ICAT 2003
December 3-5, Tokyo, JAPAN

Distance Learning of Chang'an in an Immersive Environment

Miho Kobayashi   Kei Utsugi   Masami Yamasaki   Haruo Takeda

Systems Development Laboratory, Hitachi, Ltd.
1099 Ohzenji, Asao-ku, Kawasaki, 215-0013, Japan
{ *mkoba, utsugi, yamasaki, takeda* }*@sdl.hitachi.co.jp*

Abstract

We have made a computer graphics model of *Chang'an* City when it was the capital of China during the Tang Dynasty (7-10th century). The real-time rendered images are projected onto a hemispherical screen by 6 projectors after being geometrically transformed, and the brightness and color adjusted to produce one seamless image on the screen. The transformation is executed on our proprietary image processors. The walkthrough systems are linked between Japan and China through the internet. An authority on Chang'an studies based in China, walks through the virtual world while explaining for students in Japan. The image in the immersive environment is synchronized to the motion of the guide. To compensate for the delays or for packet losses over the internet, the optimal motion of the student viewpoint is estimated based on received sparse information, educational knowledge, and heuristics that predict the guide's next destination.

Key words: Hemispherical Screen, Immersive Projection Display, Distance Learning, Virtual City

1. Introduction

There have been a lot of recent developments of effective applications that show walkthrough content in a virtual space by using *immersive projection displays* (IPD). These display the surround-images on a special screen (cylindrical, spherical, plane, etc)[1][2]. Some content is especially efficient for educational purposes because of the quasi-real experience. Examples of such content are the reconstruction of a now-defunct ancient city or applications that show distant places that people cannot visit easily.

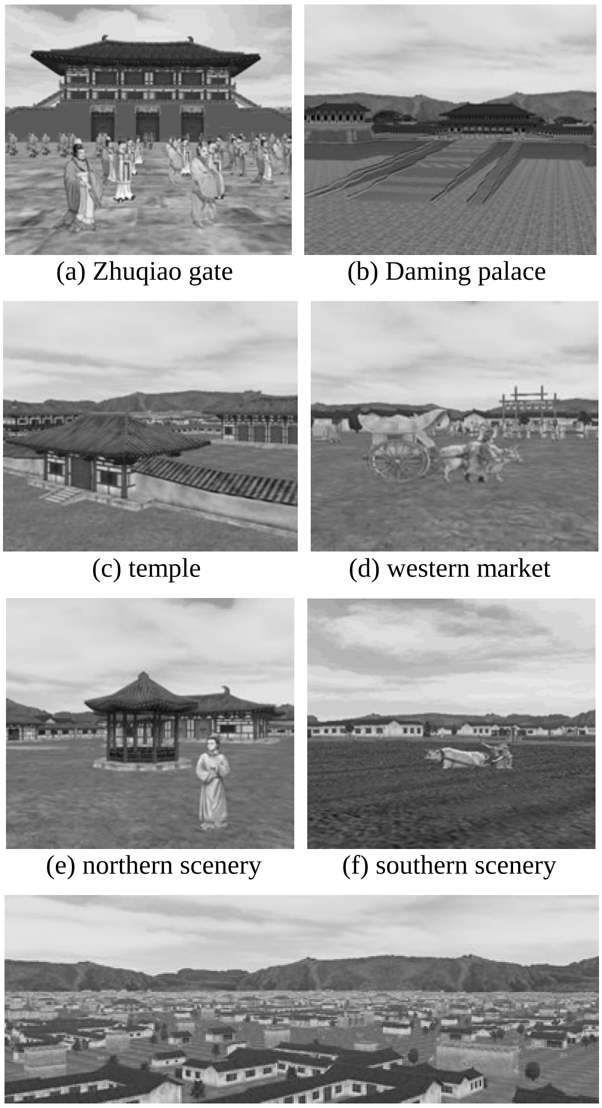
Our study is concerned with distance learning that uses the IPD system in a classroom. Most of the walkthrough content applications allow one user to control the viewpoint in the virtual world and others can only look. So if a class doesn't have enough time, the teacher should control the viewpoint for efficiency. When the teacher uses a city walkthrough as a teaching material for students, it can provide more information than a book or video because of its 3-dimensionality and interactivity. To satisfy a student's intellectual appetite, the teacher should have rich knowledge of the place. But it is difficult for teachers to know about all the available places well. Therefore, the purpose of our study is to let the foremost authority of the city take the students on the tour and explain about it in detail. The guide doesn't need to come to the classroom and can even explain things from overseas.

