Second International Workshop
on
Cooperative Distributed Vision

Nov. 5-6, 1998

Kyoto, Japan

Sponsored by

Cooperative Distributed Vision Project
Japan Society for the Promotion of Science

Contents

Imaging a 3D Lecture Room by Interpreting its Dynamic Situation

Michihiko Minoh, Yoshinari KAMEDA

Center for Information and Multimedia Studies, Kyoto University

Abstract We envision a new computer supported environment named *information media environment*. Users can watch what happens in a fixed real space where people get together to do something through raw/synthesized video images in real-time with this environment.

The information media environment tries to adopt each user's favorite way of imaging the scene. Currently we approach this environment in two ways. The first approach is making the synchronized virtual space that permits users to browse the scene from any viewpoint. The other is making active video imaging system which controls active cameras according to the dynamic situation of the scene.

1 Introduction

People usually gather together and do various activities in a certain fixed space. For instance, there are conferences, amusement events such as sports and concerts, lectures in schools, and business activities. It is technically becoming possible to watch or even participate in such an on-going activity from a remote place thanks to the power-up of computers and speed-up of the transmission lines.

When a person in a remote place participates in such an activity, he/she will enter a kind of *information media environment* . In the target real space , there are discrete objects which do activities. Users not only watch the objects in the real space but also obtain related information about them in the information media environment .

The synchronized virtual space is a space which reflects the changes of the geometric and photometric information of the real space . The changes in the real space are called *dynamic situation* . Since the real space changes in real-time, the synchronized virtual space has to change accordingly. The only difference between the real space and the synchronized virtual space is the time delay which is caused by the transmission and processing of the data in recovering shape and surface information of the objects.

It is not necessary to construct the synchronized virtual space completely to construct the information media environment . Conceptually, consider one axis one end of which is the real space and the other end is the synchronized virtual space (Figure 1). Between these two spaces, there exist many spaces which are mixture of these two spaces. In other words, there is a seamless transition from the real space to the synchronized virtual space.

Video image is considered to be a typical presentation media of the information media environment and is suitable to convey visual information of what happens in the real space. In the near future, it will be possible for the computer to present virtual three dimensional space that looks exactly the same as the real space in geometric and photometric sense with VR equipment.

When the information media environment is presented to the users with the video image, active video cameras are used to observe the real space . The cameras are set to

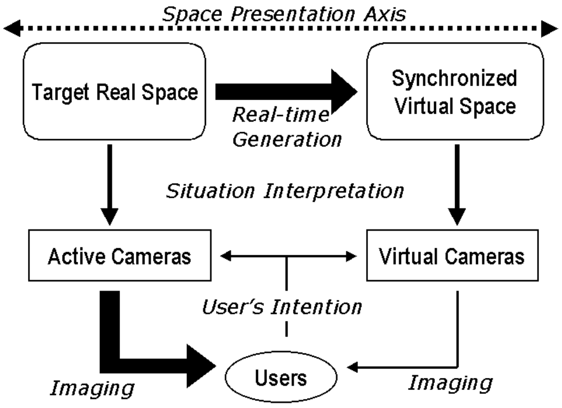


Figure 1: Information Media Environment

surround the real space to sense the activities in the real space and can change its direction and zooming parameter dynamically. One of the interesting topics in our research is how to control the cameras and generate interesting video for the users . We have to consider all the kinds of the spaces between the real space and the synchronized virtual space and determine which space is suitable for imaging. It is important that the information media environment can present various information extracted not only from the real space but also from the internet computer network such as WWW pages. In this sense, a synthesized video image of the real space has to be displayed together with other related information in the image at the same time. To make a relation between an object in the real space and information in the database, it is necessary to recognize the objects.

The view point from which the video image is generated and the related information overlapped on the video image are selected by a user . The generated video image is fully user-adaptive.

The information media environment is not for a user but for multiple users. Therefore, the information media environment have to generate multiple videos each of which serves the request of the individual user . One of the desirable functions is that each user can watch the real space from any camera viewpoint he likes, if he would like to.

For the synchronized virtual space , it is relatively easy to realize the function because we can put virtual cameras wherever we like. However, for the real space , it is difficult because the camera locations are fixed and restricted.

Descriptions of dynamic situation of the real space are helpful for the user to select the video image and related information. The information media environment snatches the dynamic situation and interpret it in order to display appropriate video images to the user . The user enjoys automatic visualization via the information media environment .

In constructing the information media environment , we introduce multi-agent concept. Since the real space is surrounded with eight active cameras, it is preferable to consider camera agents which control the cameras. In some applications, all the camera agents do the same function, and in other applications, several camera agents have different functions from the others. The main problem to construct the information media environment is how to divide the whole function into simple independent agents and how to make them cooperate.

We set a lecture room in the Kyoto university as the target real space for the first implementation of the information media environment . This has two reasons. One is that the activities in the real space are relatively simple. The other is that lecture is one of the most important activities in the university, and there is a strong request to watch a lecture from remote places.

In this paper, the camera control mechanism which can cope with multi-user requests under the constraint of CDV framework is proposed. In section 2, the method of generating the synchronized virtual space is described. The users can move a viewpoint wherever they like in the generated synchronized virtual space . In section 3, we mention our research which shows the information media environment at the opposite side of the space presentation axis from the synchronized virtual space . The camera control principles are discussed under the physical constraints. Section 4 describes experiments and result and we conclude this paper in Section 5.

2 Generation of Synchronized Virtual Space

2.1 Background

With improvement in processing speed of computers and with increase of their storage size, it may come true to synchronize a virtual space in computers with a real 3D space[10]. Our goal here is to construct the synchronized virtual space which displays real-time human activities in the real space (see [1]).

Once the synchronized virtual space is constructed, anyone outside the real space can observe the human activities in the real space from any viewpoint with a little delay.

Slit light projection methods and structured light projection methods achieve real-time 3D reconstruction, but these methods require active sensing which disturbs human activities in the real space . On the contrary, passive vision based approach[8, 9] does not affect the activities. Stereo vision methods achieve real-time 3D reconstruction though they cannot reconstruct backside shapes that cannot be seen by stereo cameras. Therefore, the cameras have to be placed so as to surround the real space. Realistic 3D reconstruction methods[6, 7] have been proposed which use over ten cameras, but their approaches need certain period to reconstruct one scene and are not suitable for real-time applications.

