Active Image Capturing and Dynamic Scene Visualization by Cooperative Distributed Vision

Takashi Matsuyama, Toshikazu Wada, and Shogo Tokai

Department of Intelligence Science and Technology
Kyoto University, Kyoto 606-8501, Japan
e-mail:{tm, twada, tokai}@i.kyoto-u.ac.jp

Abstract. This paper addresses active image capturing and dynamic scene visualization by *Cooperative Distributed Vision* (CDV, in short). The concept of CDV was proposed by our five years project starting from 1996. From a practical point of view, the goal of CDV is summarized as follows: Embed in the real world a group of network-connected *Observation Stations* (real time video image processor with active camera(s)) and mobile robots with vision. And realize 1) wide-area dynamic scene understanding and 2) versatile scene visualization. Applications of CDV include real time wide-area surveillance, remote conference and lecturing systems, interactive 3D TV and intelligent TV studio, navigation of (non-intelligent) mobile robots and disabled people, cooperative mobile robots, and so on. In this paper, we first define the framework of CDV and give a brief retrospective view of the computer vision research to show the background of CDV. Then we present technical research results so far obtained: 1) fixed viewpoint pan-tilt-zoom camera for wide-area active imaging, 2) moving object detection and tracking for reactive image acquisition, 3) multi-viewpoints object imaging by cooperative observation stations, and 4) scenario-based cooperative camera-work planning for dynamic scene visualization. Prototype systems demonstrate the effectiveness and practical utilities of the proposed methods.

1 Introduction

This paper addresses active image capturing and dynamic scene visualization by *Cooperative Distributed Vision* (CDV, in short). The concept of CDV was proposed by our five years project starting from 1996.

From a practical point of view, the goal of CDV is summarized as follows (Fig. 1):

Embed in the real world a group of network-connected *Observation Stations* (real time video image processor with active camera(s)) and mobile robots with vision, and realize

1. wide-area dynamic scene understanding and
2. versatile scene visualization.

We may call it *Ubiquitous Vision.*

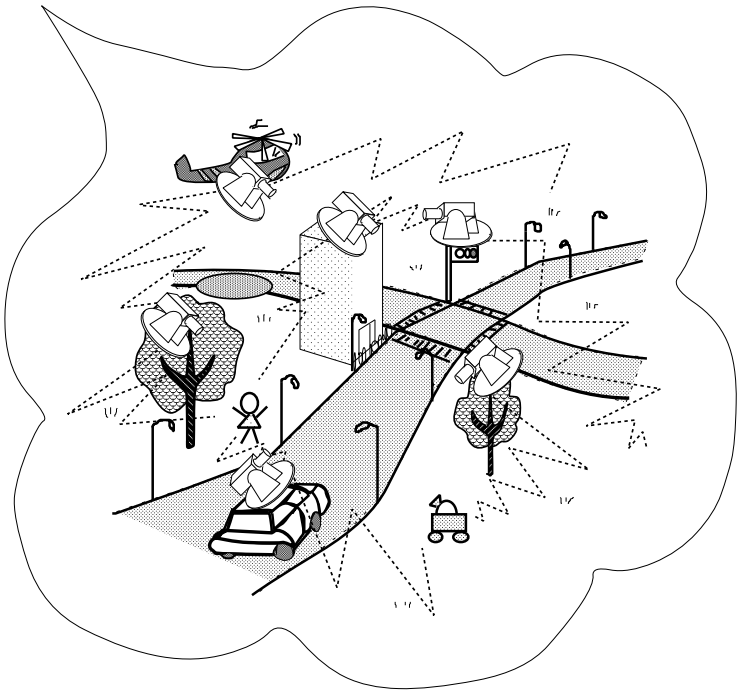


Fig. 1. Cooperative distributed vision.

Applications of CDV include

- Real time wide-area surveillance and traffic monitoring systems
- Remote conference and lecturing systems
- Interactive 3D TV and intelligent TV studio
- High fidelity imaging of skilled body actions (arts, sports, medical operations)
- Navigation of (non-intelligent) mobile robots and disabled people
- Cooperative mobile robots.

The aim of the project is not to develop these specific application systems but to establish scientific and technological foundations to realize CDV systems enough capable to work persistently in the real world.

From a scientific point of view, we put our focus upon *dynamic integration of visual perception, action, and communication.* That is, the scientific goal of the project is to investigate how the *dynamics* of these three functions can be characterized and how they should be integrated *dynamically* to realize intelligent systems [1].

From a technological point of view, we design and implement hardwares and softwares to embody these three functions:

Visual Perception : versatile and high precision visual sensors, parallel and distributed real time vision systems.

Action : active camera heads, mobile robots with vision, and their dynamic control systems.

Communication : high speed wired and wireless network systems, communication protocols for cooperation, and cooperative distributed problem solving methods.

In this paper, we first define the framework of CDV and give a brief retrospective view of the computer vision research to show the background of CDV. Then we present technical research results so far obtained: 1) fixed viewpoint pan-tilt-zoom camera for wide-area active imaging, 2) moving object detection and tracking for reactive image acquisition, 3) multi-viewpoints object imaging by cooperative observation stations, and 4) scenario-based cooperative camerawork planning for dynamic scene visualization. Prototype systems demonstrate the effectiveness and practical utilities of the proposed methods.

2 Background and Basic Idea

Roughly speaking, the history of the computer vision research can be summarized as follows (Fig. 2):

–**1970s: Image Processing**: 2D Image $\mapsto$ 2D Image
A *given* input image is transformed into an output image to enhance its quality and to detect image features.

1980s: Computer Vision[1] : 2D Image $\mapsto$ 3D Scene
Recover 3D scene information from observed 2D image(s) based on geometric and photometric models of the imaging process.

1990s: The following two disciplines are being studied:

1. **Active Vision**: Computer Vision $\times$ Physical Action $\mapsto$ Active Scene Understanding
   Integrate visual perception and physical action for active exploration of complex scenes [2], [3].
2. **Image Media Processing**:
   Computer Vision $\times$ Computer Graphics $\mapsto$ Versatile Scene Visualization
   Integrate image analysis and synthesis methods to realize versatile scene visualization. Fig. 3 illustrates an example of the integration process: 3D Scene — *Imaging* $\rightarrow$ 2D Image(s) — *Computer Vision* $\rightarrow$ 3D Scene Description — *Edit* $\rightarrow$ Augmented 3D Scene Description — *Computer Graphics* $\rightarrow$ Image(s) of Virtualized/Augmented Scene.

The key idea of CDV is to *introduce network communication capabilities into active vision and image media processing.* That is, with the introduction of network communication capabilities, CDV systems are endowed with three

[1] Here we use "computer vision" in a narrow sense denoting computational and physics-based vision.

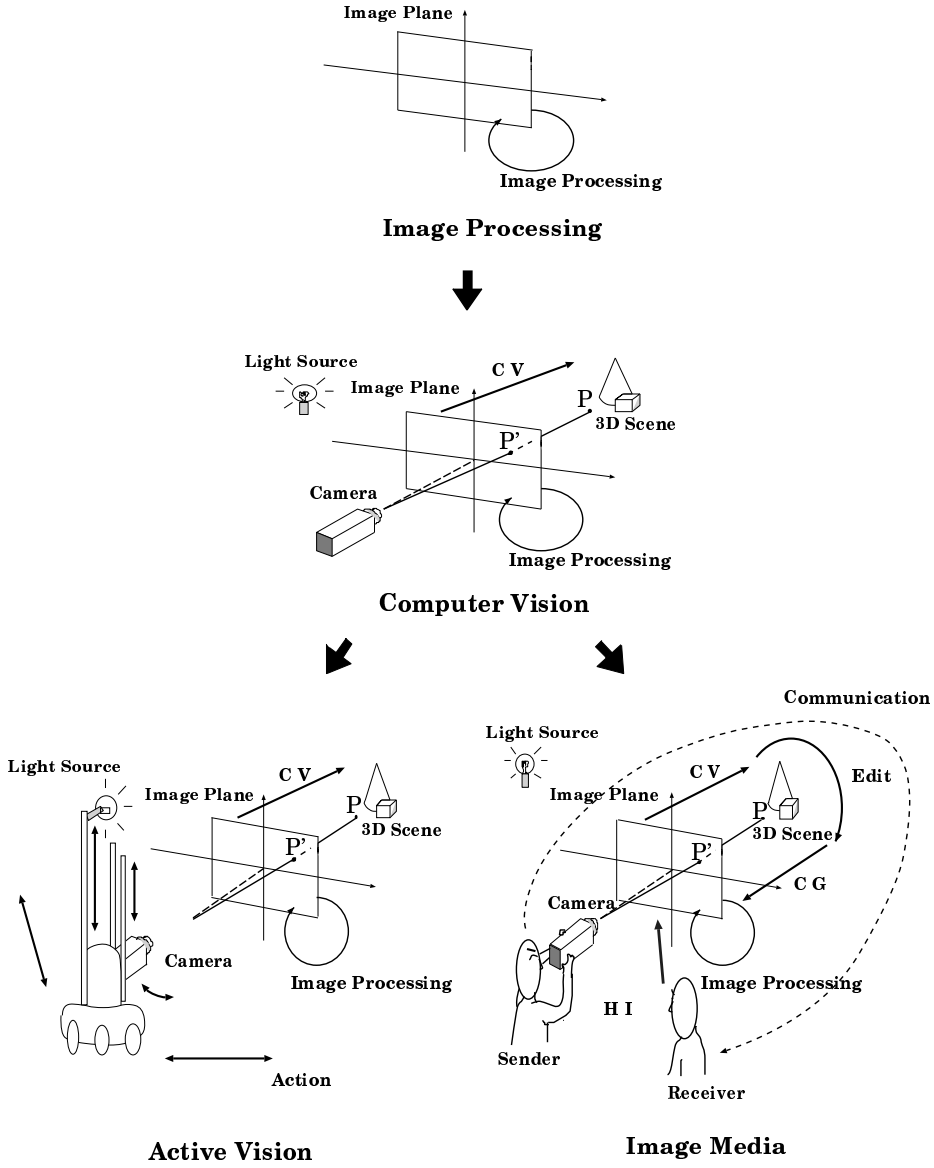


Fig. 2. History of the computer vision research.

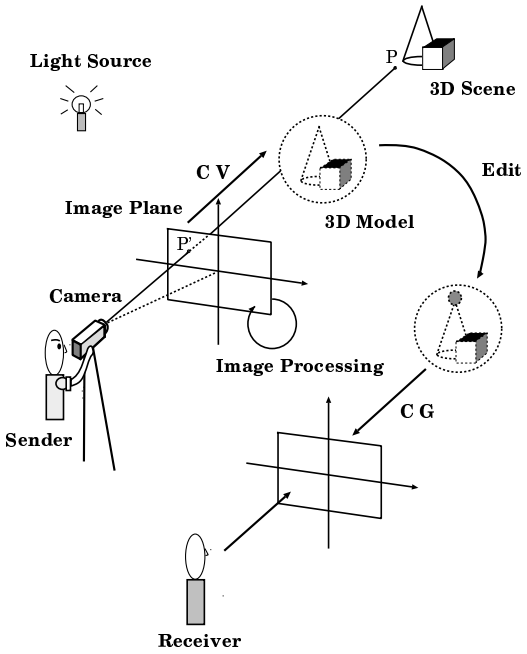


Fig. 3. Integration of computer vision and computer graphics.

functions of Visual Perception, Action, and Communication. The goal of CDV is to integrate these functions to realize the following cooperative distributed processing mechanisms:

Dynamic Wide-Area Image Capturing : A group of network-connected cameras are distributed over a wide spread area to realize dynamic multi-viewpoint object/scene imaging.