In this distance learning system, we assume that a control signal is sent by a normal internet connection. Delays and packet loss often occur when connected abroad because of the great deal of traffic volume and network failures. To achieve stress-free communication within a shared virtual space, it is important that all the observers should have the same viewpoint [3]. We estimated a mechanism that will enable the guide to conduct classes smoothly even form regions with low level networking infrastructures. We limited the control information that needs to be sent and received to a small size. The application was projected onto an immersive environment worked smoothly using only limited information.

2. Modeling of the Chang'an city

We studied various walkthrough application [4][5]. We developed a virtual model of the 8th-century Japanese capital *Nara*. The content allows users to watch a lecture by an authority based in a different location in Japan. We then made a model of Chang'an (Fig. 1, 2), which allows for a lecture from China. Chang'an is said to be the basis from which Nara, which we modeled, was constructed. Studying the similarities or differences of a number of cities adds educational value compared to studying only one city [6].

Chang'an was the center of the world around that time. The population is estimated to have been more than one million, and the city had an area of 9.7km x 8.6km. Objects (buildings, trees, and people) are too numerous to arrange precisely one by one. This model is for historical education; therefore we made some rules concerning arrangement so we can be as faithful to historical facts as possible. At first we made some typical object arrangement patterns of various blocks based on the following examples of rules.



(a) Zhuqiao gate (b) Daming palace

(c) temple (d) western market

(e) northern scenery (f) southern scenery

(g) view by jumping

Fig. 1 Screen shots of Chang’an

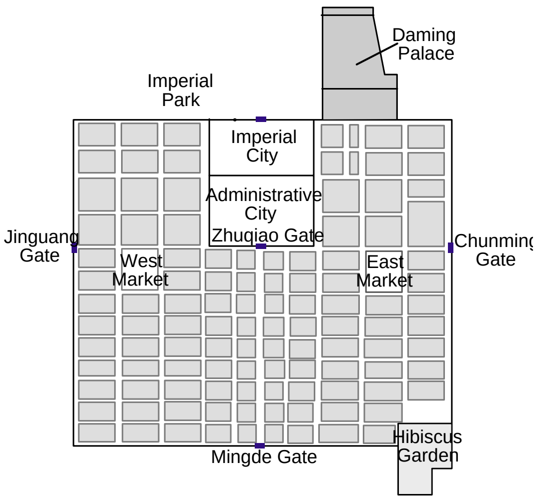


Fig. 2 Map of Chang’an

- East Asians, such as Chinese, Koreans, and Japanese, lived on the east side of the city. While Central and Western Asians, such as Persian, and Arabs, lived on the west side.

- Rich people including aristocrats lived on the northern side of the city (Fig. 1(e)) in generally large homes and so the cityscape was fairly prosperous. However south side (Fig. 1(f)) was the home for common people and poor people who lived in narrow houses. Here there were also and there are also fields and wasteland.

- The economic zone was limited inside the two blocks (“Eastern market” and “Western marked (Fig. 1(d))”) and activated by a policy of the Tang Dynasty.

We arranged the various blocks around the city. We also added some famous landmarks (Fig. 1 (a)(b)) for educational use, as described in Section 3.4. As a result, the students can get a lot of knowledge by walking around the city.

