Cooperative Distributed Vision

– Dynamic Integration of Visual Perception, Action, and Communication –

Takashi Matsuyama

Department of Intelligence Science and Technology
Graduate School of Informatics, Kyoto University
Sakyo, Kyoto, 606-8501 JAPAN
tm@i.kyoto-u.ac.jp

Abstract

This paper gives an overview of our five years project on *Cooperative Distributed Vision* (CDV, in short). 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 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. 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. In this paper, we first discuss functionalities of and mutual dependencies among perception, action, and communication to formally clarify the meaning of their integration. Then we present technical research results so far obtained on moving target detection and tracking by cooperative observation stations. Prototype systems demonstrate the effectiveness and practical utilities of our approach.

1 Introduction

This paper gives an overview of our five years project on *Cooperative Distributed Vision* (CDV, in short). The project was started from October 1996 under the support of the Research for the Future Program, the Japan Society for the Promotion of Science.

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

Embed in the real world a group of network-connected *Observation Stations* (real time 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.*

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, etc)

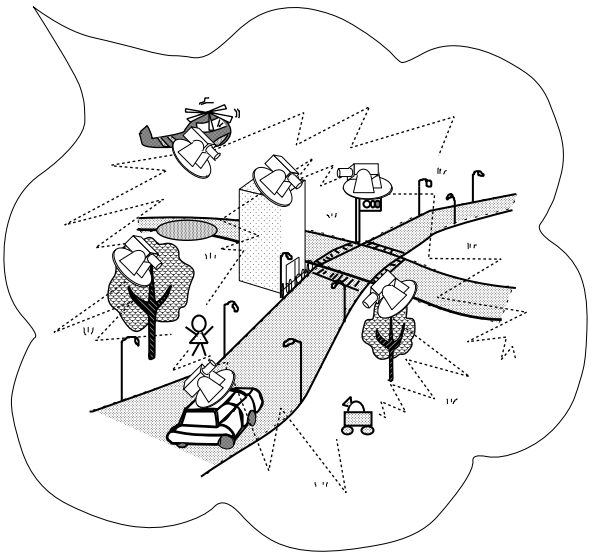


Figure 1: Cooperative distributed vision.

- Navigation and guidance of (non-intelligent) mobile robots and disabled people
- Cooperative mobile robots.

We believe CDV offers a fundamental scheme of computer vision systems in the 21st century.

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.

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 control systems.

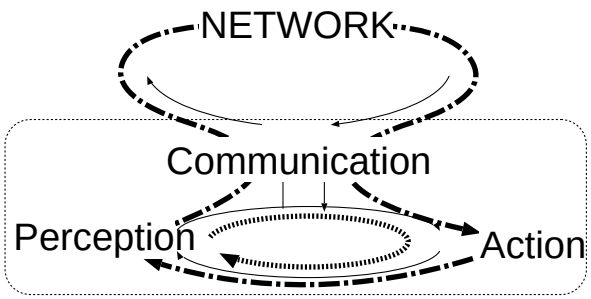


Figure 2: Information flows formed by integrating perception, action, and communication

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

In this paper, we first discuss functionalities of and mutual dependencies among perception, action, and communication to formally clarify the meaning of their integration. Then we present research results so far obtained on moving target detection and tracking by cooperative observation stations:

1. *Visual Perception*: Fixed-Viewpoint Pan-Tilt-Zoom (FV-PTZ) Camera for wide-area active imaging,
2. *Visual Perception* $\oplus$ *Action*[1] : Real time object detection and tracking by an FV-PTZ camera,
3. *Visual Perception* $\oplus$ *Action* $\oplus$ *Communication*: Cooperative object tracking by communicating active vision agents.

Prototype systems demonstrate the effectiveness and practical utilities of the proposed ideas.

2 Integrating Perception, Action, and Communication

2.1 Modeling Intelligence by Dynamic Interactions

To model intelligence, (classic) Artificial Intelligence employs the scheme

[1] The meaning of $\oplus$ will be explained later.

Intelligence = Knowledge + Reasoning

and puts its major focus upon symbolic knowledge representation and symbolic computation. In this sense, it may be called *Computational Intelligence*[1].

When we apply this scheme to real world problems, however, its competence is limited by the incompleteness of knowledge; we cannot describe all possible objects, their mutual relations, or dynamic situations in the world.

In CDV project, on the other hand, we propose an idea of *modeling intelligence by dynamic interactions*, which can be represented by the following scheme:

Intelligence =

Perception $\oplus$ *Action* $\oplus$ *Communication*,

where $\oplus$ implies dynamic interactions among the component modules.