The main problem of 3D reconstruction with such camera surrounding layout is that it requires much calculation time because there are many images at each frame. This problem is resolved by distributed computing in our approach. We reconstruct the real space by preparing one computer for each camera to execute image processing, and other computers to calculate 3D reconstruction. All the computers are connected one another with 100baseT Ethernet and 155Mbps ATM LAN.

We describe the reconstructed space by voxel representation. In our method, we improve throughput by dividing video processing into some stages and forming them as the pipeline processing, and decrease latency by dividing a real 3D space into some subspaces and reconstructing each subspace simultaneously with several distributed computers. We can also control throughput and latency by changing the pipeline formation in the system and satisfy the requirements of the applications.

2.2 3D Reconstruction Method

The reconstruction algorithm has to be suitable for the distributed computing, so that the algorithm has the following two characteristics.

- It is possible to equalize processing time of each process by dividing program and data.
- The amount of communication among processes is not so much.

The viewing frustum method (VFM in short) satisfies these two characteristics.

With VFM, we reconstruct the real space in real-time by generating voxel data from several images taken at the same time. We call the part of the real space which can be imaged by the cameras the *target space* .

2.2.1 Static Object Occupation Subspace (SOOS)

Since our objective is to reconstruct the fixed real space , it is reasonable to have a knowledge of static objects in advance. As the static objects do not change their locations and shapes, we can exclude the subspace where the static objects occupy. We call the subspace as *static object occupation subspace (SOOS)* denoted by $\mathcal{S}$. For example, see Figure 2 where a circle is a dynamic object and a rectangle is a static object. $\mathcal{S}$ corresponds to the rectangle region.

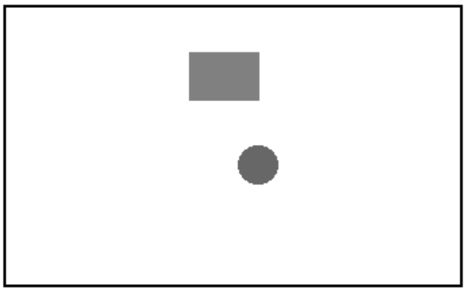


Figure 2: Dynamic Object (Circle) and Static Object (Rectangle)

If we see the target space from the viewpoint of camera i, some subspaces cannot be seen because $\mathcal{S}$ occludes them. We merge these occluded subspaces into $\mathcal{S}$ and call it *static object influence subspace* (SOIS) $\mathcal{S}_i$. Figure 3 shows two $\mathcal{S}_1, \mathcal{S}_2$ for two cameras.

From now on, we concentrate on reconstructing the voxels which represent dynamic objects in the target space , i.e. dynamic situation .

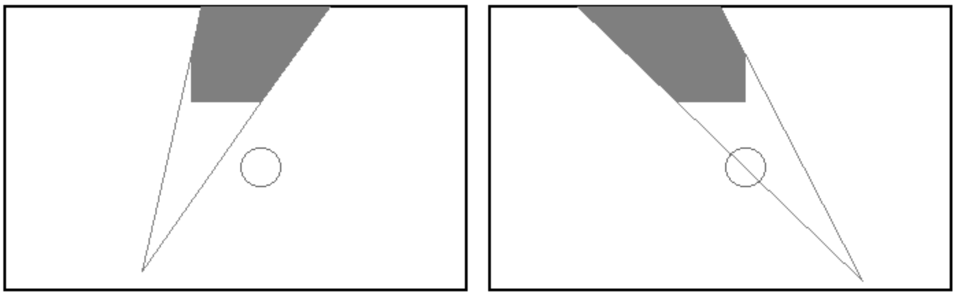


Figure 3: Static Object Influence Subspace $\mathcal{S}_1, \mathcal{S}_2$

2.2.2 3D Reconstruction

When the dynamic objects are imaged by a camera i, they exist within frustums that circumscribe their projected regions on the image and whose apexes are focus point of the camera. We call all the projected regions in the same image together a *dynamic region* D_i and let us denote a subspace consisting of these viewing frustums by $\mathcal{V}_i$. Figure 4 shows two $\mathcal{V}_1, \mathcal{V}_2$ for two cameras.

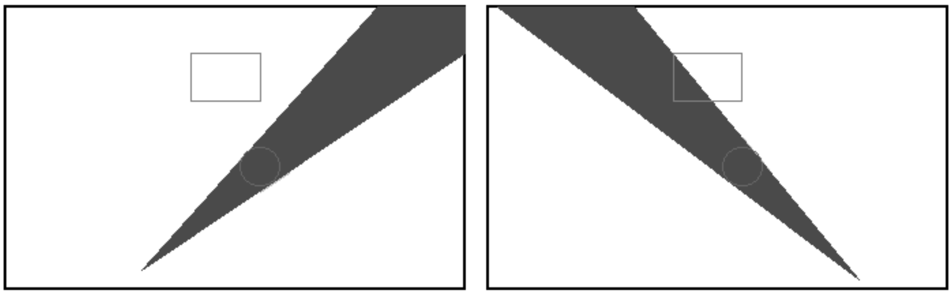


Figure 4: Viewing Frustum $\mathcal{V}_1, \mathcal{V}_2$

As the dynamic objects can exist only outside $\mathcal{S}_i$, we only care a subspace named *existence shadow subspace* (ESS) $\mathcal{U}_i$ defined by Equation (1).

$$\mathcal{U}_i = \mathcal{V}_i \cap \overline{\mathcal{S}_i} \tag{1}$$

The dynamic objects exist somewhere inside $\mathcal{U}_i$. Figure 5 shows two $\mathcal{U}_1, \mathcal{U}_2$.

In the case where the dynamic objects are imaged by n cameras, they exist within the product of all of these frustums. We denote this subspace as $\mathcal{U}$ where

$$\mathcal{U} = \bigcap_{i=1}^{n} \mathcal{U}_i \tag{2}$$

Figure 6 shows the result $\mathcal{U}$. If the number of the cameras is small, there exists shape difference between $\mathcal{U}$ and the shape of the dynamic object, but it comes small as the number of the cameras increases.