3. Remote guide of VR contents

3.1 System architecture

Fig.3 is the local environment architecture of our IPD system for students to watch a guided tour of Chang’an at a remote environment. We used a hemispherical screen. Given a viewpoint, the Chang’an model is rendered in real-time by 6 *Render PC*s. The rendered images are projected onto the screen by 6 projectors after being geometrically transformed, with the brightness and color adjusted by 6 graphic accelerators to produce one seamless, complete image on the screen. A *Control PC* controls the I/O and synchronicity of the 6 Render PCs. A user controls the movement acceleration, sets rotation speed, can jump up, can stop, turn a map on/off, and select a mode (Master or Slave), all with a joystick or keyboard. The Control PC communicates to the other Control PC in the remote environment via a *Server PC*. Since the Control PC can run the content alone, the remote environment can use either an IPD or only the Control PC with display. The Server PC pairs up one Control PC with another, and manages the data passing between the pair. It can also handle multi-pairing. Each Control PC can get its partner’s information by connecting to the Server PC. This contains information about the position, angle, speed and mode. It then sends its own information to the Server PC. In this paper, the Control PC in Master mode is called the Master, and same way is applied to Slave, too.

A user selects the mode, Master or Slave, with the Control PC. When both of the pair are in the same mode, each Control PC can freely move the view point in the virtual world. However when one is the Master and the other is the Slave, the Slave’s viewpoint follows the Master’s as if they ware connected by a piece of elastic.

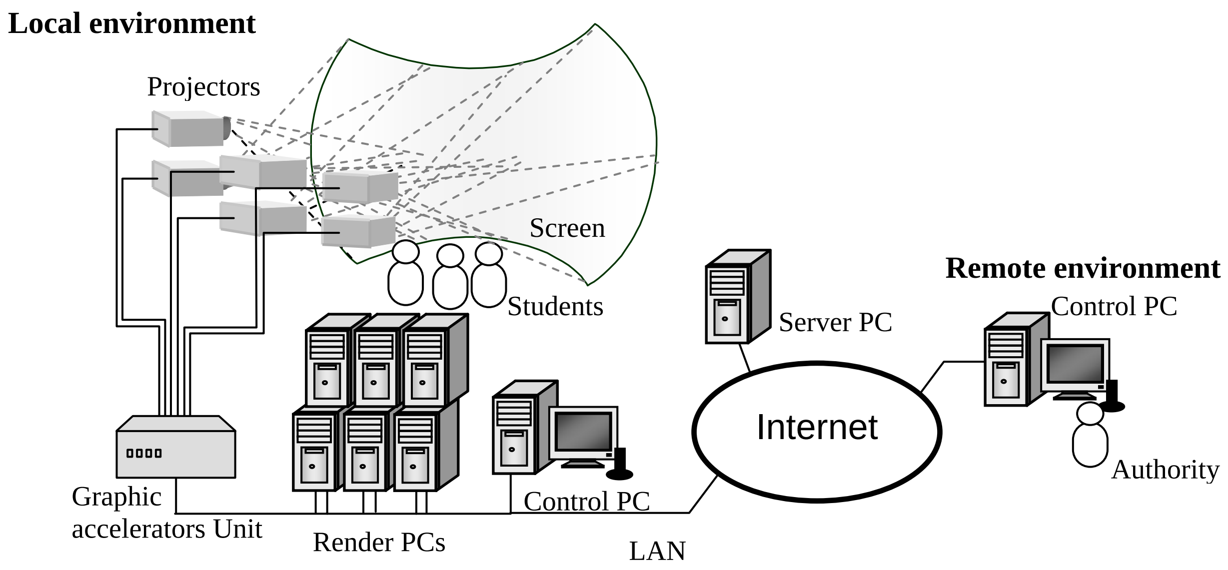


Fig. 3 Example of system architecture

For distance learning, we assume the following - the authority in the remote environment selects the Master, and the students or a local teacher select the Slave. The Slave moves towards the Master when it is away from the Master.

3.2 Problem of Distant Guidance

For the application using *Nara* [4][5], we used an *ISDN line* to telecommunicate between the classroom and Nara city, where the authority acting as guide was staying (the distance between these two locations was more than 300 km). There was a slight delay so we didn't have to interpolate any data. However, when using Chang'an, we assumed that the guide lives in China. Despite the network infrastructure in China is now developing rapidly, there are still many delays and packet losses in telecommunication between China and Japan over the internet.

