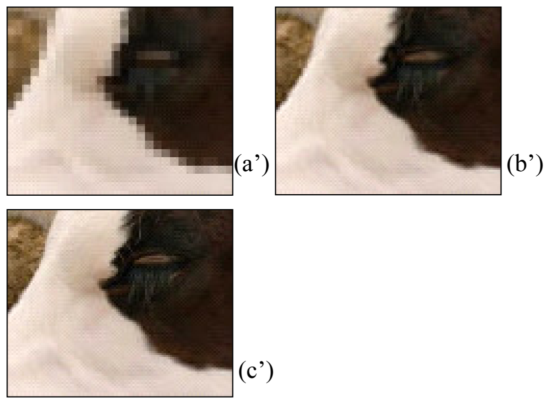


Figure 8. Weighted transmission 2

format. The priority of annotation is ordered as a giraffe, a rabbit, a cow, and a monkey. Figure 7 (a), (b), (c) show the results after 60KB, 120KB, and 180KB transmission. Whereas the images are transferred in the order of priority in case without weighted transmission, the progress bar of the images of the rabbit, the cow, and the monkey gain simultaneously from (a) to (b) and from (b) to (c) in Figure 7 and multiple image data are transferred simultaneously. In addition, images are rendered from the roughest state of progressive jpeg and in detail gradually from (a') to (b') and from (b') to (c') in Figure 8, which shows close-up images of (a), (b), and (c) in Figure 7.

4. Experiment

We performed evaluation experiments to prove the effectiveness of the proposed priority evaluation technique. We prepared two methods to comparison. One (hereinafter called the distance method) uses only $d(O, A)$ and gives higher priority to annotations that are closer to ther user, and the other (hereinafter called the angle method) uses only $\theta(O, A)$ and assigns higher priority to annotation that give smaller $\theta(O, A)$.

4.1. Assumed Wearable AR System

In these experiments, we targeted a typical wearable AR system, which is composed of a set of off-the-shelf hardware devices; a GPS and a gyro sensor for position and orientation tracking, a 3G data transmission device for networking, and a standard HMD for displaying information. According to typical specification of these devices, the horizontal and vertical field of views of the AR client are set to 40 and 30 degrees, respectively, and the network bandwidth between the server and the client is set to 200 kbps.

4.2. Experiment 1

Figure 9 shows the simulation environment of the first experiment. The virtual user stands at the center of 125 m x 125 m room in which 200 annotations are placed randomly. The simulation was performed for two cases. In one case, the virtual user's viewing direction is fixed, and in the other case, the viewing direction rotates horizontally at 360 degrees per second. The server process and the client process were ran concurrently on the same Inspiron8100 (Pentium III 1.2GHz, 256MB RAM), and annotation information managed by the server process was transferred to the client process at 200kbps. The data of each annotation is a progressive jpeg image (approximately 50KB) and all of the images are rendered as 0.8 m^2 square planes in 3D space.

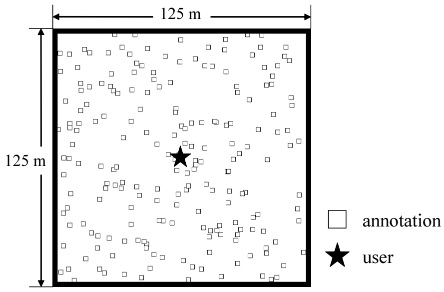
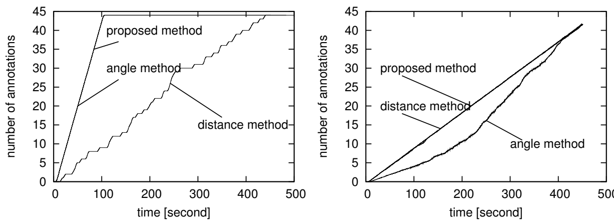