Reactive Image Acquisition : The active and coordinated control of the distributed cameras enables reactive image acquisition: object/scene images are captured depending on their dynamic behaviors/situations.

Rich and Robust Observation : Multiple pieces of information from different cameras are integrated to increase the accuracy and reliability of image analysis/synthesis as well as to measure 3D information.

Adaptive System Organization : Groups of cooperative observation stations are adaptively formed to cope with dynamically changing situations in the real world.

Using these mechanisms, both wide-area dynamic scene understanding and versatile scene visualization systems can be implemented.

We believe CDV offers a fundamental framework of visual information processing systems in the 21st century.

3 Fixed-Viewpoint Pan-Tilt-Zoom Camera for Wide-Area Active Imaging

3.1 Realization of Wide View Cameras

To develop wide-area video monitoring systems, we first of all should study methods of expanding the visual field of a video camera:

1. Omnidirectional cameras using fish-eye lenses and curved mirrors[4], [5], [6], or
2. Active cameras mounted on computer controlled camera heads[7].

In the former optical methods, while omnidirectional images can be acquired at video rate, their resolution is limited. In the latter mechanical methods, on the other hand, high resolution image acquisition is attained at the cost of limited instantaneous visual field.

In the CDV project, we took the active camera method;

- High resolution images are of the first importance for object recognition and scene visualization.
- Dynamic resolution control can be realized by active zooming, which increases adaptability and flexibility of the camera system.
- The limited instantaneous visual field problem can be solved by incorporating a group of distributed cameras.

Then, the next issue to be studied is how to design an active camera system. In this section, we first present an idea of a fixed viewpoint pan-tilt camera[8] and show the active camera head designed based on this idea. In the latter half of the section, we describe a sophisticated camera calibration method to make a commercial active video camera work as a fixed viewpoint pan-tilt-zoom camera. Experimental results demonstrate its practical utilities.

3.2 Fixed Viewpoint Pan-Tilt-Zoom Camera

Suppose we design a pan-tilt camera, where its optical axis is rotated around pan and tilt axes. This active camera system includes a pair of geometric singularities: 1) the projection center of the imaging system[2] and 2) the rotation axes. In ordinary active camera systems, no deliberate design about these singularities is incorporated, which introduces difficult problems in image analysis. That is, the discordance of the singularities causes photometric and geometric appearance variations during the camera rotation: varying highlights and motion parallax. In other words, 2D appearances of a scene change dynamically depending on the 3D scene geometry. To cope with such appearance variations, consequently, sophisticated image processing should be employed[7].

The following active camera design eliminates the appearance variations and hence greatly facilitates the image processing [8].

[2] We model the optical process of a camera by the perspective projection.

1. Make pan and tilt axes intersect with each other. The intersection should be at right to facilitate later geometric computations.
2. Place the projection center at the intersecting point. The optical axis of a camera should be perpendicular to the plane defined by the pan and tilt axes.

We call the above designed active camera the *Fixed Viewpoint Pan-Tilt Camera* (FV-PT camera, in short).

Usually, zooming can be modeled by the shift of the projection center along the optical axis[9]. Thus to realize the *Fixed Viewpoint Pan-Tilt-Zoom Camera* (FV-PTZ camera, in short), either of the following additional mechanisms should be employed:

- Design such a zoom lens system whose projection center is fixed irrespectively of zooming.
- Introduce a slide stage which adjusts the projection center fixed depending on zooming.

3.3 Image Representation for FV-PTZ Camera

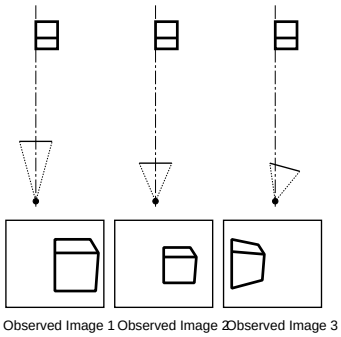


Fig. 4. Images observed by an FV-PTZ camera.

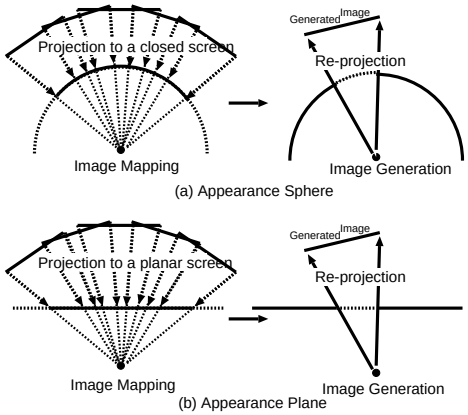


Fig. 5. Appearance sphere and plane.

While images observed by an FV-PTZ camera do not include any geometric and photometric variations depending on the 3D scene geometry, object shapes in the images vary with the camera rotation (Fig. 4). These variations are caused by the movement of the image plane, which can be rectified by projecting observed images onto a common virtual screen. On the virtual screen, the projected images form a seamless wide-area panoramic image.

For the rectification, we can use arbitrarily shaped virtual screens. The following are typical examples:

APS: When we can observe the 360° panoramic view, a spherical screen can be used (Fig. 5 (a)). We call the omnidirectional image on the spherical screen *APpearance Sphere* (APS in short).

APP: When the rotation angle of the camera is limited, we can use a planar screen (Fig. 5 (b)). The panoramic image on the planar screen is called *APpearance Plane* (APP in short).

As illustrated in the right side of Fig. 5, once an APS or an APP is obtained, images taken with arbitrary combinations of pan-tilt-zoom parameters can be generated by re-projecting the APS or APP onto the corresponding image planes. This enables the virtual look around of the scene.

The above mentioned omnidirectional image representation is equivalent to those proposed in [10] ∼ [12] in Computer Graphics and Virtual Reality. Our objective, however, is not to synthesize panoramic images natural to human viewers but to develop an active camera system that facilitates the image analysis for wide-area video monitoring. That is, in our case both the image acquisition and the projections onto/from virtual screens should be enough accurate to match well with physical camera motions. To attain such accuracy, we have to develop sophisticated camera calibration methods.

3.4 Camera Calibration

Fig. 6. Developed FV-PT camera head.

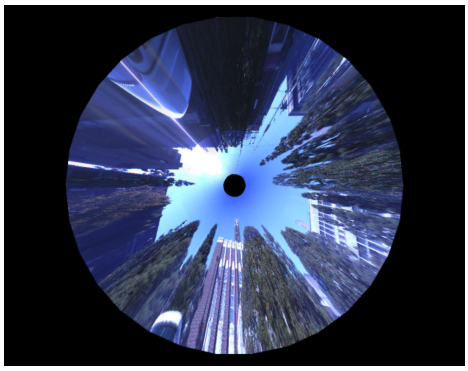


Fig. 7. High resolution APS representation of Kyoto University Clock Tower scene.

Fig. 8. FV-PTZ camera.

Fig. 6 shows the FV-PT camera head we developed, where a video camera is mounted on a group of adjustable slide and slant stages. We developed a high-precision camera calibration method using a laser beam to make the projection

center coincide with the rotation center [8]. The wide rotation angles (i.e. -180° $\leq$ pan $\leq$ 180° and 0 $\leq$ tilt $\leq$ 45°) enables the APS representation of a scene (Fig. 7). Note that using this camera head, any (compact) video camera with any lens system can be calibrated to realize an APS camera.

Fig. 8, on the other hand, illustrates an off-the-shelf active video camera, SONY EVI G20, which we found is a good approximation of an FV-PTZ camera ($-30° \leq$ pan $\leq 30°$, $-15° \leq$ tilt $\leq 15°$, and zoom: $15° \leq$ horizontal view angle $\leq 44°$) . We developed a sophisticated camera calibration method for this camera, with which we can use it as an FV-PTZ camera [1]. Note that this calibration method does not require any reference objects, and can be conducted automatically without any human support.

Fig. 9 illustrates a group of observed images with different (pan, tilt) parameters. Fig. 10 show the generated APP image. We verified that the physically accurate image mosaicing is realized on the APP image.

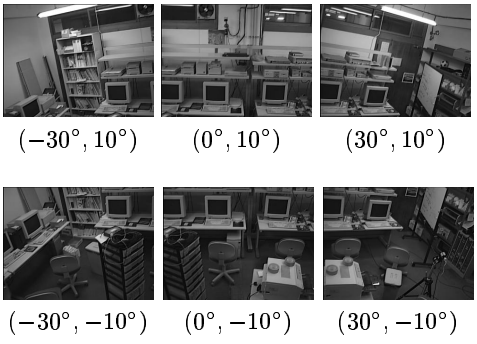


Fig. 9. Observed images.

Fig. 10. APP image generated from those in Fig. 9.

4 Moving Object Detection and Tracking for Reactive Image Acquisition

Since scenes in the real world are dynamically changing, image acquisition for computer vision and scene visualization should be done adaptively to dynamic situations. We call such adaptive image acquisition *Reactive Image Acquisition.*

This section first proposes a real time active vision system for object detection and tracking using the FV-PTZ camera. The tasks of the system are 1) detect an object which comes into the scene, 2) track it by controlling pan-tilt parameters, and 3) capture object images in as high resolution as possible by controlling the zoom. The system incorporates a sophisticated prediction-based dynamic control method 1) to cope with delays involved in image processing and physical camera motion and 2) to synchronize image acquisition and camera motion.

Experimental results demonstrated that the proposed dynamic control method greatly improves the performance of the object tracking.

In the latter part of the section, we develop a dynamic scene visualization system using the above proposed active object tracking method. With this system, we can monitor detailed high-resolution object behaviors as well as its surrounding wide-area environments by a single FV-PTZ camera.

4.1 Basic Scheme of Object Detection and Tracking

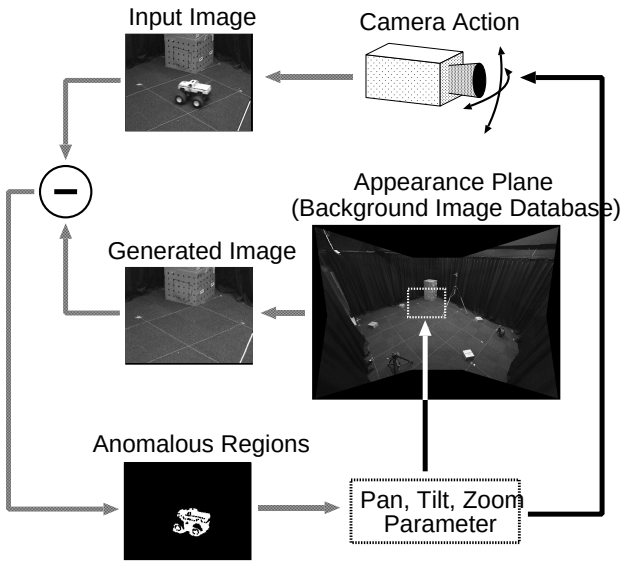


Fig. 11. Basic scheme of the object detection and tracking system.

Fig. 11 illustrates the basic scheme of real-time moving object detection and tracking by the FV-PTZ camera:

1. Generate the APP image of the scene.
2. Extract a window image from the APP according to the current pan-tilt-zoom parameters and regard it as the background image.
3. Compute difference between the background image and an observed image.
4. If anomalous regions are detected in the difference image, select one and control the camera parameters to track the selected target.
5. Otherwise, move the camera along the predefined trajectory to search for an object.

This scheme is too naive and should be augmented in the following points:

Robust background subtraction : Although the background subtraction is a useful method to detect and track moving objects in video images, its effectiveness is limited; the stationary background scene assumption does not hold always in the real world.