When the Master's information doesn't arrive at the Slave because of a network problem, if the velocity of the Slave is simply function due to the distance, the sparse Master's information causes unnatural acceleration-deceleration. This will lose the walkthrough presence in the virtual world, and can make the students feel "Cyber sickness" [7]. Therefore, our main purpose is to move the Slave's viewpoint to follow the Master even if the Master's information is missing. We started by estimating the Master's destination on the assumption that the guide will visit some landmarks to give historical information about the city.

3.3 Notations of the Master and the Slave

Fig. 4 shows the notation for viewpoints in the virtual world. We look at the case when the mode of the local Control PC is the Slave and the remote one is the Master. Arrows are vectors of the velocity. θ_M and θ_S are the angles showing the direction of velocity. We assumed the direction of the Master's viewpoint is the same as the direction of the Master's movement. This means that the Master moves in a forward direction. In contrast, the Slave can look in any direction without recourse to its direction of movement. If the slave can't do that, then it has a different view to the Master, as in Fig. 5. Therefore, the Slave should rotate to have almost the same view as the Master when it comes close to the Master but it should still continue to move towards the Master. In Fig. 4, the direction of the Slave's viewpoint is omitted. This will be mentioned in Section 3.4.

$d(M, S)$ is the distance between the Master and Slave, and $\theta_{M,S}$ is the angle from the Slave to the Master as calculated in the following equations.

$$d(M,S) = \sqrt{(x_M - x_S)^2 + (y_M - y_S)^2} \quad (1)$$

$$\theta_{\mathrm{M,S}} = \tan^{-1}\left(\frac{y_M - y_S}{x_M - x_S}\right) \quad (2)$$

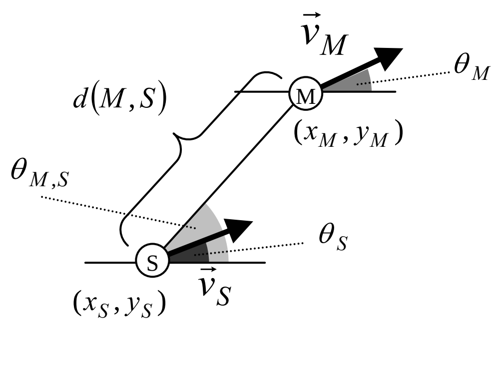


Fig. 4 Notation of the Master and the Slave

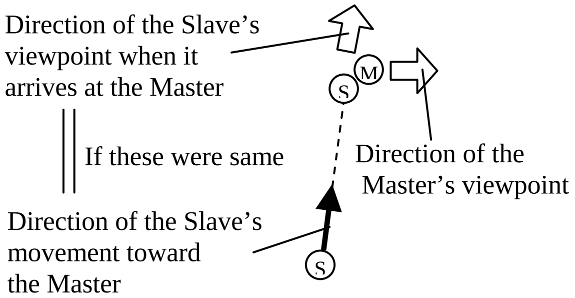


Fig. 5 Problem with a disagreement of the direction

3.4 Estimation of the Master movement

When the Slave knows the Master's information, it can move towards the Master. But when it loses the Master, it needs to estimate the Master's movement. To estimate the next angle of the Master, $\theta_{M\text{-}E}$, we use the latest Master's information and some educational heuristics.

First, we make a list of some landmarks. For each landmark i, the position (x_i, y_i) and a number n_i which means how many times the landmark i was visited, are stored. $\theta_{M\text{-}E}$ is described as equation (3).

$$\theta_{M\_E} = \theta_M + \delta \sum_{i \in Z} \frac{n_i}{n_{all}} (\theta_i - \theta_M) \qquad (3)$$

n_{all} is the sum of all landmark's n. Z is the set of the landmarks in which the angle between the angle to the landmark and $\theta_{M.}$ is smaller then $\pi/2$ (in the gray zone marked Z in Fig.6). δ is the constant used to decide the contribution of the heuristics. With this equation it is assumed that the Master will change its angle towards the "most often visited landmark". However, it is also possible that the Master can move towards other landmarks. This means that the coefficient n_i works as the weight of movement towards landmark i. For example in Fig. 6, $\theta_{M.\text{-}E}$ is between θ_M and the angle with landmark 2. Once the application starts, the Master increments the visited number of the landmark if the Master visits the area around it. And once visited, this landmark is excluded from the set Z until next time so that it doesn't return to the same place many times.