That is, we define an *agent* as an intelligent system with perception, action, and communication capabilities and regard these three functions as fundamental modules to realize dynamic interactions between the agent and its outer world (i.e. scene and other agents):

Function		From		To
Perception	:	**World**	$\rightarrow$	**Self**
Action	:	**Self**	$\rightarrow$	**World**
Communication	:	**Self**	$\leftrightarrow$	**Others**

By integrating perception, action, and communication, various dynamic information flows are formed (Figure 2): for example,

- Perception–Action Cycle: **World** – *Perception* $\rightarrow$ **Self** – *Action* $\rightarrow$ **World**
- Communication Cycle: **Self** – *Communication* $\rightarrow$ **Other Agents** – *Communication* $\rightarrow$ **Self**.

In our model, reasoning implies the function which dynamically controls such flows of information.

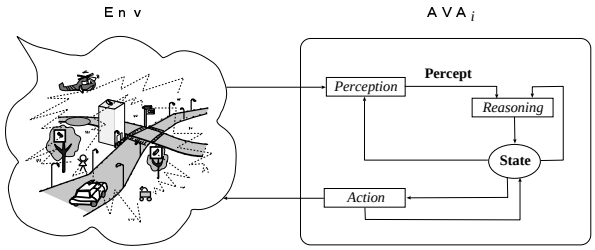


Figure 3: Primitive model of interaction between perception and action.

We believe that intelligence does not dwell solely in brain but emerges from active interactions with environments through perception, action, and communication. In other words, we do not regard intelligence as symbolically representable objects but as dynamic events fabricated by intermingled information flows.

2.2 Model of Active Vision Agent

2.2.1 Basic Functional Modeling

First we define *Active Vision Agent* (AVA, in short) as a *rational agent* with visual perception, action, and communication capabilities. Let $\mathbf{Sate}_i$ denote the internal state of ith AVA, AVA$_i$, and **Env** the state of the world where a group of AVAs are embedded.

Intuitively, perception[2] and action by AVA$_i$ can be modeled by the following mapping functions (Figure 3):

$$\mathit{Perception}_i \ : \ \mathbf{Env} \times \mathbf{Sate}_i \mapsto \mathbf{Percept}_i, \quad (1)$$

$$\mathit{Action}_i \ : \ \mathbf{Sate}_i \mapsto \mathbf{Sate}_i \times \mathbf{Env}, \quad (2)$$

where $\mathbf{Percept}_i$ stands for entities perceived by AVA$_i$. We introduce $\mathit{Reasoning}_i$ to link $\mathit{Perception}_i$ and Action_i. It means the perception-driven and/or autonomous internal state transition by AVA$_i$:

$$\mathit{Reasoning}_i : \mathbf{Percept}_i \times \mathbf{Sate}_i \mapsto \mathbf{Sate}_i. \quad (3)$$

Since $\mathbf{Sate}_i$ includes camera parameters such as camera position, viewing angle, zooming factor, focus, and so on, $\mathit{Perception}_i$ depends on $\mathbf{Sate}_i$; $\mathbf{Percept}_i$ changes depending on

[2] Here by 'perception' we mean visual perception.

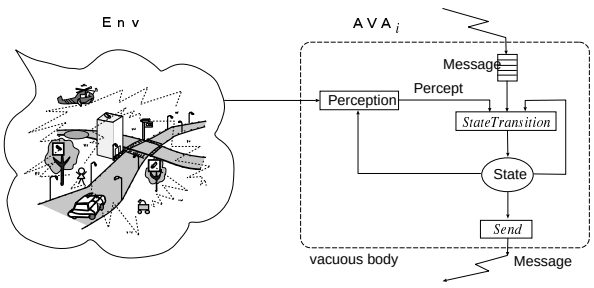


Figure 4: Model of a vacuous AVA.

$\mathbf{State}_i$. $Action_i$, on the other hand, can modify its own internal state $\mathbf{Sate}_i$ as well as the state of the world **Env**:

- AVA with a physically actionable body (e.g. a mobile robot with manipulators) can change both $\mathbf{Sate}_i$ and **Env**. We call such AVA *embodied* AVA. Equation (2) models such physical actions.

- When AVA is equipped only with active camera(s), on the other hand, its action implies the change of its internal state without any side effects on the world state[3] . We call such AVA *vacuous* AVA and model its action by

$$Action_i^{\mathrm{v}} : \mathbf{Sate}_i \mapsto \mathbf{Sate}_i. \qquad (4)$$

  We call the implementation of a vacuous AVA an *observation station*: real time image processor with active camera(s).