System dynamics : The system dynamics realized by repeating the above steps sequentially is too simple to make the system adaptable to dynamically varying target object behaviors.

To augment the background subtraction for non-stationary scenes, [13], [14], and [15] employed probability distributions to model intensity variations at each pixel and used probabilistic anomaly computation methods for object detection. In [16], we proposed a novel robust background subtraction method for non-stationary scenes, where non-stationarities are modeled by 1) variations of overall lighting conditions and 2) local image pattern fluctuations caused by soughing leaves, flickering CRTs and so on. Since this method is time consuming, the current system employs the standard background subtraction followed by several auxiliary image processing operators.

In what follows, we concentrate ourselves on the design of the system dynamics.

4.2 Dynamic Planning of Camera Action and Image Acquisition Timing

The basic scheme requires that the image acquisition should be done taking the following points into account:

- **State of Action**: To prevent motion blurs from being included in an observed image[3] , the image acquisition should be done when the camera stops or its speed is very slow. This means that the image acquisition cannot be done based on periodic clocks but should be triggered depending on the state of camera motion.
- **State of Target**: The image acquisition is to be done only when observed images are meaningful. That is, the images should include the target object in good appearance.

Thus, the determination of the image acquisition timing becomes a major concern in designing the system dynamics.

Fig. 12 shows the time chart of the perception-action cycle. Suppose the image acquisition is initiated at t_0. The right vertical bar in Fig. 12 illustrates the video cycle, which is not synchronized with the system; our FV-PTZ camera cannot accept the external trigger. Then, what the system has to determine are

1. $t_0 + \widehat{t}_d$: the next image acquisition time and

[3] Motion blurs in an observed image incurs many false alarms in the background subtraction.

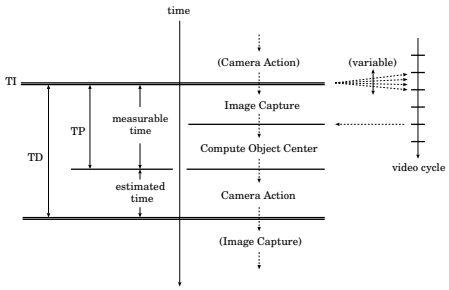


Fig. 12. Time chart.

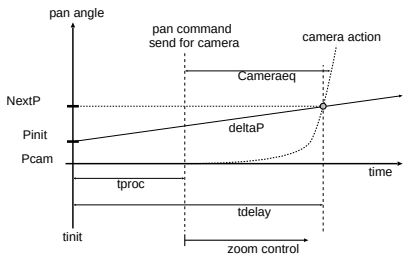


Fig. 13. Estimation of the next view direction and image acquisition timing.

2. such camera control command that satisfies
   1) A good target object image is taken at $t_0 + \widehat{t}_d$.
   2) The camera motion is enough slow to apply the background subtraction at $t_0 + \widehat{t}_d$.

To solve theses problems, we first estimate the camera action dynamics. We conducted extensive experiments to model the dynamics of our FV-PTZ camera and obtained the following linear model:

$$t = \mathcal{T}(\Delta P_{cam}, \Delta T_{cam}) = 0.007745 \times max\{\Delta P_{cam}, \Delta T_{cam}\} + 0.2986, \qquad (1)$$

where $\mathcal{T}(\Delta P_{cam}, \Delta T_{cam})$ denotes the time required to change pan and tilt angles by $(\Delta P_{cam}, \Delta T_{cam})$ and t is measured in second.

During tracking, the system measures t_p (see in Fig. 12) based on its internal clock and estimates the 2D target motion from the centroid displacement between object regions in a pair of consecutive video frames.

Then the system estimates both $(\Delta P_{cam}, \Delta T_{cam})$ to guide the camera toward the next view direction and $\widehat{t}_d$, the next image acquisition timing in the following way. Suppose $\Delta T_{cam} < \Delta P_{cam}$. Fig. 13 graphically illustrates the dynamics of the target motion and camera action. That is, $\widehat{t}_d$ and ΔP_{cam} are determined by the intersection point between the straight line representing the predicted target motion and the bent line representing the camera dynamics.

4.3 Dynamic Zoom Control

The dynamic zoom control should be implemented taking into account the following trade-off:

- To keep the target captured in observed images, wider view angles should be used; wider view fields can accommodate errors involved in the target motion and camera action estimations as well as image processing.
- To acquire detailed object images, larger zooming factors should be used.

To solve this trade-off, we employed the following dynamic zoom control method. During tracking, the system computes the instantaneous uncertainty degree at the ith observation time t_i, $\Delta UD(t_i)$:

$$\Delta UD(t_i) = \frac{POS_{error}(t_i)}{T(t_i) \times \sqrt{AREA(t_i)}}, \tag{2}$$

where $POS_{error}(t_i)$ denotes the positional prediction error at t_i, $T(t_i)$ the time interval between t_{i-1} and t_i, and $AREA(t_i)$ the area size of the target observed at t_i. Then, the system records the maximum possible uncertainty degree

$$\Delta UD_{max} = max\{\Delta UD(t_i)\}. \tag{3}$$

Then, the system determines the zooming factor $\alpha(t_{i+1})$ for the next observation so that the maximum possible position error, $POS^{max}_{error}(t_{i+1})$, defined by the following equation becomes less than the prefixed threshold.

$$POS^{max}_{error}(t_{i+1}) = \Delta UD_{max} \times (t_{i+1} - t_i)\sqrt{AREA(t_{i+1})} \tag{4}$$

$$AREA(t_{i+1}) = AREA(t_i) \times \alpha(t_{i+1}). \tag{5}$$

We conducted experiments to investigate the dynamics of the zoom control mechanism of our FV-PTZ camera and got the following observations:

- The zoom control can be done independently of the pan-tilt control.
- After the latency of about 0.05 sec, the zooming factor changes almost linearly.

Considering these observations and equation (1), which represents the dynamics of the pan-tilt control, the following zoom control method was implemented. 1) The pan-tilt control should have higher priority than the zoom control. 2) The former requires at least 0.2986 sec. Consequently, 3) the zoom can be changed in parallel with the pan-tilt control if the zoom control time is less than 0.2986 sec (see the bottom of Fig. 13). That is, after computing $\alpha(t_{i+1})$, the system modifies the zooming factor only by such an amount that satisfies this temporal constraint.

4.4 Performance Evaluation

To demonstrate the effectiveness of the proposed object tracking method, we conducted experiments to detect and track a radio controlled toy car. The car is manually controlled by a human; it moves around on the 4m $\times$ 4m flat floor avoiding several obstacles and sometimes stops and changes directions. The FV-PTZ camera is placed at about 2.5m above the floor corner looking downward obliquely. Fig. 14 shows a sequence of observed images and detected target silhouettes. Figs. 15 and 16 illustrate the histories of pan-tilt and pan-zoom controls during the tracking, respectively. The number i in the figures means the ith observation. The vertical axis of Fig. 16 denotes the horizontal view angle, which is inversely proportional to the zooming factor.

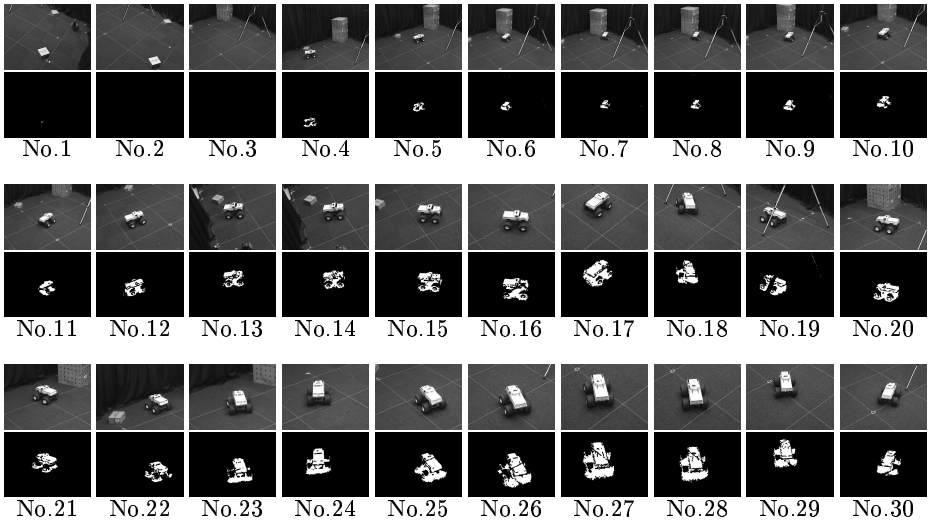


Fig. 14. Images observed during tracking(Upper:input images, lower:detected object silhouettes).

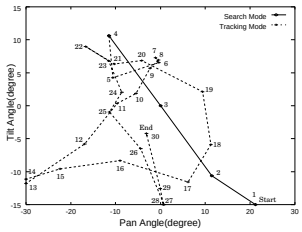


Fig. 15. History of pan-tilt control.

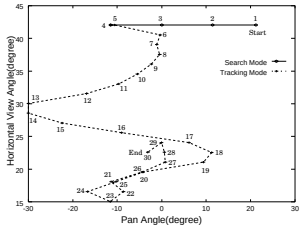


Fig. 16. History of pan-zoom control.

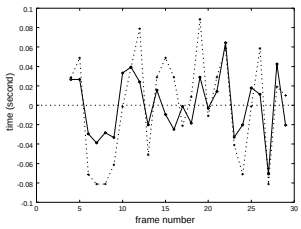
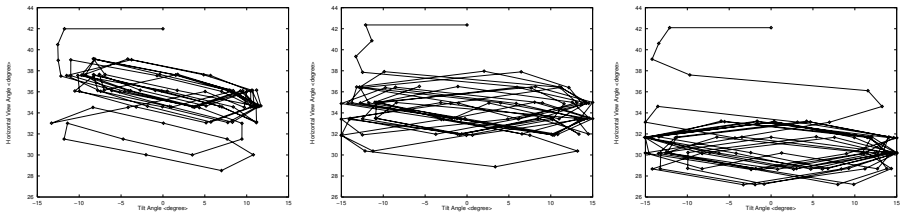


Fig. 17. Dynamics of the system (see text).

The entire tracking period is 13.77 seconds (i.e. about 2.1 image-acquisitions / second in average). Fig. 17 illustrates the dynamics of the image acquisition timing control. The solid line denotes the timing error, i.e. the difference between the predicted and practical image acquisition times. It almost stayed less than ± 0.05 sec, the inevitable temporal fluctuation involved in the mechanical camera motion. The dotted line shows the time interval between a pair of consecutive image acquisitions, where 0 denotes the average. These results verify that the adaptive system dynamics is realized depending on the target motion and the camera action.

To evaluate the effectiveness of the proposed dynamic control method, we conducted the following comparative study. The car is controlled to move continuously along almost the same circular track. Three FV-PTZ cameras, placed



Method 1 Method 2 Method 3 (proposed method)

Fig. 18. Performance evaluation: histories of tilt-zoom controls.

at almost the same position and with almost the same viewing direction, simultaneously track the car. The following three control methods are employed respectively.

Method 1 : Control the view direction to $(P_{obj}(t_0), T_{obj}(t_0))$, i.e. observed target location, without taking into account the target motion and the camera dynamics. The next image acquisition is done when the camera almost stops.

Method 2 : Control the camera view direction by predicting the target motion while assuming the camera dynamics is constant. In the experiment, the camera motion is assumed to complete in 0.5 sec.

Method 3 : The proposed method.

Note that all these three methods share the same zoom control method described before.