After estimating angle $\theta_{M.\text{-}E}$, the next position of the Master is calculated with this angle and the latest speed of the Master. In the next frame to be rendered, if the Slave cannot receive the Master's information again, the Slave re-estimates the Master's movement, using the latest $\theta_{M.\text{-}E}$ as the new θ_M.

If the Slave can receive the Master's information and the latest estimation is not valid, the Slave will change its objective point to the new one. To make this change smooth, the acceleration is capped. We describe the Slave's movement in the following sections.

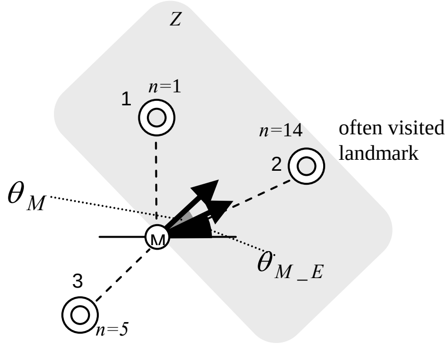


Fig. 6 Estimation of the Master angle

3.5 Movement of the Slave

In this section we describe the movement of the Slave. The direction its viewpoints look towards is omitted. The Slave doesn't have to stay at the same place as the Master, but it should follow the Master. The Slave is concerned with 3 factors, the latest Master's position, the speed, and the angle. However, when the Slave cannot get the Master's information, it estimates the next position and angle of the Master and uses that as the latest Master's information. It continues to do this until it can get real information.

The amplitude of the Slave's velocity is calculated depending on the distance $d(M, S)$. The further the Master, the faster the Slave moves. And it stops when it arrives at almost the same position as the Master.

$$\vec{v}_S = |\vec{v}_S| \cdot \vec{e} \qquad (4)$$

$$|\vec{v}_S| = d(M,S) \cdot \exp\{-\alpha \cdot d(M,S)\} \qquad (5)$$

$$\vec{e} = (\cos\theta_{\mathrm{M,S}}, \sin\theta_{M,S}, 0) \qquad (6)$$

α is the parameter which affects the rate of change. It should be adequately changed when sudden changes in the Master's position occur.

However the velocity change is not smooth if we only use this approach. Therefore, we apply limits to the change of both $|\vec{v}_S|$ and $\theta_{M,S}$. This process also shows good results for when the Slave's estimation is not valid. In this case, the Slave has to correct its course smoothly.

3.6 Direction of Slave's Viewpoint

We notate the angle of the Slave's viewpoint as φ_S to distinguish it from θ_{S}. The Slave moves in the direction of angle $\theta_{\mathrm{M,S}}$, but its view is the direction of angle φ_S. At first, φ_S changes form its value to $\theta_{M,S}$, if they are different. Then, φ_S doesn't change its value until the distance between the Master and itself is adequately small. If the Slave comes close to the Master, φ_S

changes to θ_M . This rotation means that at first the Slave faces in the direction in which the Slave is moving, and when it comes close to the Master, it tries to face in the same direction as the Master. The reason why the rotation is limited to only the start and end of the movement is to avoid moving with rotation, as doing this causes an uncomfortable feeling for observers, similar to being in "a coffee cup" ride in an amusement park.

In summery, we described the estimation of the Master's movement by referring to educational heuristics and we outlined how the Slave can smoothly follow the Master.

4. Cost array of the entire map

As well as estimating the Slave's movement, we add a cost array for the entire map for both the Master and Slave. We attached a lattice over the map. Each lattice point has an 8 bit "cost" value p. Some lattices to which the viewpoint should go have lower costs than others. Our program doesn't judge collisions with other objects. So the cost array is used to avoid some objects, to walk straight along a street, to look in at some interesting place.