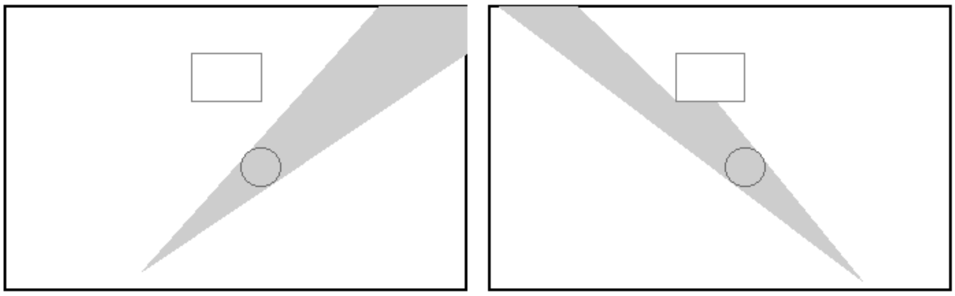


Figure 5: Existence Shadow Space $\mathcal{U}_1, \mathcal{U}_2$

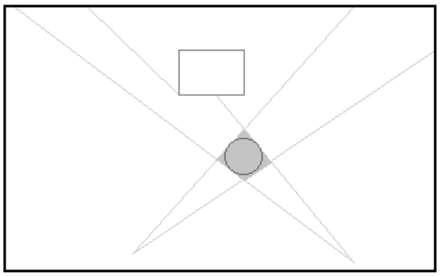


Figure 6: Reconstructed Shape $\mathcal{U}$

Suppose there are n cameras in the real space and the cameras capture images simultaneously. We call this set of images *a frame*. 3D reconstruction process named *3D composer* can generate $\mathcal{U}$ at each frame under the condition that $D_1, D_2, \cdots, D_n$ are given in advance, because $\mathcal{S}$ and the focus locations of the cameras are known. Since D_i is described by a binary image, the amount of transferred data is quite small.

This 3D reconstruction calculation in Equation (2) is easily expanded to parallel distributed computing because several 3D composers can reconstruct different subspaces simultaneously. Thus we achieve spatio-division of the 3D reconstruction process based on the locality of 3D reconstruction calculation.

In the actual implementation, 3D composer generates $\mathcal{U}$ by voxel representation.

2.2.3 Pipeline Processing

Our 3D reconstruction system named SCRAPER can be divided into three stages.

1. image capture
2. extraction of dynamic region
3. ESS calculation by VFM method

Let us call this sequence of stages a *path*. If the 3D reconstruction is done in this order sequentially, some parts of the system always idle. For example, when images are being captured, extraction and ESS calculation cannot be done. As a result, throughput is low and that is not desirable for real-time applications. To improve the throughput, we propose to activate several paths simultaneously in the pipeline architecture. We prepare three kinds of processes: *image captor*, *extractor*, and *3D composer* and increase the number of these processes to support multiple paths.

In our prototype system, an image is captured by the video capture card for which CPU power is not necessary whereas an extractor needs CPU power because it extracts $\mathcal{D}_i$ by detecting regions where the pixel values differ from its background image taken beforehand, so the two processes need only one CPU to work together. In addition, captured image data which is transferred to the extractor is not small and so it is not desirable to use physical network device to transmit the data between them. Therefore, we assign one video image captor and one extractor on the same workstation. As a result, the number of video image captors and that of the extractors are the same as that of the cameras.

On the contrary, the number of 3D composers can be increased because the calculation on the 3D composer is completely localized. The system can improve the throughput by preparing the 3D composers on different workstations distributed in a LAN.

As a consequence, the throughput is improved by preparing the multiple paths in the pipeline architecture, which means temporal division of 3D reconstruction process. The number of the paths are subjected to the number of the 3D composers the system can offer. Figure 7 shows the process timing chart when the system has three cameras and four 3D composers and assigns two 3D composers at each path.

We introduce a process named *scheduler* to synchronize the processes in the pipeline architecture.

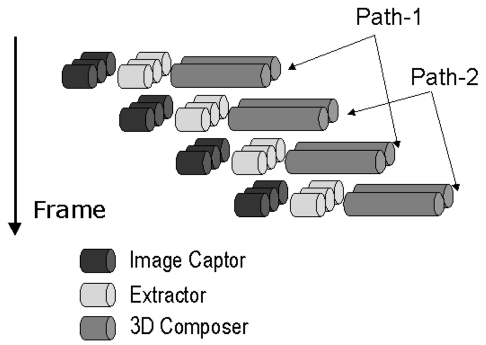


Figure 7: Pipeline Architecture

3 Multi-User Live Video Imaging

3.1 Video Imaging for Multiple Users

Another good approach to the information media environment is raw video based visualization because the raw video images framed by the active cameras are familiar for the users to see what happens in the target space .

As the users desire video images taken from various viewpoints and zoom values, multiple active cameras are set in the target space . Therefore the problem here is how to get the users' favorite video images.

Since it is difficult to see multiple video images at a time, the user is assumed to watch only one video image. A video image selection process for each user should be prepared which satisfies desires of the user under the physical constraint of the active cameras such as locations, pan/tilt angles, and camera ranges.

We solve this problem by introducing camera works that are used to proclaim user's favorite imaging method . The system adopts the camera work according to the description of the dynamic situations snatched in real-time.

Each user proclaims his/her favorite imaging method in advance by specifying several camera works at each dynamic situation that would be snatched in the target space .

Formulated camera works in movies[3] and computer supported movie makers[4][5] have been proposed, but they assume that the sceenplays or shooting scripts exist. Our case does not allow us to have such scripts, so we use the description of the dynamic situation instead.

In Section 3.2, the description of the dynamic situation and the camera work is presented, and mediation of the camera works among the requirements of the multiple users is discussed. Dynamic situations are detected in the way explained in Section 3.3. After those discussions, we examine agent design on constructing the information media environment in Section 3.4.

3.2 Description of Dynamic Situation and Camera Work

Since each user's favorite imaging method is different from those of other users, the user proclaims his/her imaging method for each dynamic situation . An imaging method consists of several camera works . We discuss the way to proclaim the imaging methods and introduce A-component so as to describe dynamic situation in the succeeding subsections.