Fig. 18 illustrates the histories of the tilt-zoom controls by these three methods. As is obvious from the figure, the more sophisticated control is employed, the larger zooming factor is attained; the average horizontal view angles (the vertical axis of the figure) are 35.7 °, 34.4 °, and 31.2 ° respectively. Considering the zoom control method, the larger zooming factor implies the less estimation error. This quantitatively verifies that the proposed dynamic control method is effective in moving object tracking as well as in capturing high-resolution object images.

4.5 Dynamic Scene Visualization by an FV-PTZ Camera

As is seen from the image sequence in Fig. 14, while the images taken by the tracking system nicely capture the target in very high resolutions, human viewers cannot understand the global target trajectory or the surrounding scene configuration. That is, foveated images are not enough for dynamic scene visualization.

[17] showed that the dynamic integration of foveated and peripheral views greatly facilitates tele-operations. They used a wide angle fixed camera for the global scene visualization and a pan-tilt-zoom camera for the local object visualization. 3D camera calibration establishes the correspondence between the images taken by these two cameras.

Using an FV-PTZ camera, on the other hand, we can easily realize seamless integration of foveated and peripheral views. That is, as is obvious from Fig. 11, foveated dynamic object images captured by the tracking system can be back-projected onto the APP image, which gives the peripheral view of the global scene. In other words, foveated and peripheral views are integrated on the APP. Fig. 19 shows an image sequence synthesized by this method, where a white quadrangle in each image illustrates the foveated person image captured by the tracking system.

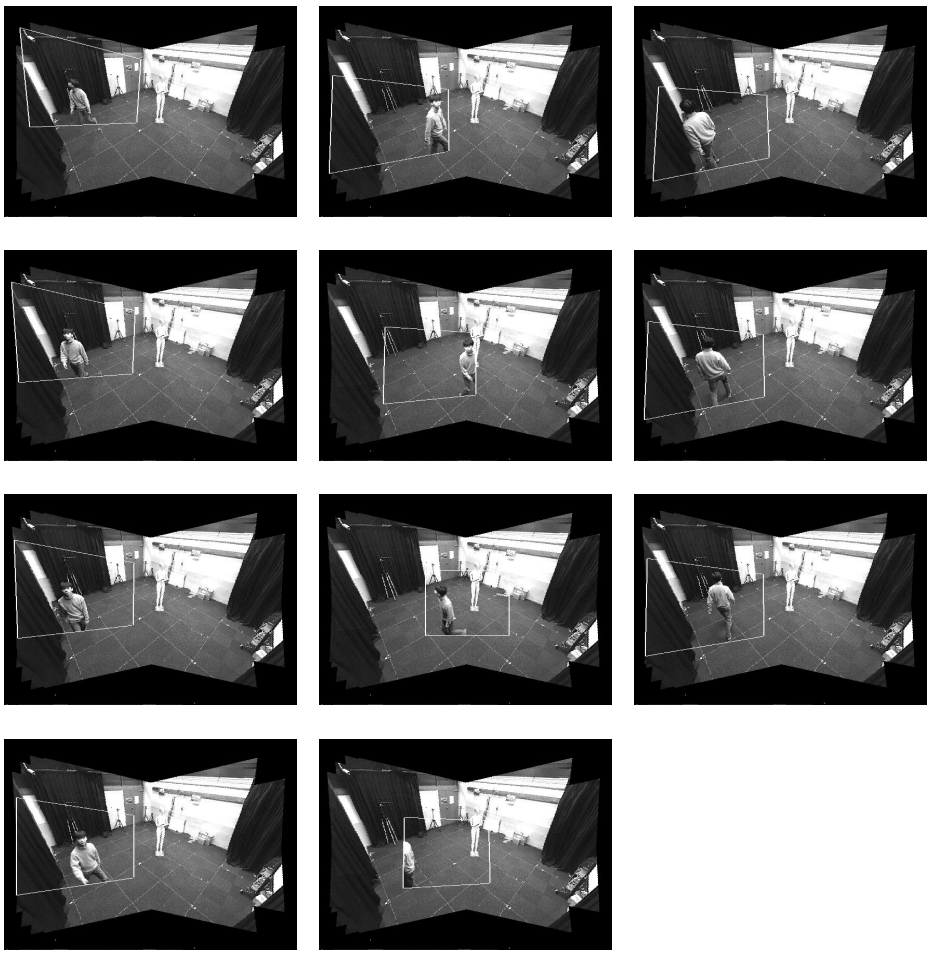


Fig. 19. Integrated foveated and peripheral views on APP. The sequence starts from the top-left and goes down followed by the next right column.

5 Multi-Viewpoints Object Imaging by Cooperative Observation Stations

To realize versatile visualization of complex dynamic scenes, we have to employ a group of observation stations which cooperatively track objects and capture multi-viewpoints object images;

- Obstacles and other moving objects often interfere the view from a camera.
- Without specialized video cameras like [18], it is difficult to obtain 3D object information by a single camera.

Here we call an observation station with visual perception, camera action control, and network communication capabilities *Active Vision Agent* (AVA in short).

This section addresses a multi-AVA system (i.e. a group of communicating AVAs) which cooperatively detects and tracks a focused target object to obtain its 3D information. The task of the system is specified as follows: 1) Each AVA is equipped with an FV-PTZ camera and mutually connected via the communication network. 2) Initially, it searches for a moving object independently of the others. 3) When an AVA detects an object, it navigates the gazes of the other AVAs toward that object (Fig. 20). 4) All AVAs keep tracking the focused target cooperatively to measure its 3D information without being disturbed by obstacles or other moving objects (Fig. 21). 5) When the target goes out of the scene, the system returns back to the initial search mode.

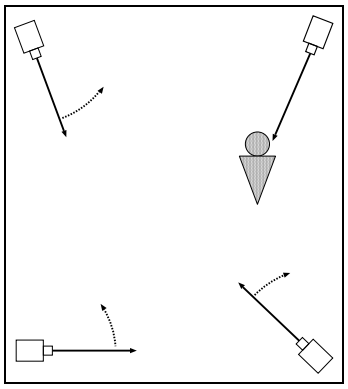


Fig. 20. Gaze navigation

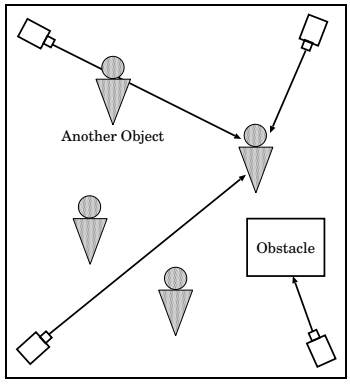


Fig. 21. Cooperative gazing

The object detection and tracking by each AVA is realized by the same method as described in Section 4. We assume that while all FV-PTZ cameras are calibrated, 3D geometric configurations of the scene and obstacles are not

known a priori. This is because the widely distributed camera arrangement makes it hard to employ stereo matching.

5.1 Integrating Visual Perception, Action, and Communication for Cooperative Object Tracking

In the cooperative object tracking, the following interactions among perception, action, and communication modules should be realized:

1. When no object appears in the scene, each AVA should search for an object autonomously by repeating its own perception-action cycle.
2. To realize the gaze navigation (Fig. 20), the camera actions of those AVAs which have not detected the target should be controlled by the information transmitted from the AVA that detected the target. This implies that the communication module in an AVA should be able to control its action module directly.
3. To realize the cooperative gazing (Fig. 21), the object identification should be established across multiple AVAs. Since all cameras are calibrated, if multiple AVAs capture object images simultaneously, the 3D location of the object can be computed, based on which the object can be identified. That is, for the object identification, the perception module of each AVA should be synchronized. Such synchronization is to be realized by communication among AVAs. Thus, the communication module in an AVA should be able to control its perception module directly.

Based on these considerations, we took the integration scheme where the communication module subordinates the perception and action modules.

5.2 Cooperative Object Tracking Protocol

In the above mentioned scheme, the design of the communication protocol becomes of the first importance in the system development. In designing the protocol, in turn, the ontology used for describing messages should be determined. Here we first propose a novel representation of the target object in the multi-AVA system, *Agency*, and then describe a cooperative object tracking protocol in terms of the agency.

Target Object Representation The most important ontological issue in the cooperative object tracking is how to represent the target object being tracked. In our multi-AVA system, "agent" means an AVA with visual perception, action, and communication capabilities. The target object is tracked by a group of such AVAs, whose perceptions and actions are tightly coupled (e.g. synchronized) by inter-AVAs communications.

Based on this consideration, we represent the target object by an *agency*, a group of those AVAs that are observing the target. With this object representation, specialized communication methods can be employed in the intra-agency

communication: high-speed and low-latency communication methods to realize real-time synchronized behaviors of the member AVAs in the agency.

The above definition of the agency implies that the agency is not a static data structure but a dynamic entity with its own dynamics. We define its dynamics by the following two protocols:

Agency Formation Protocol: how and when the agency is formed.

Role Assignment Protocol: what roles the member AVAs in the agency take to cooperate.

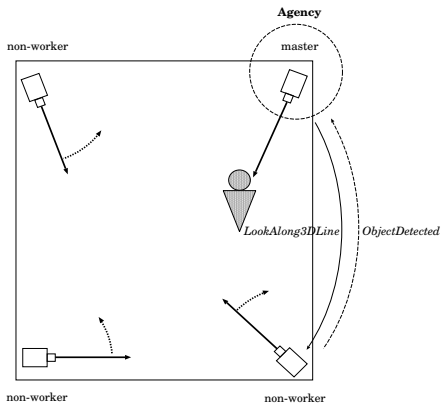


Fig. 22. Agency formation.

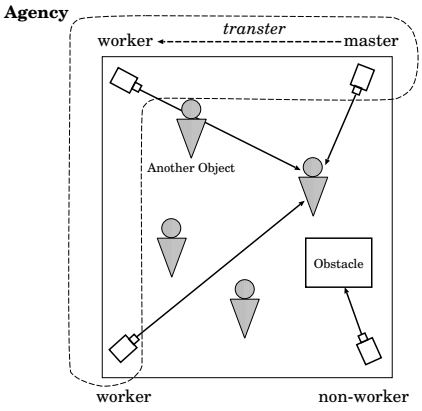


Fig. 23. Role assignment.

Agency Formation Protocol Specifically speaking, the task of the prototype system is to track cooperatively by all AVAs such object that is first detected. That is, while multiple moving objects can appear in the scene, the system tracks just one of them without paying any attention to the others. This task specification greatly simplifies the agency formation protocol.

1. Agency Generation Suppose no agency is generated yet. Note that as will be explained below, all AVAs know whether or not an agency is formed already. When AVA_i detects an object, it broadcasts the object detection message. If no other AVAs detect objects, then AVA_i generates an agency consisting of itself alone (Fig. 22). When multiple object detection messages are broadcast simultaneously, AVA_i can generate an agency only if it has the highest priority among those AVAs that have detected objects. That is, even if multiple AVAs detect objects simultaneously, which may or may not be the same, only one of them is allowed to generate an agency.

2. Joining into the Agency Once AVA_i has generated an agency, the other AVAs can know it by receiving the object detection message broadcast from AVA_i. Then they stop the autonomous object search and try to join into the agency.

Gaze Navigation :After generating an agency, AVA_i broadcasts the 3D line, L_i, defined by the projection center of its camera and the object centroid in the observed image. Then, the other AVAs search for the object along this 3D line by controlling their cameras respectively (Fig. 22).

Object Identification : Those AVAs which can successfully detect the same object as AVA_i are allowed to join into the agency. This object identification is done by the following method. Suppose AVA_j detects an object and let L_j denote the 3D view line directed toward that object from AVA_j. AVA_j reports L_j to AVA_i, which then examines the nearest 3D distance between L_i and L_j. If the distance is less than the threshold, a pair of detected objects by AVA_i and AVA_j are considered as the same object and AVA_j is allowed to join the agency.