Figure 9. Overview of experiment 1



fixed viewing direction　　rounded viewing direction

Figure 10. Result of exp 1: Number of transferred annotations

Figure 10 shows the results of Experiment 1. The x axis is the time, and the y axis is the number of annotation for which the data was transferred completely and could be rendered at the client. When the viewing direction was fixed, the angle method required only 100 seconds to complete transmission of annotation in the user's view, whereas the distance method required 450 seconds, which sends an annotation close to the user, even if it is out of the user's view. When the viewing direction rotates, the user's view changes constantly and the angle method requires more time than the distance method, which doesn't change the priority of each annotation in this experiment. Using the proposed technique and setting the constants as $D(A) = 100, \pi(A) = 180, W(A) = 360$, and $S(A) = 0.5$, in case of a fixed viewing direction, the number of annotation is the same as for the angle method, and in case of a rotating viewing direction, the number of annotation is the same as for the distance method. In any cases, the performance result of the proposed method is indistinguishable from that of the better method among two existing methods. Therefore, we find that adequate priority evaluation was performed in the proposed method, depending on the user's action pattern.

If we use the number of transferred annotations as the index, annotations that are close to the user and are rendered large, and annotations that are far from the user and are rendered small have the same weight. However, annotations that are rendered larger should be more significant. Therefore, we introduced the "satisfaction rate" as a new index to consider how large each annotation should be rendered in the user's view. In this experiment, all of the annotation will be renderad in the same size in 3D space, closer annotations to the user will be rendered larger. Then, the satisfaction rate $sat(A, t)$ of the user A at time t is expressed as:

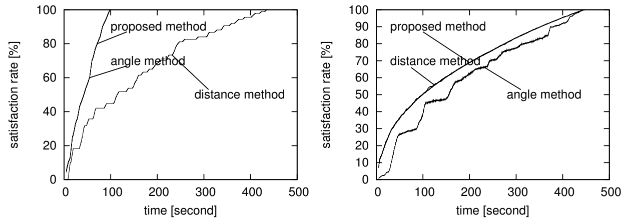
$$sat(A,t) = \frac{\sum_{O \in viewable(t)} \frac{1}{d(O,A)}}{\sum_{O \in should\_be\_viewable(t)} \frac{1}{d(O,A)}} \qquad (3)$$

The image data used in this experiment is progressive jpeg format, and the images can be rendered using approximately 20% of the total data. Then, $viewable(t)$ is defined as the set of annotations that within view of the user when 20% of the data have been transferred, and $should\_be\_viewable(t)$ is defined as a set of annotations within view of the user. Equation (3) can give a higher satisfaction rate when many closer annotation to the user are displayed in the user's view.

Figure 11 shows the change in the satisfaction rate with time. The angle method gave higher satisfaction rate when the viewing direction was fixed, and the distance method gave a higher satisfaction rate when the viewing direction was rotated. The proposed method gave as high satisfaction rate as for the angle method when the viewing direction was fixed, and for the distance method when the viewing direction was rotated. In any cases, the performance result of the proposed method is indistinguishable from that of the better method among two existing methods.

4.3. Experiment 2

Figure 12 shows the simulation environment of the second experiment. The virtual user follows the green line and twice looks 90 degrees to the left and right when the user



fixed viewing direction rounded viewing direction

Figure 11. Result of experiment 1: Satisfaction rate

arrives at certain specified points. The velocity of the user is 4 km/h. We measured the angular velocity while a real user was looking around in a real environment and used this value as the angular velocity of the virtual user while looking around in the simulation. The server sends annotation information to the client at 200 kbps. As in Experiment 1, the data of each annotation is a progressive jpeg image (approximately 50 KB), and all of the images are rendered as 0.8 m^2 square planes in 3D space.

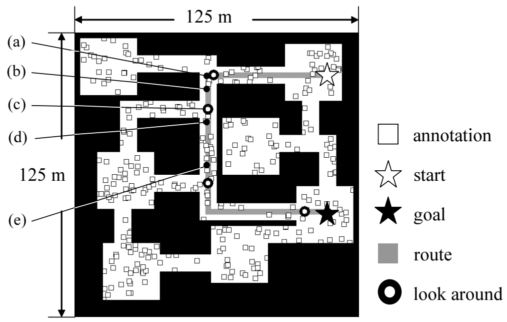


Figure 12. Overview of experiment 2

Figure 13 shows the results of the experiment. We performed the steepest descent method to determine the combination of constant parameters that gives the best result for the proposed method. A total of 50 random combinations of parameters were applied to the steepest descent method, and the best result was selected from the 50 results, which gives the 50 local best. The range of random numbers used to initialize each parameter was $10 \le D(A) \le 300$, $10 \le \pi(A) \le 180$ and $10 \le W(A) \le 360$. The fixed value of 1 was used for $H(A)$ and 0.5 was used for $S(A)$.