The discrimination between embodied and vacuous AVAs plays a crucial role in defining the meaning of communication.

2.2.2 Communication between Vacuous AVAs

The communication between vacuous AVAs can be defined by the following pair of message exchange functions:

$$Send_i : \mathbf{Sate}_i \mapsto \mathbf{Message}_j \qquad (5)$$

$$Receive_i : \mathbf{Message}_i \times \mathbf{Sate}_i \mapsto \mathbf{Sate}_i, \qquad (6)$$

[3] Strictly speaking, since active cameras in the real world have physical bodies, their actions can change the world state. But we neglect such exceptional cases.

where $\mathbf{Message}_i$ and $\mathbf{Message}_j$ denote messages sent out to AVA$_i$ and AVA$_j$ via the communication network, respectively. This model has the following characteristics:

- $Send_i$ does not depend on the receiver's state $\mathbf{State}_j$. That is, it represents an asynchronous message transfer and can support broadcast.

- $Receive_i$, on the other hand, depends on the receiver's state $\mathbf{State}_i$. That is, while a sender submits a message at its convenience, a receiver can accept and/or reject the message depending on it own state. To implement such deliberate message processing, a message buffer should be introduced (Figure 4).

We believe these asynchronous message processing functions should be supported in the communication between AVAs since they are autonomous agents with self-determination.

Based on the functional definitions given above, we can derive the following observations about interactions among perception, action, and communication of vacuous AVAs.

1. Comparing equations (3) and (4), the action by a vacuous AVA is nothing but a special case of the reasoning. This view leads us to a new model of action. As shown in equation (4), the essence of action is the state transition. This holds also in the case of embodied AVAs, because physical body actions can be modeled by state changes of mechanical parts. The world state transition caused by the action should be modeled by the *side-effect* of the action. Consequently, equation (2) is refined to

$$Action_i : \mathbf{Sate}_i \mapsto \mathbf{Sate}_i, \qquad (7)$$

$$ActionToWorld_i : (\mathbf{Sate}_i \mapsto \mathbf{Sate}_i) \mapsto \mathbf{Env}. \qquad (8)$$

The essential difference between embodied and vacuous AVAs rests in whether or not function $ActionToWorld_i$ is supported. Further implications of carrying a physical body will be discussed in the next section.

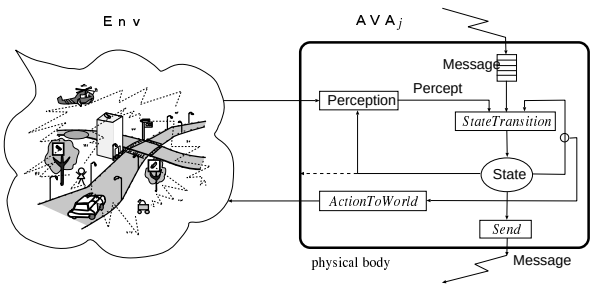


Figure 5: Model of an embodied AVA.

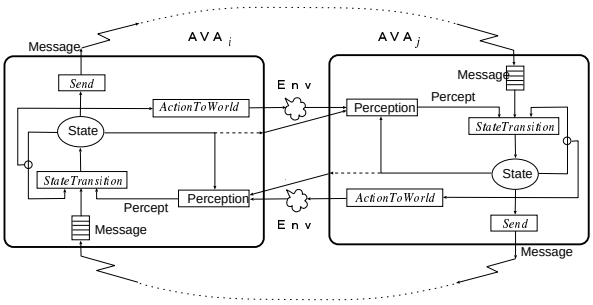


Figure 6: Multi-channel communication between embodied AVAs.

2. The state transition is caused by the action, the perception followed by the reasoning, and/or the message acception. Thus we can summarize these processes into

$$StateTransition_i\text{:}$$

$$\mathbf{Percept}_i \times \mathbf{Message}_i \times \mathbf{Sate}_i \mapsto \mathbf{Sate}_i. \quad (9)$$

Then, the behavior of a vacuous AVA can be modeled by equations (1), (5), and (9) (Figure 4).

2.2.3 Communication between Embodied AVAs

In the discussion so far presented, the distinguishing feature of embodied AVAs is characterized by $ActionToWorld_i$. Here we will show that carrying a physical body has further implications.

Let AVA$_j$ denote an embodied AVA (Figure 5). Then, (some parts of) $\mathbf{State}_j$ can be observed by other AVAs;

- Direct Body State Observation: Some parts of $\mathbf{State}_j$ define (i.e. are reflected on) the physical state of AVA$_j$'s body, which can be observed by other AVAs.