Object Tracking in 3D : Once multiple AVAs join the agency and their perception modules are synchronized, the 3D object location can be estimated by computing the intersection point among a group of 3D view lines emanating from the member AVAs. Then, the 3D object location is broadcast to the other AVAs which have not detected the object. The communication protocol among the member AVAs in the agency will be described later.

3. Exit from the Agency When the object goes behind an obstacle, some AVA in the agency may fail to track it. Then, such AVA exits from the agency and again searches for the object guided by the information broadcast from the agency. When all AVAs in the agency loose the object (e.g. when the object goes out of the scene), the agency dies out.

Role Assignment Protocol Once the agency is formed, its member AVAs works cooperatively to track the target object. To realize efficient cooperation among the member AVAs, we assign them different roles depending on situations. Here we address the role assignment protocol by which the role of each member AVA is specified. Note that since situations change dynamically, the roles of member AVAs are to be changed dynamically through mutual communications.

Since the agency represents the target object being tracked, it has to maintain the object motion history, which is used to guide the search of non-member AVAs. Such object history maintenance should be done exclusively by a single AVA in the agency to guarantee the consistency. We call the right of maintaining the object history the *master authority* and the AVA with this right the *master* AVA. The other member AVAs in the agency without the master authority are called *worker* AVAs and AVAs outside the agency *non-worker* AVAs (Fig. 23).

The transition between worker and non-worker is defined before in the agency formation protocol. So what we have to specify here is the protocol to transfer the master authority.

When an AVA first generates the agency, it immediately becomes the master. The master AVA conducts the object identification described before to allow other AVAs to join the agency, and maintains the object history. All these processings are done based on the object information observed by the master AVA. Thus, the reliability of the information observed by the master AVA is crucial to realize robust and stable object tracking. In the real world, however, no single AVA can keep tracking the object persistently due to occluding obstacles and interfering moving objects.

The above discussion leads us to introducing the dynamic master authority transfer protocol. That is, the master AVA always checks the reliability of the object information observed by each member, and transfers the master authority to such AVA that gives the most reliable object information (Fig. 23).

The reliability can be measured depending on observed object characteristics (size, speed), scene situations (occluding objects, local lightings), AVA's visual perception capabilities (size of view field, view direction) and action characteristics (camera head speed), and so on. The prototype system employs a simple method: the master AVA transfers the authority to such member AVA whose object observation time is the latest in a predefined time period, since the latest object information may be the most reliable. Note that using this role assignment protocol, the master authority is continuously transfered around among member AVAs.

5.3 Implementation by a State Transition Network

Fig. 24 illustrates the state transition network designed to implement the above mentioned cooperative object tracking protocols. The network specifies event driven asynchronous interactions among perception, action, and communication modules as well as communication protocols with other AVAs, through which behaviors of an AVA emerge.

In Fig. 24, state i in the double circles denotes the initial state. Basically the states in rectangular boxes represent the roles of an AVA: master, worker, and non-worker. Since the master AVA conducts several different types of processing depending on situations, its state is subdivided into many substates. Those states in the shaded area show the states with the master authority. Each arrow connecting a pair of states is associated with the condition under which that state transition is incurred. ε means the unconditional state transition.

The right side of the figure shows what kind of processing, i.e. perception, action, receive, or send, is executed at each state. Those state in double rectangular boxes denote the states where perception is executed, while at those states in triple rectangular boxes, the camera action is executed. Thus, each state has its own dynamics and dynamic behaviors of an AVA are fabricated by state transitions.

Note that the prototype system assumes that the communication network is free from failures and delays. More robust and real time communication protocols should be developed for real world applications.

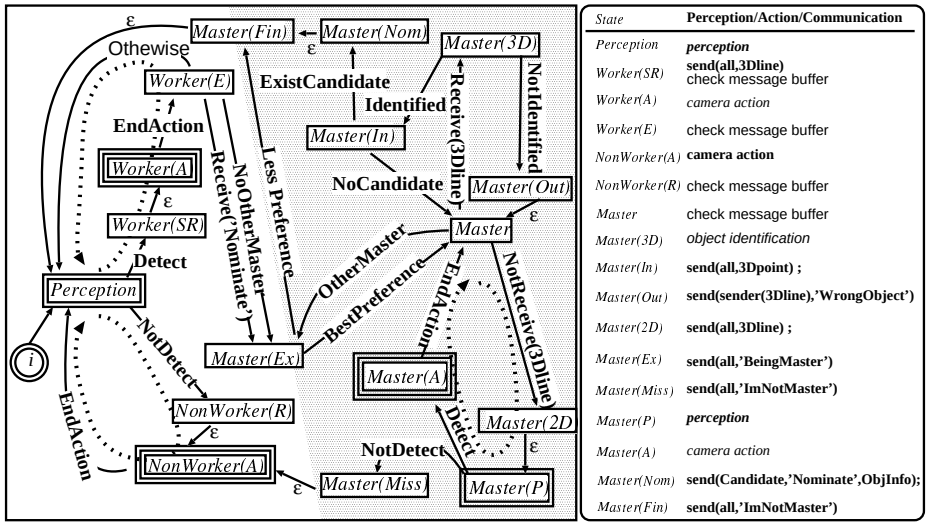


Fig. 24. State transition network for the cooperative object tracking.

5.4 Experimental Results

While the prototype system is far from complete, we conducted experiments to verify its potential performance. Two persons walked around a large box located at the center of the room (5m $\times$ 6m). Four FV-PTZ cameras are placed at the four corners of the room respectively, looking downward obliquely from about 2.5m above the floor. The person who first entered in the scene was regarded as the target. He crawled around the box not to be detected by the cameras. The other person walked around the box to interfere the camera views toward the target person. Then, both went out from the scene and after a while, a new person came into the scene.

Fig. 25 illustrates partial image sequences observed by the four cameras, where the vertical axis represents the time when each image is captured. Each detected object is enclosed by a rectangle. Note that while some images include two objects and others nothing, the gaze of each camera is directed toward the crawling target person. Note also that the image acquisition timings of the four cameras are almost synchronized. This is because the master AVA broadcasts the 3D view line to or the 3D position of the target to the other AVAs, by which their perception processes are activated. This synchronized image acquisition by multiple cameras enables the computation of the 3D target motion trajectory (Fig 26).

Fig. 27 illustrates the dynamics of the system, the state transition histories of the four AVAs. We can see that the system exhibits well coordinated behaviors as designed. That is, the entire system works in the following three modes:

Mode 1: All AVAs are searching for an object.

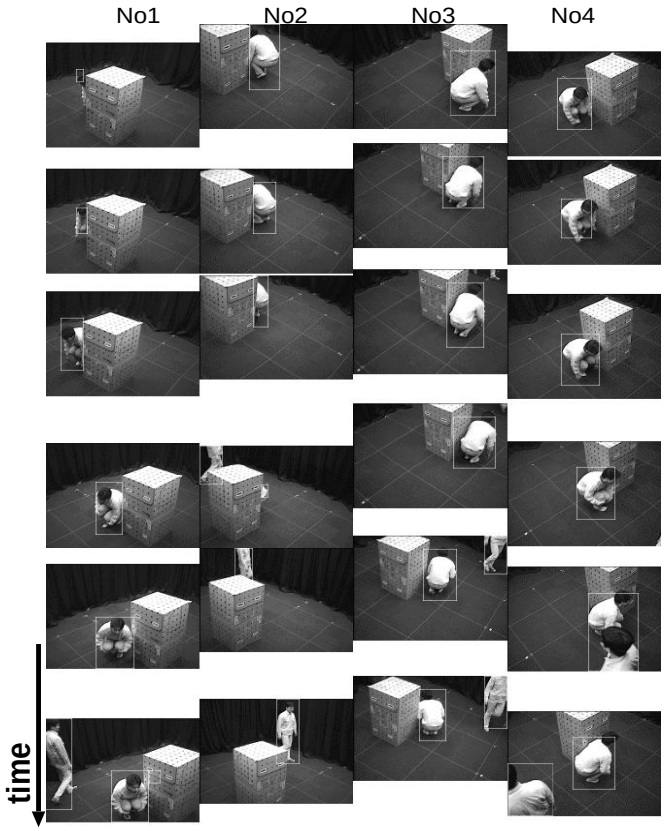


Fig. 25. Partial image sequences observed by four cameras. The vertical length of an image represents 0.5 sec.

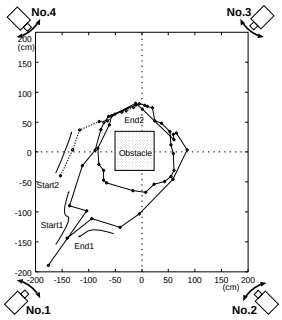


Fig. 26. 3D target motion trajectories.

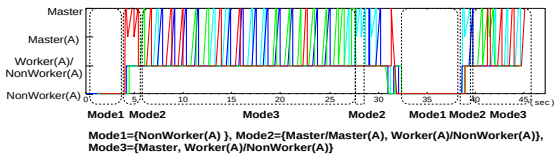


Fig. 27. State transition histories of the four AVAs.

Mode 2: The master AVA itself tracks the object since the others are still searching for the object.

Mode 3: All AVAs form the agency to track the object under the master's guidance.

The zigzag shape in the figure shows the continuous master authority transfer is conducted inside the agency.

Once a group of multi-viewpoints object images are obtained, we can generate a 3D object shape as well as measure its 3D location. We developed a sophisticated camera calibration method among widely distributed cameras and an efficient 3D shape reconstruction algorithm based on the 3D shadow volume intersection method[19]. Fig. 28 shows multi-viewpoints APP images of a mannequin and its reconstructed 3D shape. Currently we are developing a real time 3D shape reconstruction system using the multi-AVA system.

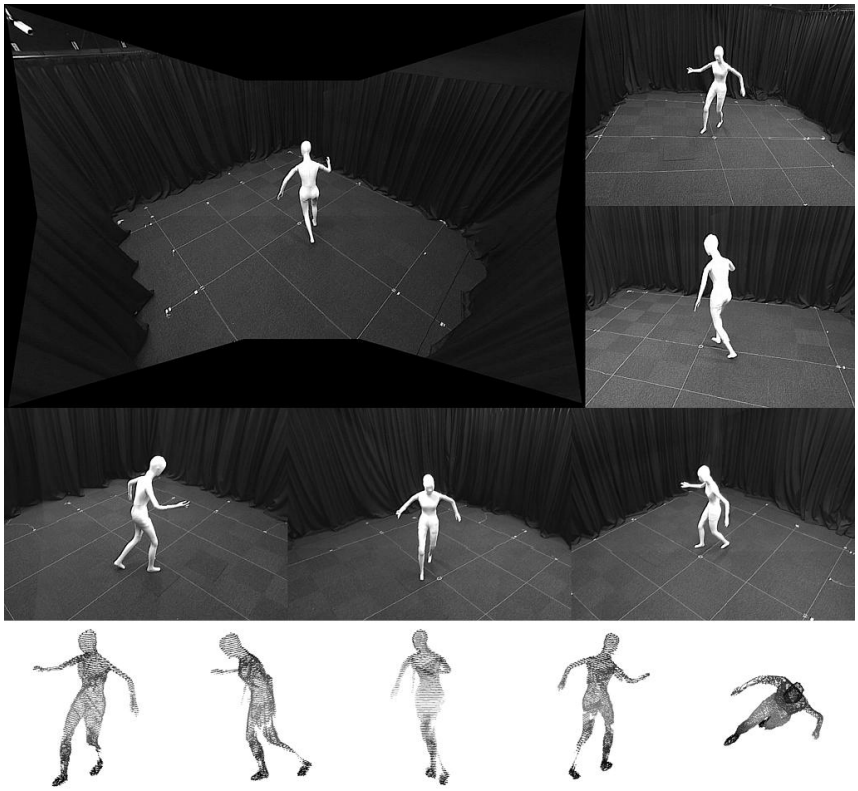


Fig. 28. Multi-viewpoints APP images and the reconstructed 3D shape of a mannequin. Top-left: entire APP image and windowed images extracted from the other five APP images.