This process is implemented after the angle of movement θ is decided from either user input or the Master's information. The closest lattice point is first selected according to the viewpoint position (the center of Fig. 7). Then we select a lattice point j, which is a neighbor of the closest lattice point, and an angle between θ and θ_j that is smaller than $\pi/2$. Perhaps 4 neighbors are selected in this way (the black lattice points in Fig. 7. The grayed zone is the 8 neighbors.). Next, we select the neighbor with the lowest cost of the 4 neighbors, and adjust the angle θ to rotate towards θ_j a little in equation (8).

$$\theta' = \theta + \varepsilon(\theta_j - \theta) \qquad (7)$$

ε is the constant used to adjust the degree. This process helps to achieve smooth movement even if the operator is inexperienced in using the input device.

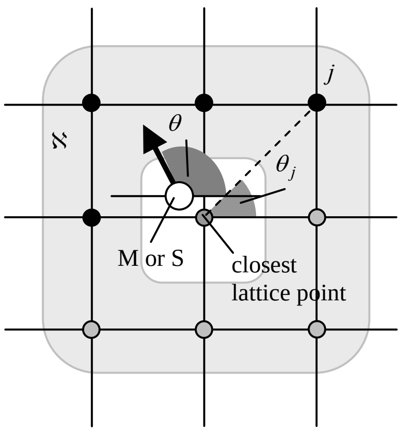


Fig. 7 Angle adjustment

5. Implementation

Fig. 8 shows a brief flowchart of part of our program. Our estimation process is needed when the local Control PC is the Slave and the remote one is the Master, and the Slave cannot get the Master's information.

We tested distance learning with a demonstration between China and Japan. A ping timeout of about 10% occurred in the early morning, but it changed to more than 50% in the afternoon. Our line was a fiber-optic network (max 100M bps) and the guide's was ADSL (max 2 Mbps), which was from a shared network in an office building. We used an international call for voice communication, such as to give explanations and Q & A.