- Indirect Side-Effect Observation: Equation (8) shows that some parts of **Env** may indirectly reflect $\mathbf{State}_j$, which can be observed by other AVAs. For example, AVA$_j$ writes down its internal state on a blackboard, which then is observed by other AVAs.

Thus, $\mathbf{Percept}_i$ in equation (1) may include $\mathbf{State}_j$. To estimate $\mathbf{State}_j$ from $\mathbf{Percept}_i$, we define the following new function:

$$StateEstimation_i^j : \mathbf{Percept}_i \mapsto \mathbf{State}_j. \quad (10)$$

Note that in order for $StateEstimation_i^j$ to function meaningfully, AVA$_i$ must first identify AVA$_j$ and then should have the knowledge about how $\mathbf{State}_j$ is reflected onto its physical body and/or the world.

The above discussion implies that in the communication between embodied AVAs, their bodies and surrounding environments can be used as communication channels through which the information about AVAs' internal states is exchanged. That is, the *communication without message exchange* can be realized between embodied AVAs; $Send_j$ is simulated by $Action_j$ and $ActionToWorld_j$, and $Receive_i$ by $Perception_i$.

In summary, in the case of embodied AVAs, multi-channel communication links are formed by versatile combinations of perception, action and message exchange processes (Figure 6). The scientific goal of our project is to investigate how we can make a multi-AVA system intelligent by *dynamically* coordinating the information flow through such communication links. The three technical results described in this paper are a step toward this goal:

1. We first introduce *Fixed-Viewpoint Pan-Tilt-Zoom (FV-PTZ) Camera* to realize wide-area active imaging (Section 3).

2. Then, a real time object detection and

tracking system with an FV-PTZ camera is presented, where a dynamic interaction mechanism between perception and action is proposed (Section 4).

3. Finally, we present a cooperative object tracking system, where a group of communicating vacuous AVAs (i.e. observation stations) cooperatively track a focused target object. The system employs a state transition network to integrate perception, action, and communication (Section 5).

The key issue studied in these works is how we can introduce dynamics into the functional dependency model described in this section. A straightforward way to define dynamics would be to incorporate time variable t into the model. For example, equation (3) is augmented to

$$Reasoning_i(\mathbf{Percept}_i(t), \mathbf{Sate}_i(t)) = \mathbf{Sate}_i(t+\Delta t). \quad (11)$$

This type of formulation is widely used in control systems. In fact, Asada[26], a core member of the project, used linearized state equations to model mobile robot behaviors. We believe, however, that more flexible models are required to implement the dynamics of a multi-AVA system and *event driven asynchronous interaction architectures* are a promising method. Detailed discussions on this topic will be given in Section 4.1.

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

3.1 Realization of Wide View Cameras

To develop wide-area video surveillance 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[2], [3], [4], or

2) Active cameras mounted on computer controlled camera heads[5].

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 identification 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[7] 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[4] 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

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

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[5].

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

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* (in short, FV-PT camera)

Usually, zooming can be modeled by the shift of the projection center along the optical axis[6]. Thus to realize the *Fixed Viewpoint Pan-Tilt-Zoom Camera* (in short, FV-PTZ camera), 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

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 motion (Figure 7). 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 panoramic image.

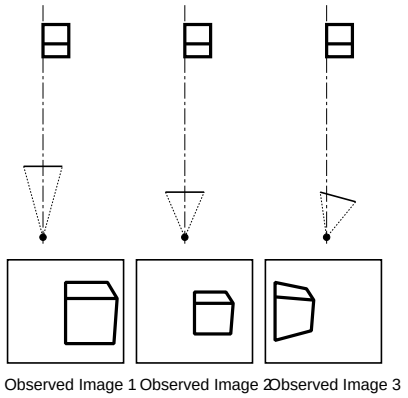


Figure 7: Images observed by an FV-PTZ camera.

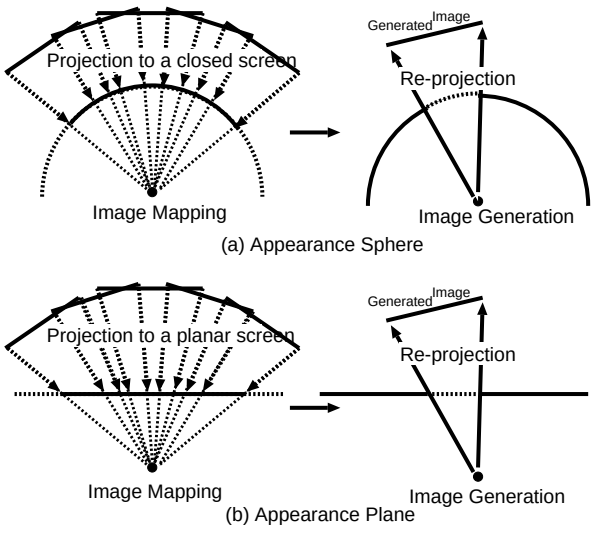


Figure 8: Appearance sphere and plane.