6 Scenario-Based Cooperative Camera-Work Planning for Dynamic Scene Visualization

6.1 Camera-Works for Intelligible and Attractive Scene Visualization

Most of active vision systems developed so far including the ones described in Sections 4 and 5 capture images to control cameras and understand scene structures. This section, on the other hand, addresses active camera control methods for dynamic scene visualization. There exists a large difference between these two tasks; while the former throws away observed image data after processing, the latter puts its focus upon how we can fabricate image sequences intelligible and attractive for human viewers.

Here, "intelligible" implies that viewers should be able to understand global / dynamic scene contexts from limited sequences of captured images. "Attractive" means fabricated image sequences should keep attracting viewers' interest without being felt tired or boring. If possible, moreover, they should be artistic.

As discussed in Section 1, CDV offers a fundamental framework for scene visualization as well as scene understanding. To realize versatile scene visualization, we have to solve the following problems:

Camera Layout : How many and where should we put a group of cameras?
Dynamic Camera Control : How should we control camera parameters dynamically?
Image Sequence Fabrication : How should we fabricate intelligible and attractive image sequence(s) from raw image data observed by the cameras?

By *camera-work planning* we mean methods to solve these problems.

Since the real world includes a wide spectrum of dynamic scenes and moreover, the intelligibility and attractivity are too abstract to define computationally, it is almost impossible to attain the meaningful camera-work planning without knowledge. The following three types of knowledge can be used for the camera-work planning (see Fig. 29):

Scenario Description : This specifies semantics and physical structures of the scene as well as dynamic events involved in the scene.
Story-Board Description : This specifies a group of characteristic snapshots in the image sequence(s) to be fabricated. That is, it defines the intelligibility and attractivity to be realized by the camera-work.
Know-Hows about Camera-Works : Many effective camera-works have been developed in cinematography[20]. They include a variety of camera layout, image framing, and camera switching techniques. We can use such know-hows for the camera-work planning.

Note that the first one is described in terms of abstract semantic and/or 3D physical scene features, the second 2D image appearances taking into account psychological effects onto human viewers, and the third includes transformation rules between them.

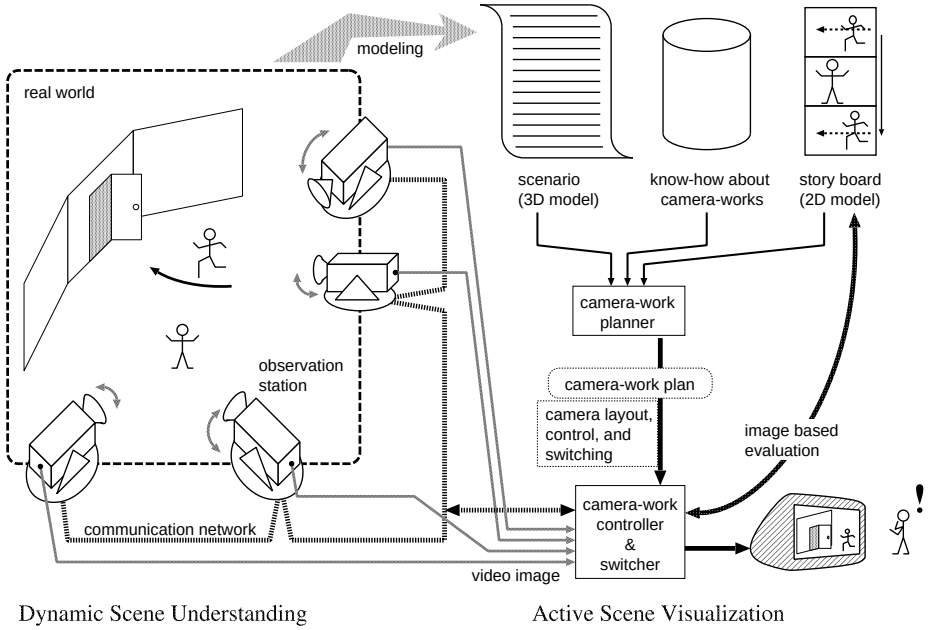


Fig. 29. Framework of the scenario-based cooperative camera-work planning for dynamic scene visualization.

Camera-work planning systems incorporate these three types of knowledge to solve the above mentioned three problems. In general, the planning should be done in the following two stages:

Off-Line Planning : Given a scenario description to be visualized, the system first makes a camera-work *plan* based on the knowledge.

On-Line Camera Control : Since the scenario is just a rough model of the real world scene, real world situations usually deviate from the scenario. Thus, on-line adaptive camera controls should be conducted during the scene visualization process. CDV systems such as those described in Sections 4 and 5 support such on-line adaptive camera controls.

Fig. 29 summarizes the framework for the dynamic scene visualization discussed above.

In this section, we describe a scenario-based dynamic scene visualization system being developed in the CDV project, where major emphasis is put upon dynamic cooperation between distributed cameras (i.e. observation stations). That is, we believe that to fabricate intelligible and attractive image sequence(s) from those observed by the cameras, flexible inter-camera coordinations are required as well as individual dynamic camera controls.

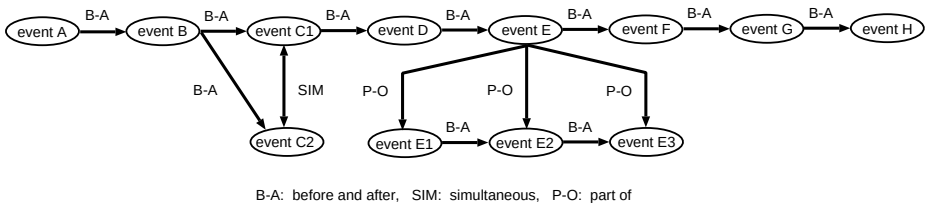


Fig. 30. Event graph.

6.2 System Organization

Here we describe specifications of each component of the scenario-based scene visualization system in Fig. 29.

Knowledge Sources As discussed before, three types of knowledge is give to the system:

1. Scenario Description There have been proposed several scenario/camera-work description methods and camera-work planning systems [21], [22], [23]. In [22], Christianson et al proposed the Declarative Camera Control Language, with which various types of camera-work patterns can be described. While the camera-work patterns can be used for the off-line planning, no mechanism is supported for the on-line camera control. In [21] and [23], on the other hand, on-line dynamic camera-work/interaction control methods are proposed. [21] used a state transition network to specify dynamic camera control and switching. [23] proposed a scripting method for interactive systems based on Allen's temporal interval algebra [24].

In our system, a scenario is described by an *event graph* (Fig. 30), where each node denotes an event representing the dynamic 3D model of a real world scene and an arc a temporal/geometric/semantic relation between events. The simplest but most popular event graph is a series of event nodes connected by a chain of directed arcs denoting the temporal order (i.e. B-A arcs in Fig. 30). Various types of semantic arcs, such as retrospection, hearsay, and illusion, may be used to enrich scenario contents.

An event node includes:

- Semantic Scene Features: type of the event and/or atmosphere of the scene, e.g. fighting, thrilling adventure, happy dining, solemn ceremony and so on.
- Background Scene Characteristics: overall geometric and illumination structures of the scene and their dynamic variations: e.g. soccer field, crowded downtown, conference room, and so on.

– Foreground Object Characteristics: attributes and dynamics of objects requiring focused imaging. Sometimes mental features and moods of objects may be associated with physical characteristics. For example, a tall man in a red shirt rushes out through the door crying loudly.

2. Story-Board Description This is described by a series of 2D sketches specifying how each *shot* in the finally fabricated image sequence looks like (Fig. 29). That is, it is the goal specification for the scene visualization. It contains the information about viewing angle, image framing, camera position, motion, and switching.

In addition, each sketch is associated with an event ID(s) in the event graph. Note that in general, associations between events and sketches are M : N. That is, it is very popular that an event is visualized by a series of shots taken from different cameras. We call a continuous video sequence taken by a camera *physical shot*. On the other hand, a single image frame sometimes includes multiple physical shots representing different events: e.g. a group of scientists in a conference room are discussing about the earth looking a video image transmitted from a satellite. We call such composite shot *logical shot*. That is, each sketch specifies characteristics of a logical shot, while an event corresponds to a physical scene.

3. Know-Hows about Camera-Works Since the story-board is just a rough and abstract goal specification, we need additional knowledge to attain the goal under the given scenario description. Know-hows about camera-works specify heuristic rules to take intelligible and attractive image sequences under various scenario situations. They include rules for camera layout, dynamic camera control, and switching. Note that camera control and switching rules specify not only actions of each individual camera but also coordination methods among distributed cameras.

Camera-Work Planning

Off-Line Planning Given three knowledge sources described above, the camera-work planner (Fig. 29) reasons about effective camera-works for the scene visualization.

1. First, for each event in a given scenario, the planner determines geometric camera layout, dynamic camera action, and temporal camera switching and coordination. Since there exist many different possible camera-works to visualize a given event, the planner uses sketches in a given story-board to select the most effective camera-work rule. Note that the camera-work plan generated at this stage specifies physical shots obtained by the cameras placed in the scene.
2. Then, the planner determines an image composition plan to fabricate logical shots specified in the story-board. Note that while most of logical shot

compositions are realized by 2D image processing, virtual images may be synthesized based on the 3D scene information restored from multi-viewpoints image sequences. Note also that the planner should make camera coordination plans across multiple events to generate well synchronized/organized logical shots.

On-Line Camera Control After placing a group of cameras according to the designed camera layout, the camera-work and logical shot composition plans are loaded onto a group of observation stations and the camera-work controller & switcher respectively (Fig. 29). Then,

1. The cameras stand by and objects in the real world start the actions specified in the scenario.
2. Each observation station captures an image sequence by controlling camera parameters according to the camera-work plan. The acquired image sequence is delivered to the controller & planner. As noted before, each observation station should adaptively control its camera since the scene usually deviates spatially and temporally from the plan. Moreover, multiple observation stations should cooperate with each other through communications to control their cameras. These adaptive and cooperative camera controls are realized using sketches in the story-board as goal specifications.
3. In the camera-work controller & switcher , a series of logical/physical shots are fabricated from a group of raw image sequences captured by observation stations. Note that the camera-work controller & switcher itself may generate virtual image sequences based on the 3D scene information restored from multi-viewpoints image sequences. Thus, it should dynamically communicate with observation stations to realize smoothly connected and/or well synchronized logical shots. The smoothness and synchronization are evaluated at the 2D image level referring the story-board (Fig. 29). In this sense, we may call it a director and/or a composer.

6.3 Prototype System

Currently we are developing a prototype system based on the framework proposed above. Here, we show two simulation results of the camera-work planning: (1) camera layout for 2D static scenes including multiple objects and obstacles and (2) scenario-based camera control and switching in 3D dynamic scenes. Simulation results demonstrate that our approach is very promising.

Planning Layout of Multiple Cameras in 2D Static Scenes Where to place a group of cameras is one of major problems to be solved in the off-line camera-work planning. We developed an optimization method for the camera layout.

First we assume the followings:

- The scene is two dimensional and static.