The movement of the Slave was smoother than when

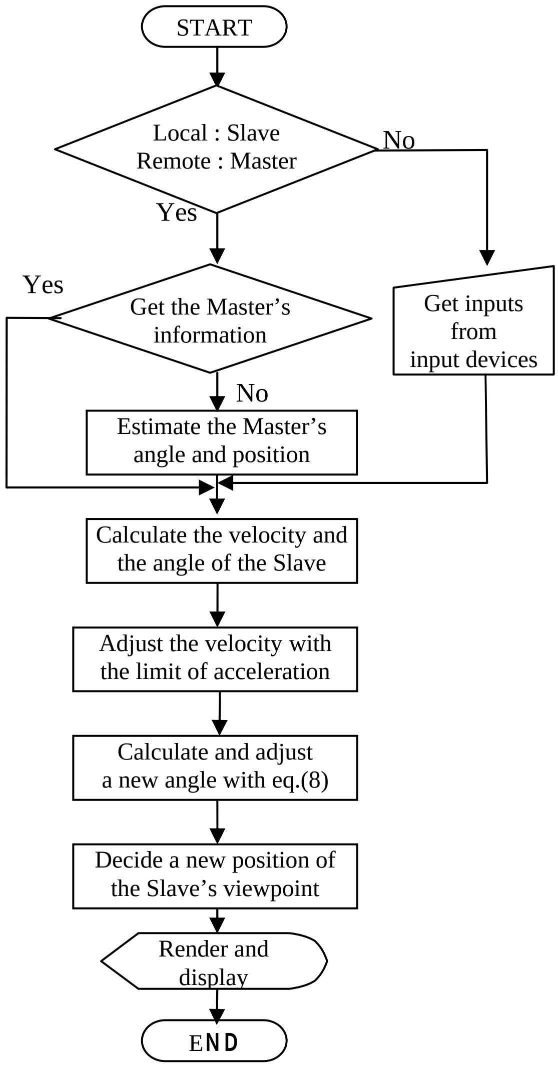


Fig. 8 Flowchart of one step of rendering

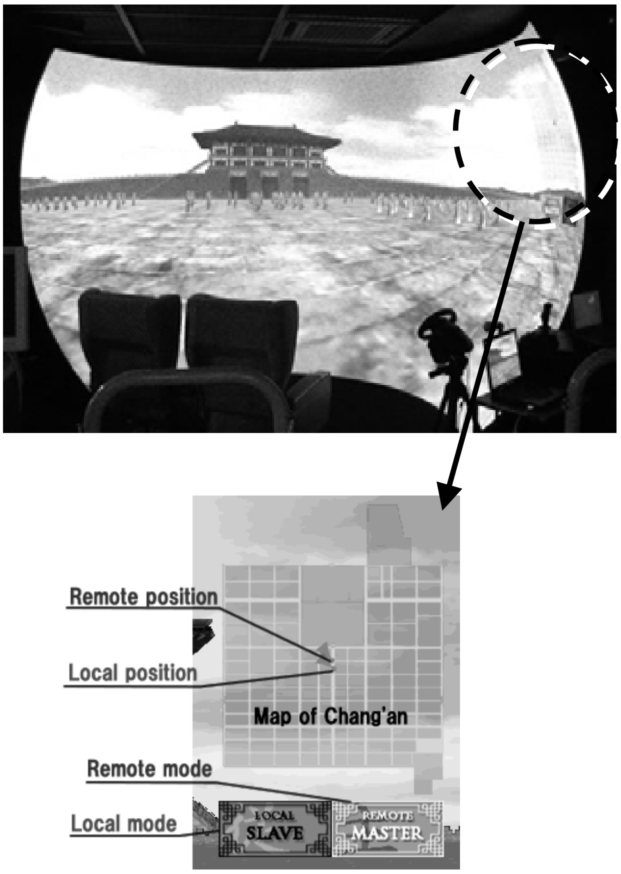


Fig. 9 Displayed Chang'an and its information map on the hemispherical screen

there was no motion estimation. And most "students" didn't notice that the Slave couldn't often get the Master's information. We will also show some animation of the results of the simulations under various conditions in our presentation.

6. Conclusion

We reported three major points in this paper (1) the CG creation of the entire Chang'an city (approximately 10 x 10 km) from an historical perspective, (2) the immersive environment with 6 projectors, 6 PCs (PC cluster), 6 graphic accelerators, and proprietary image processors to produce one seamless video image on the screen, and (3) synchronization between the displays over a distance over the internet. The main subject was how to estimate the Slave motion even when it gets too little information from the Master. If the network environment is improved, we can communicate with more value-added information, such as with Video avatar [8].

Acknowledgements

This study was supported by the Telecommunications Advancement Organization of Japan (TAO).

References

1. C. Cruz-Neira: "Surround-screen projection–based virtual reality: The design and implementation of the CAVE," *proceedings of ACM SIGGRAPH '93*, pp.135-142 (1993).
2. H. Takeda, S. Kiyohara, K. Chihara, H. Kawase, Y.Matsuda, and M. Yamasaki: "Multi screen environment with a motion base," *Lecture Notes in Artificial Intelligence,* Vol.1834, pp.303-312 (2000).
3. H. Ohnishi, K, Mochizuki, K. Nagaoka; "Effect of Misalignment on Distance Collaboration," *The journal of the Institute of Electronics and Communication Engineers of Japan*, vol. J83-D-I, (2000) (in Japanese).
4. K. Utsugi, T. Moriya, and H. Takeda: ''Digital Heijyokyo, A walkthrough model of a Japanese ancient city," *VSMM2001 (7th International Conference on Virtual Systems and Multimedia)*, pp.163-164 (2001).
5. H. Takeda, K. Utsugi, T. Moriya, K. Chihara, and N. Yokoya: ''Nara in the 8th century by video-based virtual reality," *KES2001, Frontiers in Artificial Intelligence and Applications,* vol.69, pp.1621-1625 (2001)
6. Taisei Corporation "Ancient Civilization City-State Virtual Trip," http://www.taisei-kodaitoshi.com/
7. C. Nakagawa, M. Ohsuga: "The present situation of the studies in VE-Sickness and its close field," *The journal of VRSJ*, vol. 3, pp.31-39 (1998) (in Japanese).
8. K. Utsugi, F. Beniyama, H. Namai, T. Moriya, and H. Takeda: "A High-Resolution Avatar System using Partial Compositions," *ICAT2002*, pp.65-71 (2002)