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 (Figure 8 (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 (Figure 8 (b)). The panoramic image on the planar screen is called *APpearance Plane* (APP in short).

As illustrated in the right side of Figure 8, 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 [8] $\sim$ [10] 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 surveillance. That is, in our case both the image acquisition and the projections on/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

Figure 9 shows the FV-PT camera head we developed, where the pan and tilt axes intersect at right and 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 [7]. The wide rotation angles (i.e. -180° $\leq$ pan $\leq$ 180° and 0 $\leq$ tilt $\leq$ 45°) enables the APS representation of a scene (Figure 10). Note that using this camera head, any (compact) video camera with any lens system can be calibrated to realize an APS camera.

Figure 9: Developed FV-PT camera head.

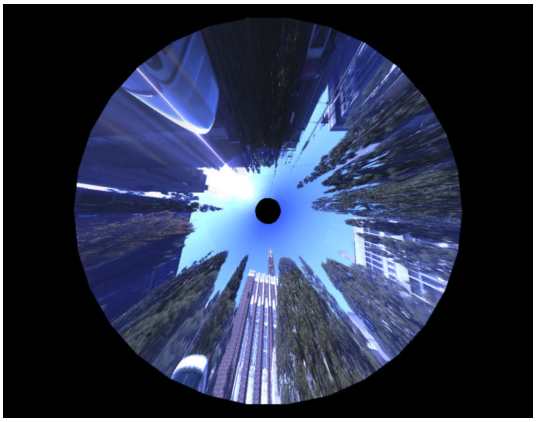


Figure 10: High resolution APS representation of Kyoto University Clock Tower scene.

Figure 11, 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 the following internal-camera-parameter calibration method for this camera, with which we can use the camera as an FV-PTZ camera.

1. Capture a set of partially overlapping images of a stationary scene by changing (pan, tilt) angles with a fixed zoom parameter (Figure 13).

2. Estimate such internal camera parameters that maximize the normalized correlation between those image areas that are mutu-

Figure 11: FV-PTZ camera.

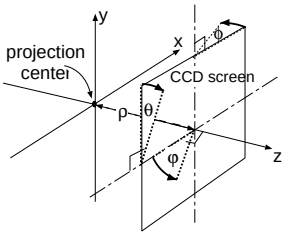


Figure 12: Slant angles of the CCD plane.

ally overlapping on APP.

The internal parameters employed include focal length (ρ), radial distortion parameter (κ), distortion center (x_0, y_0), and aspect ratio (α). To increase the calibration accuracy, we additionally introduce slant angles of the CCD plane (θ, ϕ, φ) (Figure 12). Figures 14 and 15 show APP images and gray level differences in the overlapping areas before and after the parameter optimization respectively. Note that this calibration method does not require any reference objects, and can be conducted automatically without any human support.

Changing the zoom parameter, we applied the above calibration to obtain

- $x_0, y_0, \alpha, \theta, \phi, \varphi$ are almost constant irrespectively of zooming.
- ρ and κ change almost linearly proportional to the zoom parameter.

These observations verify that we can model the camera as an FV-PTZ camera[5] .

3.5 Applications

Besides the wide panoramic scene visualization as illustrated in Figure 15, applications of the FV-PTZ camera include 1) moving object detection and tracking by background subtraction, which will be described in the next section, and 2) ego motion estimation and moving object detection and tracking based on optical flow. The latter uses such property of the FV-PTZ camera that irrespectively of the 3D scene geometry, 'homogeneous' optical flow fields[6] are generated by camera motions. In [11], we demonstrated practical utilities of our FV-PTZ camera in the optical flow based image analysis.

Figure 13: Observed images.

4 Dynamic Integration of Perception and Action for Real-Time Moving Object Detection and Tracking

This section 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. The control system is designed based on what

[5] The laser-beam-based calibration[7] showed that the projection center is about 1.1cm off the rotation center along the optical axis when the zooming factor is set smallest, and that as the zooming becomes large, the former comes closer to the latter. This displacement, however, does not cause any serious problems in the later applications; image distortions introduced by it stay less than 2 pixels when the observed scene is farther than 2.5m.