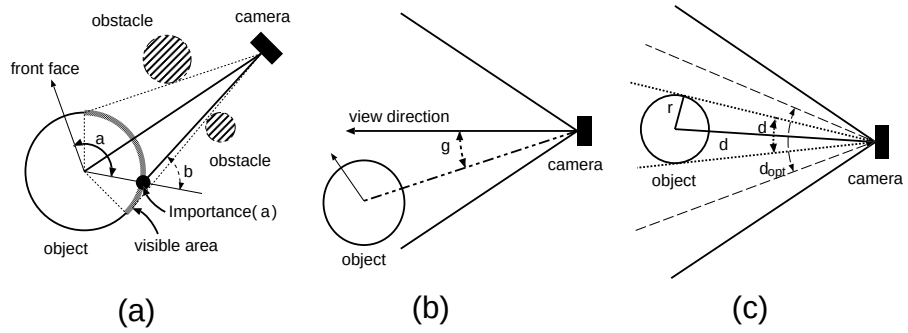


Fig. 31. Evaluation functions for the camera layout.

- The background scene is defined as a rectangular area, in which foreground objects, obstacles, and cameras are placed (Fig. 31(a)).
- An object is represented by a circle with a specific "front face" direction (Fig. 31(a)). Each point on the circular object surface is associated with a weight representing the importance for the visualization. In the current simulation, we used the following function to model the weight distribution over the surface:

$$Importance(\alpha) = \frac{1 + \cos(\alpha)}{2}, \tag{6}$$

  where α denotes the angular distance from the front face direction.
- An camera is modeled by a projection center with a fixed viewing angle (i.e. fixed zoom, Fig. 31(b)(c)). Note that this angle specifies the size of the image frame (i.e. area covered by an image). In addition, each camera is associated with a list of foreground objects to be imaged.

First, the size and shape of the background scene, locations and characteristics of objects and obstacles, and the number and viewing angles of cameras are given to the camera layout planner. Then, the position and viewing direction of each camera, (x, y, θ), is determined by optimizing the following evaluation function.

$$E_{total}(x, y, \theta) = \sum_{i \in Object-List} \{E^i_{visibility}(x, y, \theta) \times E^i_{position}(x, y, \theta) \times E^i_{size}(x, y, \theta)\}, \tag{7}$$

where $Object - List$ denotes a list of the objects to be imaged by the camera.

Each component evaluation function is defined for an object-camera pair as follows:

$E_{visibility}$: Object Surface Visibility This evaluates how well the object looks in the captured image. We used the following function (see Fig. 31(a)):

$$E_{visivility}(x, y, \theta) = \int_{visible} Importance(\alpha)\ \cos\beta\ \cos\alpha\ d\alpha, \tag{8}$$

where α denotes the angular distance of a surface point from the front face direction and β the angle between the surface normal at that point and the view direction from the camera. The integral covers only those surface points that can be seen from the camera without being interfered by obstacles.

$E_{position}$: Object Position in the Image We assume the object is to be captured at the center of the image. Then, the following function evaluates the goodness of the object position (see Fig. 31(b)):

$$E_{position}(x, y, \theta) = \frac{1}{2}(1 + \cos\gamma), \tag{9}$$

where γ denotes the angle between the object center direction from the camera and the view direction of the camera.

E_{size}: Object Size in the Image The object size in the captured image is also an important factor in visualization. We assume that a certain optimal object size is specified in the story-board. Then the following function evaluates the goodness of the object size (see Fig. 31(c)):

$$E_{size}(x, y, \theta) = \begin{cases} \frac{1}{2}(1 + \cos\frac{\delta - \delta_{opt}}{\delta_{opt}}\pi) & \text{if } \delta \leq \delta_{opt} \\ \frac{1}{2}(1 + \cos\frac{\delta - \delta_{opt}}{\pi - \delta_{opt}}\pi) & \text{if } \delta_{opt} < \delta \leq \pi \end{cases} \tag{10}$$

where $\delta = 2\sin^{-1}(r/d)$, r denotes the radius of the object, d the distance between the camera and the object, and δ_{opt} the pre-specified optimal size parameter.

We conducted several simulations to examine the effectiveness of the above mentioned camera layout method. Fig. 32(a) illustrates the geometric configuration of a pair of objects to be visualized. Fig. 32(b) shows (1) the spatial distribution of $E_{total}(x, y, \theta)$ and (2) the optimal camera position and its viewing direction when both objects are required to be imaged simultaneously by a single camera. To depict (1), the optimal view direction, θ^*, is first computed at each position and $E_{total}(x, y, \theta^*)$ is encoded by the gray level: the brighter the gray level is, the higher value the evaluation function takes. (2) is depicted by a group of three line segments: their intersection point denotes the camera position, the central segment the view direction, and the pair of marginal ones the viewing angle. Figs. 32(c) and (d) show the optimal camera layouts when each object is required to be imaged by a single camera, respectively.

Fig. 33 illustrates the optimal layout of a pair of cameras when camera-A and camera-B are used for imaging {object-0, object-1} and {object-2, object-3}, respectively, where $\{\cdot\}$ denotes the list of objects to be imaged by a camera.

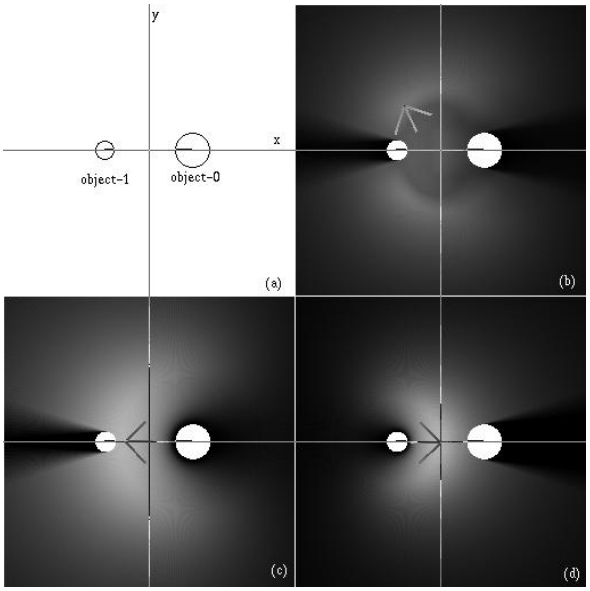


Fig. 32. Optimal camera layouts: (a) a pair of objects to be visualized, (b) optimal camera layout for simultaneous imaging of the objects, (c),(d) optimal camera layouts for object-0 and object-1, respectively.

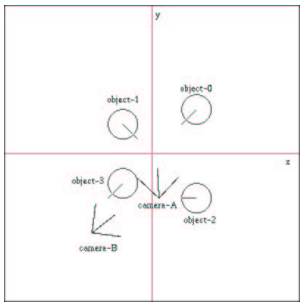


Fig. 33. Optimal layout of a pair of cameras for four objects.

While these simulation results are simple and include many points to be improved, we believe they showed practical utilities of our framework. Currently we are developing a novel camera layout method which utilizes the story-board as the evaluation function.

Dynamic Camera Coordination for Smooth Camera Switching Here we demonstrate the importance of the on-line coordinated camera control and switching in visualizing dynamic scenes.

Suppose a scenario description specifies that "A man is running along the long straight path at the constant speed." and the story-board requires that his zoomed-up face should be captured continuouly since changes of his facial expressions are the crucial factor for visualizing the scene.

Based on these knowledge sources, the camera-work plan illustrated in Fig. 34 is generated at the off-line planning stage. The plan specifies (1) a pair of cameras are placed at the same side along the path, (2) each camera tracks the face by dynamically rotating the view direction[4] , and (3) the image sequence taken by camera-1 should be switched to that taken by camera-2 when both image sequences can be smoothly connected. Here we assume the smoothness is

[4] For simplicity, we assume only the 2D panning is allowed for each camera.

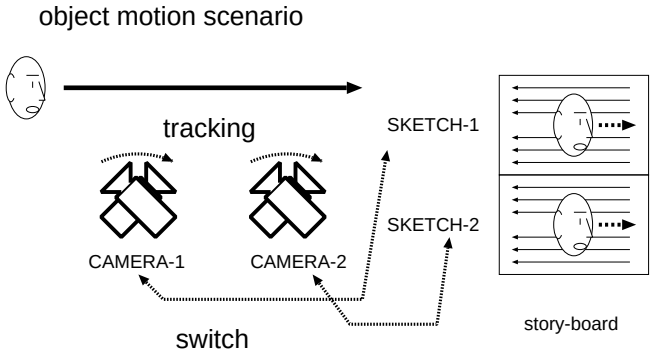


Fig. 34. Camera-work plan for dynamic scene visualization.

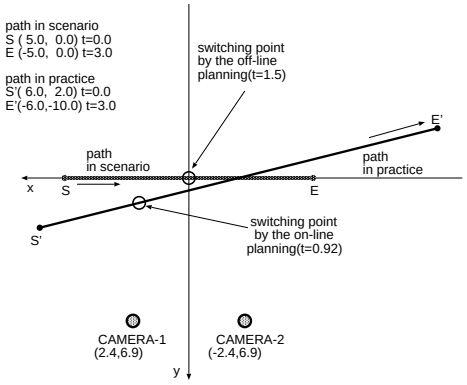


Fig. 35. Deviation of object motion.

evaluated by the apparent face motion against the backgound scene in captured image sequences.

This camera-work plan is loaded onto a pair of observation stations and the camera-work controller & switcher in Fig. 29. When the action in the scene is started, the object detection and tracking process such as described in Section 4 is executed at each observation station. Then, the camera-work controller & switcher monitors a pair of image sequences captured by the observation stations and determines the optimal camera switch timing.

As noted before, the actual scene usually deviates spatially and temporally from the scenario. Fig. 35 illustrates the geometric configuration of the scene, the camera layout, and the object motion path described in the scenario. Here we assume that the actual object motion path deviates from the plan as shown

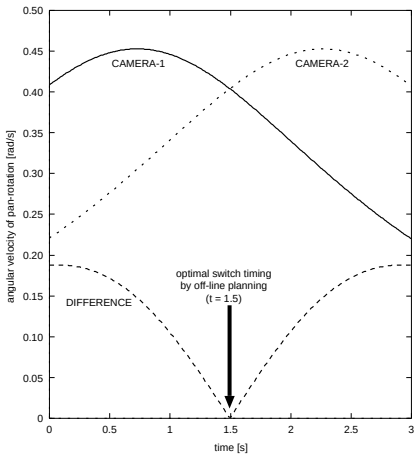


Fig. 36. Optimal camera switch timing determined by the off-line camera-work planning.

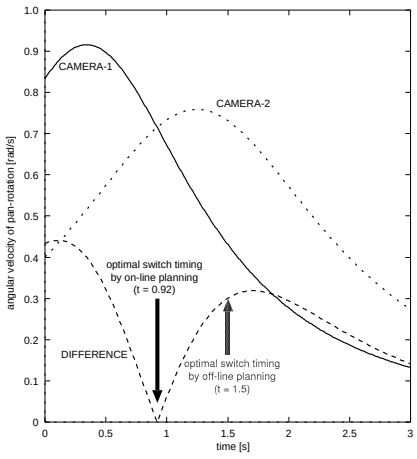


Fig. 37. Optimal camera switch timing determined by the on-line camera control.

in the figure. In what follows, we will demonstrate the importance of the on-line camera control in determining the optimal camera switch timing.