3.2.1 A-component

A dynamic situation indicates a status of the real space at a time. Since there are some objects which are active in the dynamic situation , it can be described by a set of actions of the active objects. A description of this action is called *action component* and is denoted by *A-component* for short.

An A-component consists of one active object , one verb, one target object , and supplemental target objects . An active object is an object that can change the dynamic situation by itself. A verb describes what the active object does. A target object and supplemental target objects are both objects that are objective to the verb. Whereas a target object can be subjective , supplemental target objects could not be subjective . In other words, the supplemental target objects are static objects. Note that multiple supplemental target objects can be added on describing one A-component .

On describing the A-components , one A-component that has both an active object and a target object could be written in two ways; in the active form and the passive form. For example, an A-component of “A hits B” could be written by “A hits B” and “B is hit by A”. Since these two descriptions indicate the same action, we prohibit describing an action in the passive form.

In the case of lectures in the lecture room, the A-components are listed in Table 1. The A-component ID 1 and 2 should be written in active form. ‘Whole students’ object represents all the students attending the lecture whereas a ‘student group’ object consists of several students who are sitting in certain part of the lecture room. Therefore, there are several ‘student group’ objects in the room.

Table 1: A-Components

ID	active object	Verb	target object	supplemental target object
1	Lecturer	Talk	Whole Students	-
2	Lecturer	Talk	Whole Students	Blackboard
3	Lecturer	Write	-	Blackboard
4	Student Group	Become Restless	-	-
5	Student Group	Stay Calm	-	-

3.2.2 Camera Work

A camera work describes how to image one object at a time. It consists of three fields; label of an object, direction, and range.

$$w(objectlabel, direction, range) \tag{3}$$

Direction field indicates the relation between the direction of the object and the camera direction. Note that it implies the camera location because the cameras do not change their location in our system. Range field tells the size of the object in the image.

In our prototype system, we adopt eight directions and five discrete range values. See Figure 8 and Figure 9.

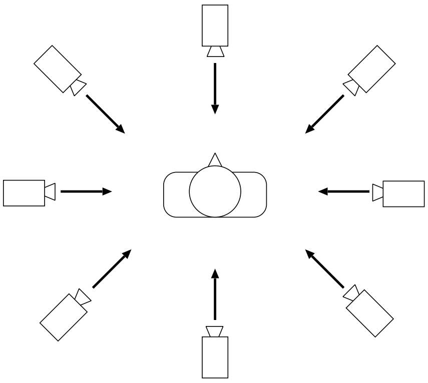


Figure 8: Camera Direction

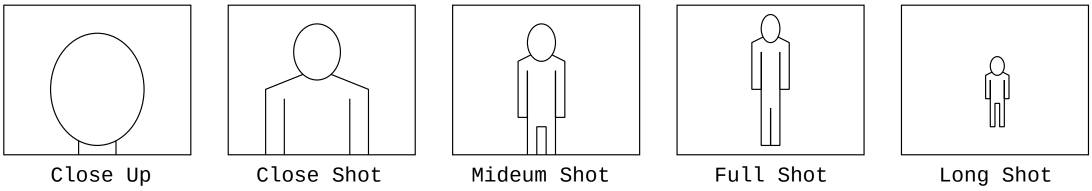


Figure 9: Camera Range

3.2.3 Imaging Method

An imaging method is a set of camera works defined for each dynamic situation by an user . If the user differs, the imaging method would be different for the same dynamic situation in the real space .

Consider a user j. An imaging method $\mathcal{I}^j$ consists of the A-components , and each A-component has a set of the camera works . The user j proclaims the camera works for the objects which appear in each A-component . The user can proclaim multiple camera works for the same object in a certain A-component if he/she wants to. Hence the number of the camera works in $\mathcal{I}^j$ is at most the summation of the number of camera works that the A-components have. For example, if $\mathcal{I}^j$ consists of two A-component p, q each of which the user j proclaims two camera works , $\mathcal{I}^j$ has four camera works .

Note that the user j does not need to define all the camera works for the objects in the A-components . For example, suppose the same example shown above and the user does not have a desire to see the active object in the A-component p, the user is required to proclaim only three camera works for $\mathcal{I}^j$. Since multiple A-components would occur in the real space at time t, the imaging method the user desires may be represented by a set of the camera works $\mathcal{I}^j(t)$.

Let us explain by an example. Suppose the user j defines the imaging method $\mathcal{I}^j$ like Table 2 against the A-components shown in Table 1. The empty rows in the Table 2 mean that the user j does not want to image the object (lecturer of the A-component No.3 for example) even if the corresponding A-component is detected.

If the A-component No.1 and No.4 are detected in the real space , three camera works w_1, w_2 and w_8 in Table 2 are submitted which represent the requirement of the user j at that time.

Table 2: An Example of Imaging Method $\mathcal{I}^j$ Defined by User j

A-Component	Object	Camera Work
1	lecturer	w_1(lecturer, front, closed shot)
	whole students	w_2(whole students, front, full shot)
2	lecturer	w_3(lecturer, front, full shot)
	whole students	w_4(whole students, right front, full shot) w_5(whole students, left front, full shot)
	blackboard	w_6(blackboard, front, full shot)
3	lecturer	-
	blackboard	w_7(blackboard, front, full shot)
4	student group	w_8(student group, right side, medium shot)
5	student group	-

In the human activities played in the real space , certain A-component sometimes occurs repeatedly, or one A-component may last in quite a long period. The users become feeling dull if there is no change in the video images due to the same A-components in the real space . To keep the video images interesting in these cases, it is a good approach to change the way of imaging the object even in the same A-component . To achieve this idea, the second and third field of the camera work would be modified within a long interval. This is called "camera work deviation ".