Although the angle method gives a higher satisfaction rate than the distance method during most of the experiment, the virtual user turned when $t = 75$ (Figure 12, 13(a)), and most of annotations that were in the user's view until this moment went out of sight. As a result, the satisfaction rate of the angle method decreased and became lower than that of the distance method, which is tolerant of the user's head when $t = 80$ (Figure 12, 13(b)).

The satisfaction rate of the angle method had been kept

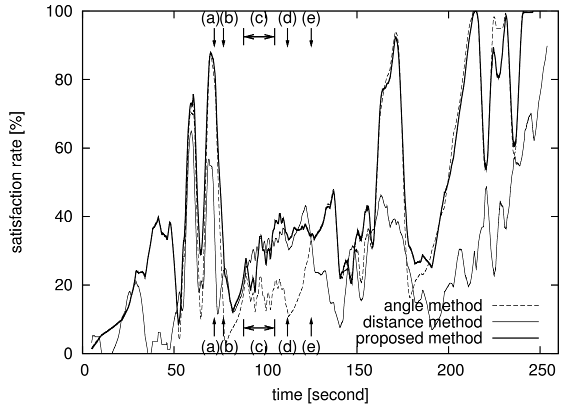


Figure 13. Result of experiment 2

low for 20 seconds (Figure 12, 13(c)) because of the user's head bobbing motion. It became higher depending on the user's straight movement after $t = 110$ (Figure 12, 13(d)), and the priority of the angle method became higher than that of the distance method when $t = 125$ (Figure 12, 13(e)). Although the user turned and looked around at points other than those described above, the satisfaction rate of the distance method did not exceed that of the angle method because the user went through annotations one after another, and the satisfaction rate of the distance method decreased with each annotation. Although the satisfaction rate of the proposed method is as high as that of the angle method during most of the experiment, it is equal to that of the distance method when $80 \le t \le 120$. The average satisfaction rate of the angle method is 42.46%, and the average satisfaction rate of the distance method is 27.51%. The average satisfaction rate of the proposed method is 45.84%. Thus, we confirmed that the proposed method can give a higher satisfaction rate.

We examined various combination of parameters generated at random to find the best combination with the steepest descent method. The average of the average satisfaction rate of all combinations is 41.89%, and the average of the average satisfaction rate of the 50 local best is 43.04%. Therefore, we find that although the best combination is necessary in order to obtain the best performatnce using the proposed method, the performance will not be decreased if a non-best combination is used, and so the proposed method can be used robustly.

5. Conclusion

In the present paper, we presented a data structure for efficient filtering of annotations, a dynamic priority control technique using the movement of the user's viewing direction, and divided and weighted transmission to reflect the results of priority evaluation rapidly. The present study considers the characteristics of a wearable AR system. Annotations handled by the proposed method belong to one of areas that have information of the real environment, and the proposed system manages annotations hierarchically using a tree structure. The server filters annotations using this information and the policy decided by each user. The priority control technique considers that the user's head bobbing varies from low to high in the wearable AR system and changes the priority of each annotation evaluated from the distance between the user and annotations and the angle between the user's viewing direction and the line from the user to the annotation, depending on the angular velocity of the user's viewing direction. Furthermore, we proposed divided transmission and weighted transmission to immidiately adapt to the change in the order of the priority. We confirmed the effectiveness of the priority control technique via two experiments.

At present, the proposed data structure cannot handle area information when an area in the category "Area" belongs to two administrative areas, such as "Town" or "City". Such areas should therefore be divided to subareas for each administrative area, and the subdivided areas should be considered as one large area by connecting the subdivided with 0-distance links. In looking for annotations, these links must be traversed, and the traversed areas and their children are the target for the search, but, in order to avoid confusion, their parents are not a target for the search. Implementation and evaluation of improved database, implementation of an actual networked wearable AR system, and experiments to examine the effectiveness of the proposed system are areas for future consideration.