In the current simulation, the switch timing is evaluated by the difference in the camera rotation speed. The reason for this is as follows. Firstly, since both cameras are tracking the object, the object image stays fixed at the center of the image frame. Thus, human viewers perceive the object motion speed based on the optical flow of the background scene. Assuming the distance of the background scene from the cameras is constant, the camera rotation speed uniquely determines the strength of the optical flow. In other words, by switching the cameras when their rotation speeds coincide with each other, human viewers perceive the object as moving at the constant speed even if the camera is switched from one to the other. Note that to realize more smooth camera switching, we should control the zoom so as to make the object sizes in the pair of captured image sequences coincide.

Fig. 36 illustrates temporal variations of the rotation speeds of camera-1 and camera-2 when they are tracking the object along the path specified in the scenario. The optimal camera switch timing is determined as $t = 1.5$ sec and the object location at that time is shown in Fig. 35. Fig. 38(a) shows the image sequence fabricated from the pair of image sequences taken by camera-1 and camera-2, assuming the object moves as specified in the scenario and the camera is switched at $t = 1.5$ sec.

If we directly applied this planned camera-work to the actual scene, we would obtain such a meaningless image sequence as shown in Fig. 38(b), which demonstrates the necessity of the on-line adaptive camera control.

Fig. 37 illustrates temporal variations of the rotation speeds of camera-1 and

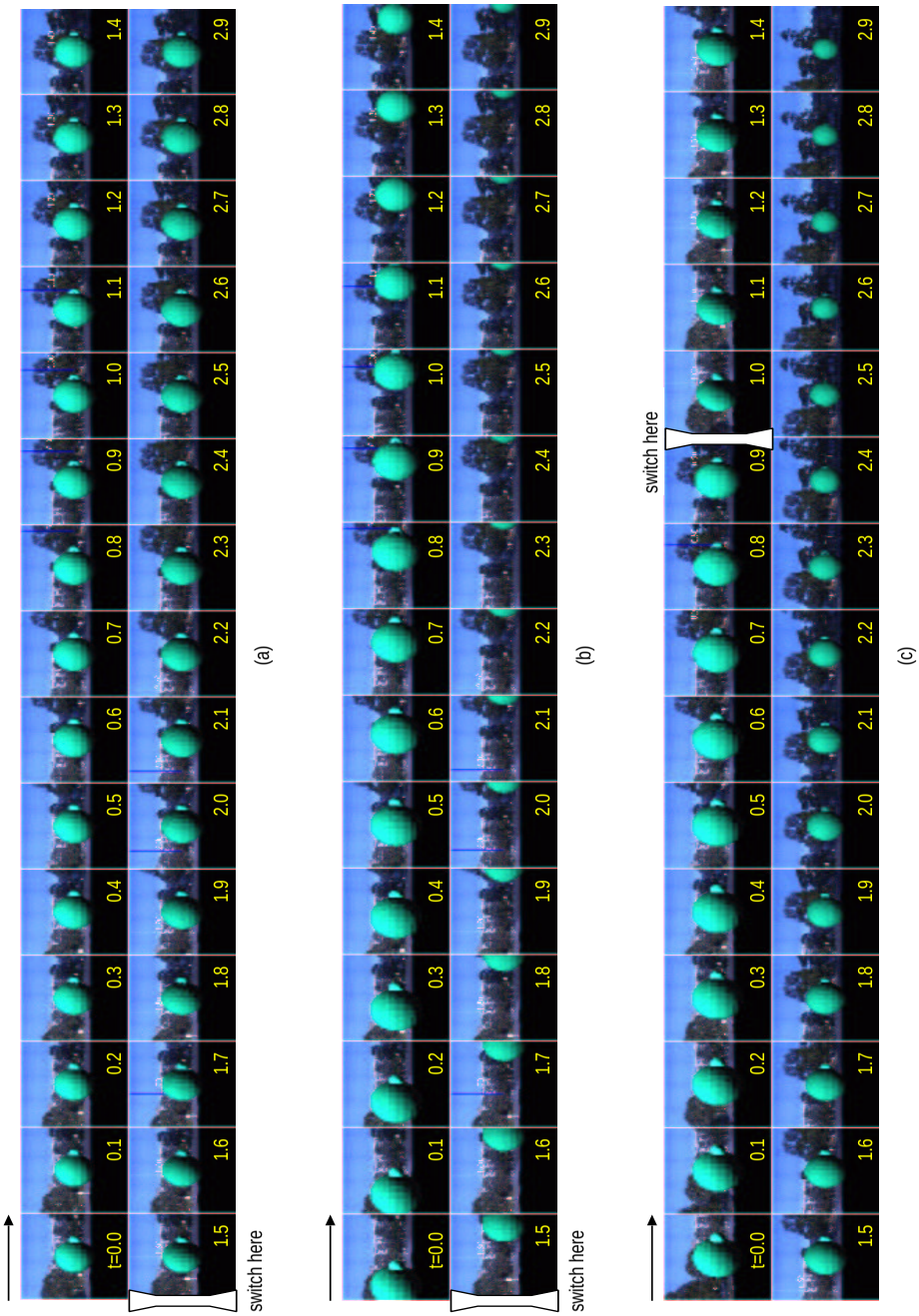


Fig. 38. Fabricated image sequences. Each sequence starts at the left of the upper row followed by the lower row including a mark denoting the point of the camera switching.

camera-2 when they are adaptively tracking the actual object motion shown in Fig. 35. The camera-work controller & switcher dynamically monitors these camera motion speeds and switches the cameras at $t = 0.92$ sec (see Fig. 37). Fig. 38(c) illustrates the image sequence fabricated by this on-line adaptive camera control and switching method, where the soomthly connected image sequence is fabricated.

6.4 Discussions

In this section we proposed a framework of senario-based cooperative camera-work planning for dynamic scene visualization. Its novel features are

- Introduction of three types of knowledge sources: scenario, know-hows about camera-works, and story-borad.
- Off-line camera-work planning followed by on-line dynamic camera control and switching.
- Cooperation among distributed active cameras (i.e. observation stations) to adaptively capture intelligible and attractive image sequences.
- Logical and virtual image shots fabrication from multi-viewpoint image sequences.

While we have shown practical utilities of our approach with several simulations, the following technical developments are required to implement a scene visualization system that can work in real world scenes.

- Description languages for the knowledge sources and the camera-work plan
- Knowledge-based camera layout and dynamic camera-work planning for 3D dynamic scenes
- Plan-guided dynamic camera control for scene/object visualization
- Dynamic cooperation protocols for well organized/synchronized multi-viewpoint visualization
- Image sequence switching and virtual image generation for intelligible and attractive image sequence fabrication
- Computational method of evaluating intelligibility and attractivity.

7 Concluding Remarks

This paper describes the idea and goal of our five years project on cooperative distributed vision and shows technical research results so far obtained on active image capturing and dynamic scene visualization: 1) fixed viewpoint pan-tilt-zoom camera for wide-area active imaging, 2) moving object detection and tracking for reactive image acquisition, 3) multi-viewpoints object imaging by cooperative observation stations, and 4) senario-based cooperative camera-work planning for dynamic scene visualization. Prototype systems demonstrate the effectiveness and practical utilities of the proposed methods.

The project holds annual international workshops, where research results are presented with working demo systems. All research results and activities of the project are shown in the homepage (URL: http://vision.kuee.kyoto-u.ac.jp/CDVPRJ).

This work was supported by the Research for the Future Program of the Japan Society for the Promotion of Science (JSPS-RFTF96P00501). Research efforts by all members of our laboratory and the assistance of Ms. H. Taguchi in preparing figures are gratefully acknowledged.

References

1. Matsuyama, T.: Cooperative Distributed Vision – Dynamic Integration of Visual Perception, Action, and Communication –, Proc. of Image Understanding Workshop, Monterey CA, 1998.11
2. Aloimonos, Y. (ed.): Special Issue on Purposive, Qualitative, Active Vision, CVGIP: Image Understanding, Vol.56, No.1, 1992.
3. Aloimonos, Y. (ed.): Active Perception, Lawrence Erlbaum Associates Publisher, 1993
4. Yagi Y. and Yachida M.: Real-Time Generation of Environmental Map and Obstacle Avoidance Using Omnidirectional Image Sensor with Conic Mirror, Prof. of CVPR, pp. 160-165, 1991.
5. Yamazawa K., Yagi Y. and Yachida M.: Obstacle Detection with Omnidirectional Image Sensor HyperOmni Vision, Proc. of ICRA, pp.1062 - 1067, 1995.
6. Peri V. N. and Nayar S. K.: Generation of Perspective and Panoramic Video from Omnidirectional Video, Proc. of IUW, pp.243 - 245, 1997.
7. Murray,D. and Basu,A.: Motion Tracking with an Active Camera, IEEE Trans. of PAMI, Vol. 16, No. 5, pp. 449-459, 1994.
8. Wada T. and Matsuyama T.: Appearance Sphere: Background Model for Pan-Tilt-Zoom Camera, Proc. of ICPR, Vol. A, pp. 718-722, 1996.
9. Lavest, J.M., Delherm, C., Peuchot, B, and Daucher, N.: Implicit Reconstruction by Zooming, Computer Vision and Image Understanding, Vol.66, No.3, pp.301-315, 1997.
10. Hall R.: Hybrid Techniques for Rapid Image Synthesis, in Image Rendering Tricks (Whitted T. and Cook R. eds.), Course Notes 16 for SIGGRAPH'86, 1986.
11. Greene N.: Environment Mapping and Other Applications of World Projections, CGA, 6 (11), pp. 21-29, 1986.
12. Chen S.E.: QuickTime VR – An Image-Based Approach to Virtual Environment Navigation, Proc. of SIGGRAPH'95, pp. 29-38, 1995.
13. Nakai, H.: Robust Object Detection Using A-Posteriori Probability, Tech. Rep. of IPSJ, SIG-CV90-1, 1994 (in Japanese).
14. Grimson, E.: A Forest of Sensors, Proc. of VSAM Workshop, 1997.
15. Davis, L.: Visual Surveillance and Monitoring, Proc. of VSAM Workshop, 1997.
16. Habe, H., Ohya, T., and Matsuyama, T.: A Robust Background Subtraction Method for Non-Stationary Scenes, Proc. of MIRU'98, Vol.1, pp.467-472, 1998 (in Japanese).
17. Yamaashi, K., Cooperstock, J.R., Narine, T., and Buxton, W.: Beating the Limitations of Camera-Monitor Mediated Telepresence with Extra Eyes, Proc. of CHI, pp.50-57, 1996.

18. Hiura, S. and Matsuyama, T.: Depth Measurement by the Multi-Focus Camera, Proc. of CVPR, pp.953-959, 1998
19. Mikoshi, Y.: 3D Image Measurement Based on Planes, Master Thesis, Kyoto University, 1998 (in Japanese)
20. Arijon, D.: Grammar of the Film Language, Focal Press Ltd., London, 1976
21. He,L., Cohen,M.F., and Salesin, D.H.: The Virtual Cinematographer: A Paradigm for Automatic Real-Tile Camera Control and Directing, SIGGRAPH'96, pp.217-224, 1996.
22. Christianson,D.B., Anderson,S.E., He,L., Weld,D.S., Cohen,M.F., and Salesin, D.H.: Declarative Camera Control for Automatic Cinematography, Proceedings of AAAI '96, pp.148-155, 1996.
23. Mase,K., Pinhanez,C.S., and Bobick, A.F.: Scripting Method Based on Temporal Intervals for Designing Interactive Systems, Trans. of IPSJ, Vol.39, No.5, pp.1403-1413, 1998 (in Japanese).
24. Allen, J.F.: Towards a General Theory of Action and Time, Artificial Intelligence, Vol.23, pp.123-154, 1984.

This article was processed using the LaTeX macro package with LLNCS style