[6] Note that since our FV-PTZ camera employs a gimbal mechanism, flow patterns vary depending on tilt angles.

a)APP image

b)Gray level difference in overlapping areas

Figure 14: APP image generated with the initial parameters.

a)APP image

b)Gray level difference in overlapping areas

Figure 15: APP image generated with the optimized parameters.

we call the *event driven asynchronous interaction architecture*, which we believe is more flexible than ordinary control theory based methods; it can realize flexible temporal coordinations between visual perception and camera control modules. Note that in a CDV system, event driven asynchronous interactions play a crucial role to realize the dynamic integration of perception, action, and communication, since message exchanges among AVAs are asynchronous in its nature. A prototype system for object detection and tracking using the FV-PTZ camera was developed to test our idea. Experimental results demonstrated that the proposed dynamic control method greatly improves the performance of the object tracking.

4.1 Dynamic Vision

The integration of visual perception and camera action has been studied in Active Vision[12][13] and Visual Servo[14][15]. In the former, while many studies have been done on *Where to Look* problem, i.e. geometric camera motion planning based on image analysis, a little analysis has been done on system dynamics. Figure 16 illustrates the information flow between perception and action modules

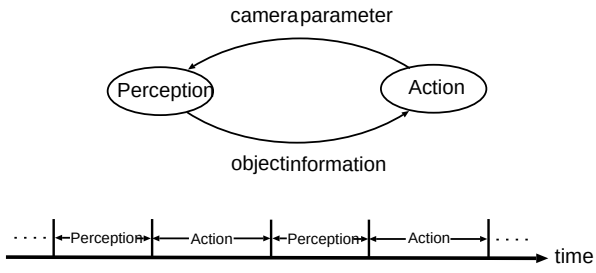


Figure 16: Information flow and dynamics in active vision.

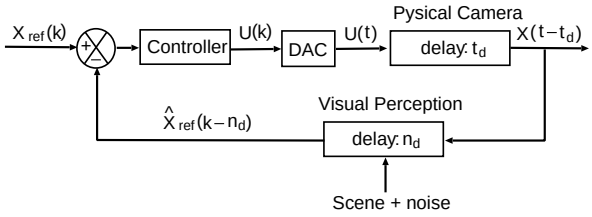


Figure 17: Position-based visual feedback system with delays.

and their dynamics. [14] called this dynamics the '*static* look and move structure,' where visual perception and camera control modules are activated sequentially.

In Visual Servo, on the other hand, various dynamic control methods have been studied based on the control theory [14][15]. For example, Figure 17 illustrates a position-based visual feedback system, where both control and perception delays (i.e. t_d and n_d) are taken into account. Brown[15] showed that the prediction-based control is effective to cope with delays.

In visual servo systems such as shown in Figure 17, visual perception and camera control modules work in parallel and the information flows continuously through the signal lines connecting the modules. Inter-module interactions, however, are rather simple and fixed. Firstly, the types of information exchanged between the modules is just the same as illustrated in the upper diagram in Figure 16. Secondly, the interactions are continuously synchronized by the analog and discrete time parameters (i.e. t and k in Figure 17) and no asynchronous in-

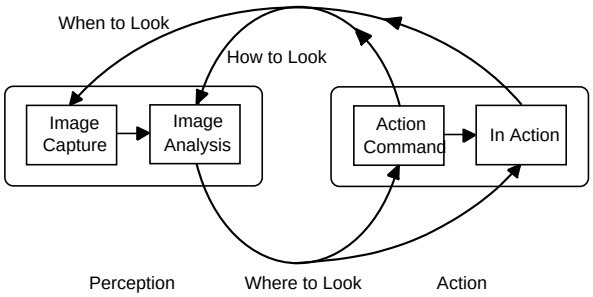


Figure 18: Information flows in dynamic vision.

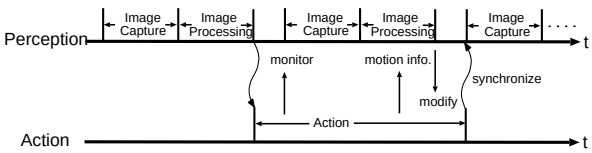


Figure 19: Event driven asynchronous interactions between perception and action in dynamic vision.

teraction mechanisms are incorporated, while asynchronous events usually happen in the real world. That is, the world itself has its own dynamics, which exhibits asynchronous features as its complexity increases; the world may include autonomous AVAs as illustrated in Figure 3. To make a system work adaptively in such complex scenes, we should develop more flexible dynamic interaction mechanisms between visual perception and camera control modules.