References

[1] S. Feiner, B. MacIntyre, and T.Hollerer, "A Touring Machine: Prototyping 3D Mobile Augmented Reality System for Exploring the Urban Environment," Proc. of the 1st International Symposium on Wearable Computers (ISWC97), pp.208-217, 1997. 1

[2] W. Piekarski and B. H. Thomas, "An Object-Oriented Software Architecture for 3D Mixed Reality Applications," Proc. of the 2nd International Symposium on Mixed and Augmented Reality (ISMAR '03), pp.247-256, 2003. 1

[3] E. Frecon and M. Stenius, "DIVE: A Scaleable Network Architecture For Distributed Virtual Environments," Distributed Systems Engineering Journal, Vol.5, No.3, pp.91-100, 1998. 1

[4] J. Calvin, A. Dickens, B. Gaines, P. Metzger, D. Miller and D. Owen, "The SIMNET virtual world architecture," Proc. of IEEE VRAIS '93, pp.450-455, Sep 1993. 1

[5] T. A. Funkhouser, "RING: A Client-Server System for Multi-User Virtual Environments," Proc. of Symposium on Interactive 3D Graphics, pp.85-92, 1995. 2

[6] M. Hosseini, S. Pettifer, N. D. Georganas, "Visibility-based Interest Management in Collaborative Virtual Environments," 2004. 2

[7] C. Greenhalgh and S. Benford, "MASSIVE: a collaborative virtual environment for teleconferencing," ACM Transactions on Computer-Human Interaction, Vol.2, Issue.3, pp.239-261, 1995. 2

[8] Jauvane C. de Oliveira and N. D. Georganas, "VELVET: An Adaptive Hybrid Architecture for VEry Large Virtual EnvironmenTs," Proc. of IEEE International Conference on Communications, No.1, pp.2491-2495, April 2002. 2

[9] S. Han, M. Lim, and D. Lee, "Scalable Interest Management Using Interest Group Based Filtering For Large Networked Virtual Environments," Proc. of the ACM Symposium on Virtual Reality Software and Technology (VRST 00), pp.103-108, 2000. 2

[10] A. K. Beeharee, A. J. West, R. Hubbold, "Visual Attention Based Information Culling for Distributed Virtual Environments," Proc. of the ACM symposium on Virtual reality software and technology (VRST '03), pp.213-222, Oct 2003. 2

[11] S. Park, D. Lee, M. Lim, and C. Yu, "Scalable Data Management Using User-Based Caching and Prefetching in Distributed Virtual Environments," Proc. of the ACM Symposium on Virtual Reality Software and Technology (VRST 01), pp.121-126, 2001. 2

[12] A. Chan, R. W. H. Lau, B. Ng, "A Hybrid Motion Prediction Method for Caching and Prefetching in Distributed Virtual Environments," Proc. of the ACM symposium on Virtual reality software and technology (VRST '01), pp.135-142, 2001. 2

[13] T. Y. Li and W. H. Hsu, "A data management scheme for effective walkthrough in large-scale virtual environments," The Visual Computer, Vol. 20, Issue. 10, pp.624-634, 2004. 2

[14] S. Julier, M. Lanzagorta, Y. Baillot, L. Rosenblum, S. Feiner, T. Hollerer and S. Sestito, "Information Filtering for Mobile Augmented Reality," Proc. of International Symposium on Augmented Reality (ISAR '00), pp.3-11, 2000. 2

[15] M. R. Macedonia, M. J. Zyda, D. R. Pratt, P. T. Barham, and S. Zeswitz, "NPSNET: A Network Software Architecture For Large Scale Virtual Environments," MIT Press Presence, Vol.3, No.4, pp.265-287, 1994. 2

[16] T. H. Kolbe and G. Groger, "Towards Unified 3D City Models," Proc. of the ISPRS Comm. IV Joint Workshop on Challenges in Geospatial Analysis, Integration and Visualiation II, 2003. 2

[17] K. Kitabayashi, H. Kanoh, R. Kijima, "An analysis of head movement in the daily life," Proc. of the 8th VRSJ Annual Conference, 1B4-7, 2003. (in Japanese) 4

ICAT 2008
Dec. 1-3, Yokohama, Japan
ISSN: 1345-1278