3.2.4 Mediation And User Satisfaction

The request of the user is satisfied if a camera work in $\mathcal{I}^j(t)$ is taken by a certain camera. Suppose there are u users and c cameras. A camera can realize only one camera work at a time, so at most c camera works can be realized at time t. As a consequence, mediation of user requests is to select at most c camera works among $\mathcal{I}^j(t)$ where $j = 1 \cdots u$. Let us denote $n(t)$ that means the number of the camera works in $\mathcal{I}^j(t)$ for all j. If it includes the same camera works , they are counted as one camera work . We can describe the mediation by the mediation matrix $M(t)$ which has $n(t)$ rows and u columns where each component $m_{ij}(t)$ is either 1 or 0. A column represents $\mathcal{I}^j(t)$ and a row corresponds one camera work i and so $m_{ij}(t) = 1$ means that user j realizes this request of the camera work i. Hence, the next equation ought to be true.

$$\sum_{i=1}^{n(t)} m_{ij}(t) = 1 \tag{4}$$

Note that at each column j, m_{ij} is always 0 if camera work i is not included by $\mathcal{I}^j(t)$. We introduce $a_i(t)$ that indicates whether the corresponding camera work of the ith row is selected or not. The number $s_i(t)$ indicates the number of the users who support the camera work i.

$$s_i(t) = \sum_{j=1}^{u} m_{ij}(t) \tag{5}$$

$$a_i(t) = \begin{cases} 1 & (\text{if } s_i(t) \geq 1) \\ 0 & (\text{otherwise}) \end{cases} \tag{6}$$

The constraint of the number of the cameras is formulated by

$$\sum_{i=1}^{n(t)} a_i(t) \leq c \tag{7}$$

The mediation process is finding the mediation matrix M that satisfies Equation (4), (6), and (7). The mediation matrix $M(t)$ is found in the case $u \leq c$, but there might be a dynamic situation where $u > c$ and so $M(t)$ does not exist. So, we select camera works in the following procedure so as to satisfy as many users as possible.

Mediation Procedure

1. Set the number of the remaining cameras $r = c$.
2. Calculate $s_i(t)$ for all i.
3. Select one camera work k where

$$s_k(t) = max_{i=1}^{n(t)} s_i(t) \tag{8}$$

4. Reduce the number of the rest cameras r to $c - 1$.
5. If r is 0, mediation procedure is finished.
6. Eliminate the users whose $\mathcal{I}^i$ contains the camera work k. Go back to the step 2.

Once the camera works are selected, the next problem is to assign the camera suitable to each selected camera works under the constraint of the camera location, direction and range. The criteria to solve this problem is to satisfy the requirement of the users as much as possible. Our strategy is : the selected camera works are ordered by $s_i(t)$ and choose the best camera in this order.

Therefore the video image user j watches is taken by the camera assigned to a selected camera work in the mediation matrix $M(t)$ if his/her imaging method is realized. On the other hand, there might be a case where some users cannot find their camera works in the selected camera works if $u > c$. Those who cannot satisfy with them select the second best camera work among them which is similar to one of the camera works in $\mathcal{I}^j(t)$ and watch the corresponding video image.

A user might have no submitted camera works in a certain dynamic situation . In this case, the user watches an arbitrarily chosen video image.

3.3 Snatching The Dynamic Situation

A dynamic situation is practically defined by the combination of the features extracted from the sensor data. We call these features the situation features . The feature extraction methods would differ if the real space and the activities are changed. Most of the situation features are extracted via image processing because the image sensor has an advantage that does not affect the human activities played in the real space .

As the imaging methods are proclaimed by A-components and the A-components consists of the objects, the situation features ought to be extracted for each object. They should be extracted with little delay because they reflect the real-time dynamic situation in the real space .

With respect to the lectures, we use three kinds of the situation features ; lecturer's location, lecturer's voice level, and activation degree of student group. The extraction methods are explained in Section 4.3.1.

3.4 Agent Design

The functions in this system are classified into three categories. One is for imaging objects based on the camera works , and another is for snatching the dynamic situation , and the other is for mediating the requirements of the multiple users . We design three types of agents for each function.

An agent that controls an active camera and images an object is called an imaging agent . Its purpose is to realize a camera work and generate video of the object. The number of the imaging agents is the same as that of the selected camera works at that time.

An agent that snatches the dynamic situation is called an observation agent . Unlike the imaging agents , the observation agents have different functions one another because each observation agent extracts different situation features . The number of the observation agents is determined by the number of the situation features that are needed to snatch the dynamic situation .

The last kind of the agents is designed mainly to mediate the requirements of the multiple users . We call this kind of agent a mediation agent . While imaging agents and observation agents are device (camera or sensor) dependent, the mediation agent is device independent. Currently, we build one mediation agent that interprets the information of the dynamic situation from situation features and selects the camera works according to the mediation procedure.

In the framework of the cooperative distributed vision[2], an agent has three functions such as perception, action, and communication. In our framework, the imaging agent plays an action role and the observation agent does a perception role. Both agents use the active cameras, but the observation agents do not exchange their cameras because the perception process should not miss what occurs in the real space and so they hold the cameras as their continuous sensors. On the other hand, the imaging agents can change their cameras one another because their purpose is to obtain the desired video image, and the exchange may lead them to achieve the specified camera works more precisely.

4 Experiments

4.1 Generation of Synchronized Virtual Space

We implemented a 3D reconstruction system named SCRAPER. We experimentally reconstructed a part of a lecture room in the graduate school of informatics in Kyoto University.

The target space is imaged by four SONY EVI-G20 video cameras fixed at the corners of the lecture room (Figure 10). Table 3 shows the camera location in the room coordinate system.

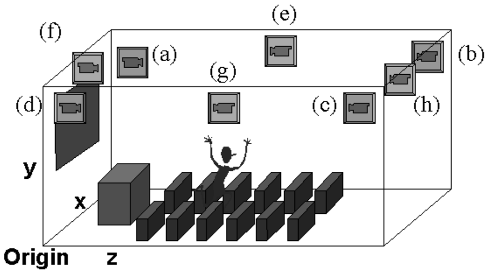


Figure 10: Camera Layout in The Lecture Room

In the experiment, we prepared four image captors and four extractors, and used four SUN Ultra2 200MHz workstations for them. We prepared four 3D composers and assigned them to four SUN Ultra1 170MHz workstations. A scheduler runs on a different workstation. All the workstations are connected on a LAN. The scheduler makes a synchronization among the image captors, the extractors and the 3D composers via 100

Table 3: Camera Location

Camera ID	X [m]	Y [m]	Z [m]
(a)	6.47	2.77	1.64
(b)	6.45	2.80	10.42
(c)	0.53	2.80	10.41
(d)	0.70	2.80	1.65