Based on the above discussions, we are proposing a novel scheme named *Dynamic Vision*, where the event driven asynchronous interaction between perception and action modules is realized. Distinguishing characteristics of dynamic vision are as follows.

- In a dynamic vision system, complicated information flows are formed between perception and action modules to solve *When to Look* and *How to Look* problems as well as ordinary *Where to Look* problem (Figure 18). For example, When to Look: Image acquisition timing should be determined depending on the camera motion, because quick motion can degrade observed images. How to Look: Camera parameters (focus, iris, zoom as well as motion parameters) can facilitate image analysis.

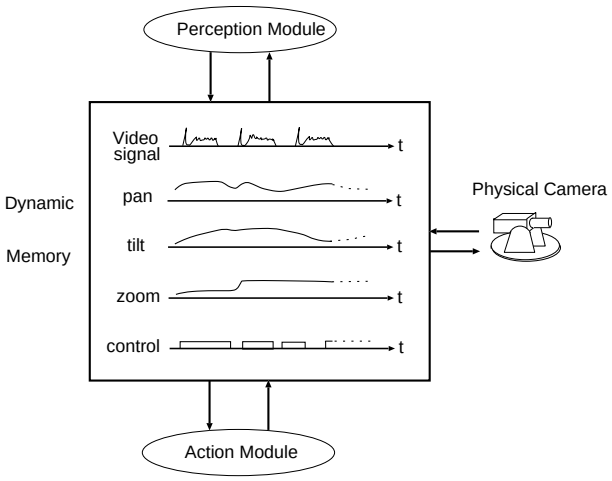


Figure 20: Dynamic memory architecture.

- The system dynamics is represented by a pair of parallel time axes, on which the dynamics of perception and action modules are represented respectively. Dynamic interactions between the modules are represented by inter-time-axes coordinations such as event-driven synchronizations and interruptions (Figure 19).

To implement a dynamic vision system, the *dynamic memory architecture* illustrated in Figure 20 can be used, where perception and action modules share what we call the *dynamic memory*. It records histories of control signals as well as state variables such as pan, tilt, and zoom. In addition, it stores their predicted values in the future (dotted lines in the figure). Perception and action modules read from and write into the memory depending their objectives and dynamics. Event driven asynchronous interactions between the modules can be realized by incorporating various temporal coordination mechanisms in concurrent/parallel processing: barrier synchronization, producer-consumer synchronization, semaphore, monitor and so on [16]. The dynamic memory architecture enables not only sophisticated prediction-based

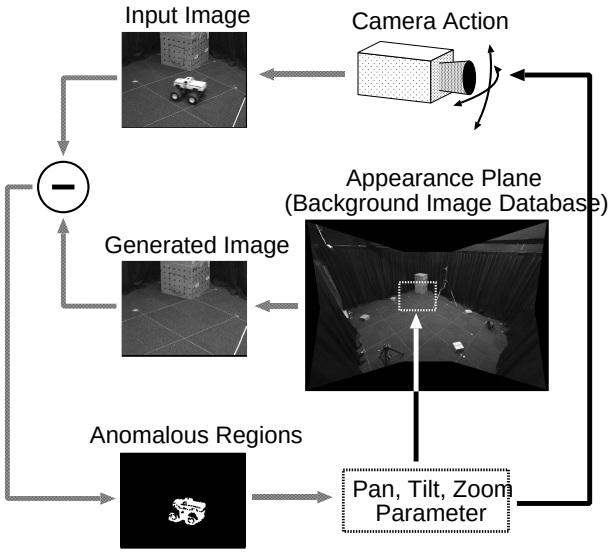


Figure 21: Basic scheme of the prototype system.

controls but also flexible dynamic interactions between perception and action modules.

4.2 Prototype System Development

4.2.1 Basic Scheme of Object Detection and Tracking

To embody the idea of Dynamic Vision, we developed a prototype system for real-time moving object detection and tracking by the FV-PTZ camera. Figure 21 illustrates its basic scheme:

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, [17], [18], and [19] employed probability distributions to model intensity variations at each pixel and used probabilistic anomaly computation methods for object detection. In [20], 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. Experimental results using real world scenes demonstrated its practical utilities. Since this method is time consuming, the prototype 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.