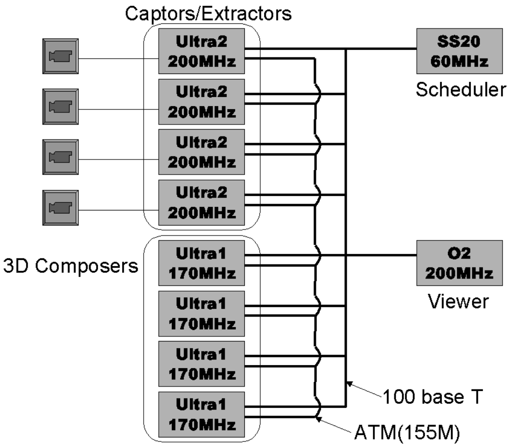


Figure 11: LAN Layout

base-T Ethernet and 155Mbps ATM LAN. The dynamic region data from the extractors to the 3D composers are transferred on ATM LAN.

Figure 12 shows SOOS defined by the static object database given in advance. The SOISs from each camera in Figure 10 are shown in Figure 13. These subspace have been calculated before the SCRAPER system starts the reconstruction.

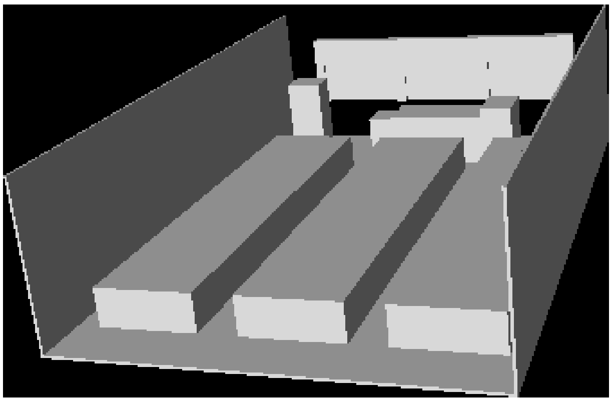
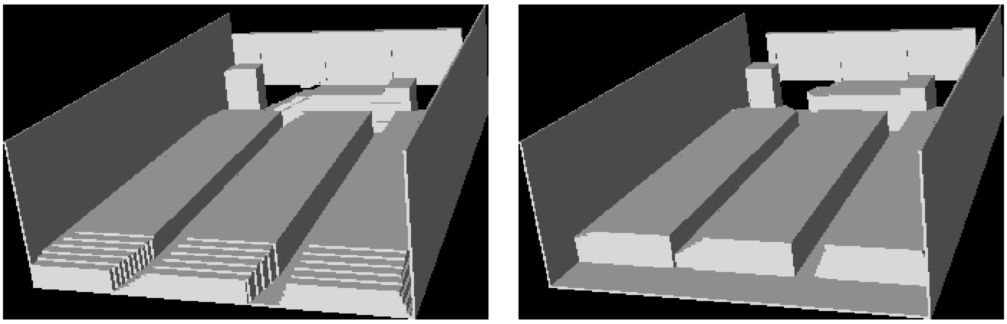
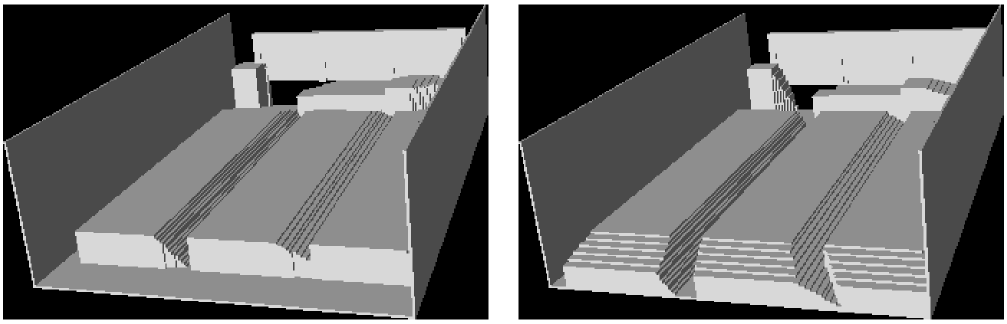


Figure 12: SOOS



Camera (a) Camera (b)



Camera (c) Camera (d)

Figure 13: SOIS of Cameras

The system reconstructed the target space which was imaged more than three cameras. Hence, several parts of the target space were observed by four cameras, and the other parts were observed by three cameras. In the case four cameras imaged the subspace, n in Equation (2) should be four, and in the other case, if a camera j could not observe the subspace, $\mathcal{U}$ is the product of $\mathcal{U}_i, (i = 1, 2, \cdots, n,\ i \neq j)$. Figure 14 displays the target space which is visible by at least three cameras in the lecture room.

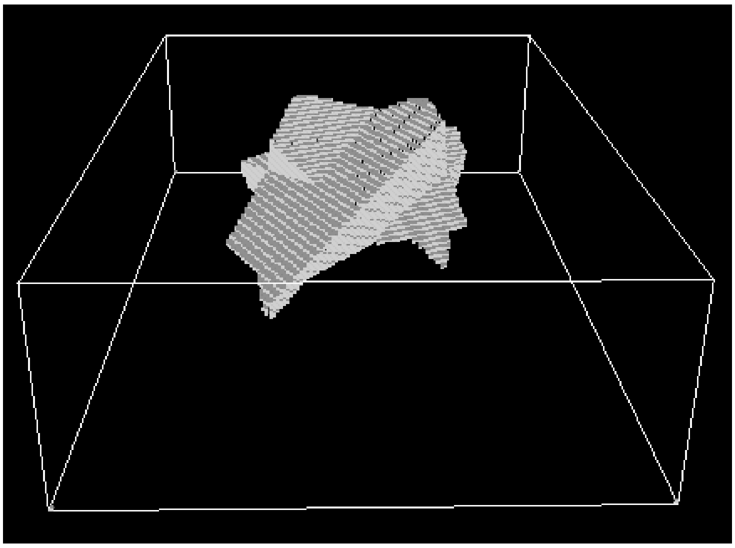


Figure 14: Target Space

In the experiment, the image captor takes images with the size of 320 × 240 pixels. The camera which locates the furthest location from the target space images a cubic subspace of 5 centimeters on a side in the target space onto one pixel in the captured image. Therefore, we set the voxel size as a cube of 5 centimeters on a side. The target space shown in Figure 14 corresponds to 96,769 voxels.

We conducted an experiment to measure the throughput and the latency of our prototype system. We put a box as a dynamic object whose size is $55cm \times 55cm \times 25cm$. The result of using four 3D composers are shown in Table 4. A variable r indicates number of 3D composers served in each path and s indicates number of paths in the system. We also conducted an experiment with only one 3D composer just for comparison and its throughput is 2.2 fps and its latency is 1,384 $msec$.

The required throughput and latency differ according to the applications. One good feature of our method is that we can change the formation suitable to the applications by changing r and s. The result indicates that the case of two 3D composers at two paths is good because the throughput is almost the same as four 3D composers at one path and the latency is as short as that of the case of the four paths.

We implemented a virtual space viewer which displays the reconstructed real space as a set of voxels in real-time. This viewer is an implementation of the information media environment . It displays not only the dynamic objects but also the static objects given

Table 4: Throughput and Latency

3D comp. per path : r	1	2	4	1
Number of paths : s	4	2	1	1
Latency [msec]	730	560	490	1,384
Throughput [fps]	7.3	7.2	6.1	2.2

to the system in advance, so a user can walk around the lecture room and observe the real space from any viewpoint with a little delay.

An example of a captured image is shown in Figure 15 . Figure 16 shows the reconstructed space displayed by the viewer. The voxels displayed in the center corresponds to $\mathcal{U}$, which were transmitted from the SCRAPER system.

Camera (a) Camera (b)

Camera (c) Camera (d)

Figure 15: Input Video Images

4.2 Camera Navigation in The Synchronized Virtual Space

The virtual space viewer is a kind of virtual camera which does not have any physical

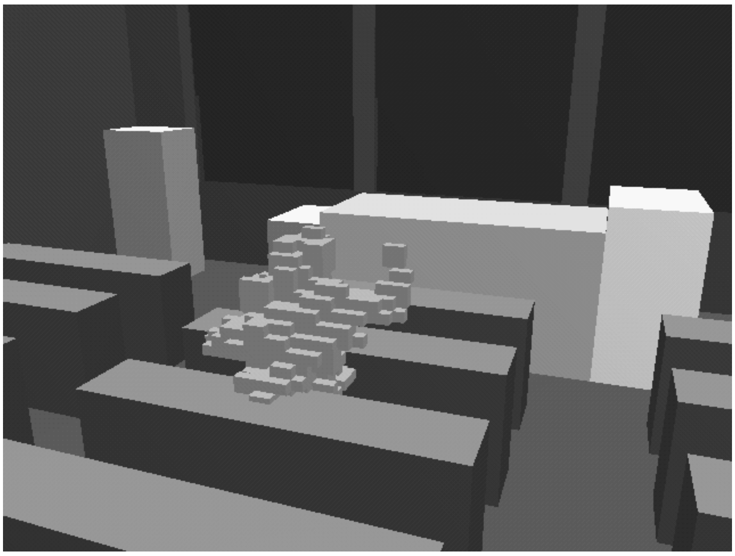


Figure 16: Reconstructed Space from Virtual Viewpoint

constraint.

Since the synchronized virtual space is fully constructed by 3D data, each user in the information media environment has his/her own cameras and imaging method and there is no need to negotiate with other users. This concept is called animated vision[11], but ours is different from others in the sense that the virtual space is linked to the real space .

We currently assume that the user wants to watch the whole of the dynamic objects in full shot. The direction field of the camera work is arbitrary.

4.3 Multi-User Live Video Imaging

4.3.1 Situation Feature

At the implementation for the lectures in the lecture room, the system uses three kinds of the situation features to snatch the five A-components shown in Table 1 : lecturer's location, voice level, and activation degree of student group.

The lecturer's location indicates the location of the lecturer in the real space . The lecturer's voice level shows the loudness of his/her voice. The activation degree of the student group becomes large as they move their bodies.

The lecturer's location is measured by the image based triangulation by two active cameras. As the lecturer walks around in the lecture room, the active cameras track the lecturer and calculate the location based on the camera parameters and the subtraction regions. We assume that the lecturer is the only wandering object.

To obtain the lecturer's voice level, the lecturer is asked to equip a wireless microphone and the input level of the A/D converter is used directly.

Table 5: Situation Features and The Dynamic Situations

Situation ID	Lecturer Location	Lecturer Direction	Lecturer Voice Level	Student Group Activation
1	-	student	positive	-
2	-	blackboard	positive	-
3	blackboard	blackboard	-	-
4	-	-	-	positive
5	-	-	-	zero

We divide the student desks into six groups and call the students in one desk group the student group in this experiment. The activation degree of the student group is presented by the area of the subtraction region in the image in which the student group is framed from their front view.

4.3.2 Experimental Result

Here we present some snapshots of two video images generated for a lecture held at the same lecture room in Section 4.1. Camera (b) and (c) in Table 3 were used for extracting the situation features related to the lecturer. Camera (a) and (d) were used for measuring the activities of the student groups. The imaging agents used four active cameras which are located as shown in Table 6.

Table 6: Camera Location for Imaging

Camera ID	X [m]	Y [m]	Z [m]
(e)	6.84	2.17	3.63
(f)	4.59	2.43	0.41
(g)	0.64	2.32	7.37
(h)	3.741	2.16	10.73

The video images in Figure 17 and Figure 18 are generated for the different users . User (A) proclaimed three camera works whereas user (B) proclaimed five camera works . The horizontal axis indicates the time flow for about four minutes and C1 to C5 are the labels of the selected camera work . Note that both users sometimes watch the same video image because their desired camera works are overlapped at that time.

5 Conclusion

We have proposed the information media environment that allows the users to view what happens in the real space with assistance of the computers. Two types of its implementation are presented in this paper.

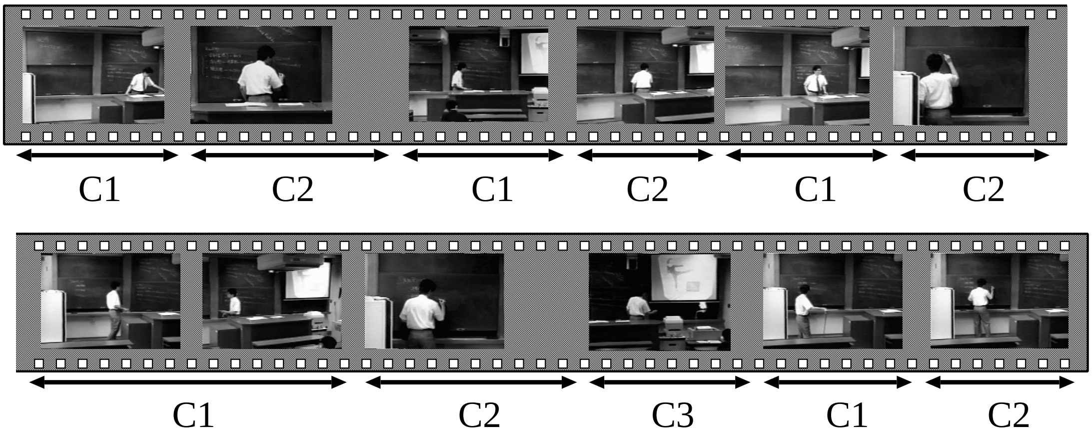


Figure 17: Video Images for User (A)

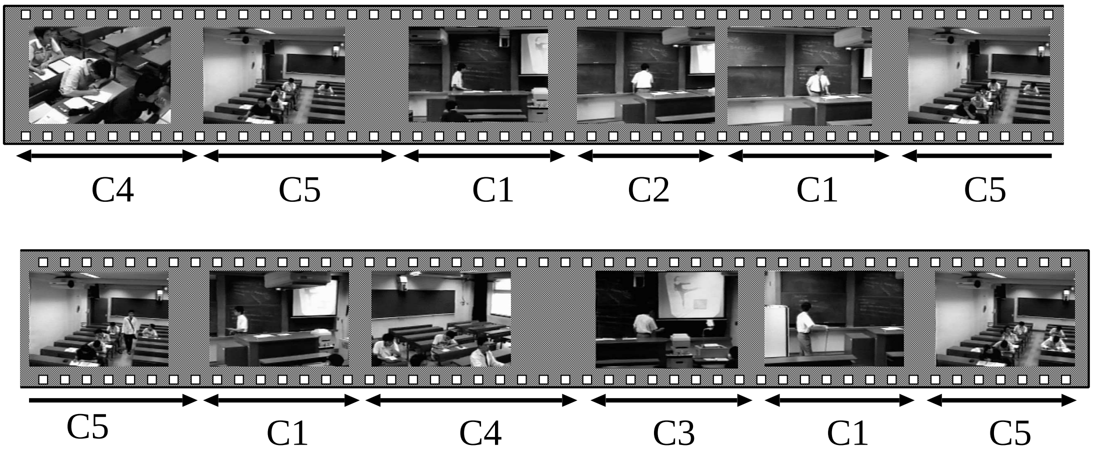


Figure 18: Video Images for User (B)

One approach of making the information media environment is to construct the synchronized virtual space that permits the users to fly in the space and see the objects from any viewpoints without any constraint. For this purpose, we have presented the method of high speed 3D reconstruction in the situation the multiple cameras surround a certain real space .

The other approach is to distribute raw video images to the users that realizes their favorite imaging methods according to the snatched dynamic situation in the real space . For this purpose, we defined the relationship between the imaging methods of the users and the camera works , and formulated the mediation between the users' imaging requests and the physical constraint of the cameras so that most users can see video images as they desire.

We are planing to add photometric information in the synchronized virtual space and merge these two approaches into one unified information media environment .

Bibliography

[1] Y. Kameda, T.Taoda, M.Minoh: "High Speed 3D Reconstruction by Video Image Pipeline Processing and Division of Spatio-Temporal Space," IAPR Workshop on Machine Vision Applications, 1998 (to be printed).

[2] T. Matsuyama: " Cooperative Distributed Vision – Dynamic Integration of Visual Perception, Action, and Communication –," Proc. of Image Understanding Workshop, Monterey CA, Nov, 1998.

[3] D.Arijon: "Grammer of the Film Language," Focal Press Limited, 1976.

[4] L. He, M.F. Cohen, D.H. Salesin: " The Virtual Cinematographer: A Paradigm for Automatic Real-Time Camera Control and Directing," SIGGRAPH'96, pp.217-224, 1996.

[5] D.B. Christianson, S.E. Anderson, L. He, D.H. Salesin, D.S.Weld, and M.F. Cohen: "Declarative Camera Control for Automatic Cinematography," Proceedings of AAAI '96, pp.148-155, 1996.

[6] P.J.Narayanan, P.W.Rander, and T.Kanade: "Constructing Virtual Worlds Using Dense Stereo," Proc. Sixth IEEE ICCV, pp.3-10,1998.

[7] J.E.Boyd, E.Hunter, P.H.Kelly, T.Li-Cheng, C.B.Phillips, and R.C.Jain, "MPI-Video infrastructure for dynamic environments," Proc. IEEE ICMCS, pp.249-254, 1998.

[8] T.Kanade, A.Yoshida, K.Oda, H.Kano, and M.Tanaka, "A Stereo Machine for Video-rate Dense Depth Mapping And Its New Applications," Proc. CVPR, pp.196-202, 1996.

[9] K.Sato, A.Yokoyama, and S.Inokuchi, "Silicon range finder-a real-time range finding VLSI sensor," Proc. IEEE 1994 CIC, pp.339-342, 1994.

[10] R.Raskar, G.Welch, M.Cutts,A.Lake,L.Stesin, and H.Fuchs, "The Office of the Future: A Unified Approach to Image Based Modeling and Spatially Immersive Displays," SIGGRAPH98 Conference Proceedings, Annual Conference Series, pp.179-188, 1998.

[11] R.Grzeszczuk, D.Terzopoulos, G.Hinton, "NeuroAnimator: Fast Neural Network Emulation and Control of Physics-Based Models," SIGGRAPH98 Conference Proceedings, Annual Conference Series, pp.9–20, 1998.