4.2.2 When to Look Problem

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[7] , 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

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

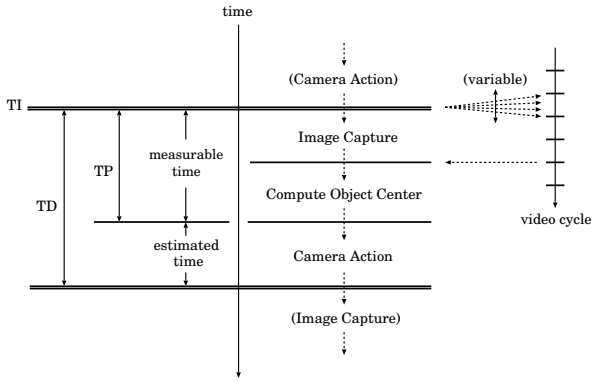


Figure 22: Time chart.

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 temporal coordination between perception and action modules.

One may claim that if the background subtraction were not employed, then such image acquisition timing control would not be required. In general, however, since computational resources of an AVA system are limited, the observation of meaningless images wastes the resources which could be used for other processing such as communication. Moreover, since the scene and other AVAs are interacting with the AVA based on their own dynamics, the image acquisition and processing should be done adaptively in accordance with such dynamics. Thus, the above mentioned *When to Look* problem has the generality.

4.2.3 Temporal Coordination between Perception and Action Modules

Figure 22 shows the time chart of the perception-action cycle. Suppose the image acquisition is initiated at t_0. The right vertical bar in Figure 22 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
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, the system first estimates the target object motion, which then is used to determines $\widehat{t}_d$ and the camera action.

4.2.3.1 Target Motion Estimation

Assuming the 2D motion vector (i.e. (Δpan, Δtilt)) of the target object is constant, its dynamics is represented by

$$P_{obj}(t_0 + t) = P_{obj}(t_0) + \frac{dP_{obj}(t_0)}{dt} \times t \quad (12)$$

$$T_{obj}(t_0 + t) = T_{obj}(t_0) + \frac{dT_{obj}(t_0)}{dt} \times t \quad (13)$$

where
$(P_{obj}(t_0), T_{obj}(t_0))$ and $(\frac{dP_{obj}(t_0)}{dt}, \frac{dT_{obj}(t_0)}{dt})$ denote the 2D position and 2D velocity of the object at t_0 respectively. In the prototype system, $(P_{obj}(t_0), T_{obj}(t_0))$ is defined by the centroid of a region detected by the background subtraction, and $(\frac{dP_{obj}(t_0)}{dt}, \frac{dT_{obj}(t_0)}{dt})$ by the centroid displacement in a pair of consecutive video frame images.

Then, to capture the target at $t_0 + \widehat{t}_d$, the current camera view direction $(P_{cam}(t_0), T_{cam}(t_0))$ should be changed by $(\Delta P_{cam}(t_0 + \widehat{t}_d), \Delta T_{cam}(t_0 + \widehat{t}_d))$:

$$\Delta P_{cam}(t_0 + \widehat{t}_d) = P_{obj}(t_0 + \widehat{t}_d) - P_{cam}(t_0) \quad (14)$$

$$\Delta T_{cam}(t_0 + \widehat{t}_d) = T_{obj}(t_0 + \widehat{t}_d) - T_{cam}(t_0) \quad (15)$$

4.2.3.2 Estimation of Camera Motion Dynamics

Although several uncertain factors are involved in the time spent by the image capturing and processing, their exact timing can be measured by the system clock (i.e. t_p in Figure

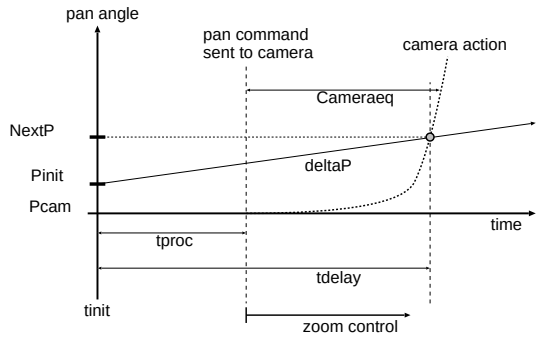


Figure 23: Estimation of the next view direction and image acquisition timing.

22). In order to estimate $\widehat{t}_d$, therefore, we have to estimate the dynamics of the camera action. Here we model the camera dynamics by $\mathcal{T}(\Delta P_{cam}, \Delta T_{cam})$, which denotes the time required to change pan and tilt angles by $(\Delta P_{cam}, \Delta T_{cam})$ and almost stop the motion. Using this model, $\widehat{t}_d$ can be represented by

$$\widehat{t}_d = \mathcal{T}(\Delta P_{cam}(t_0 + \widehat{t}_d), \Delta T_{cam}(t_0 + \widehat{t}_d)) + t_p \tag{16}$$

We conducted extensive experiments to model the dynamics of our FV-PTZ camera and obtained the following linear model:

$$t = \mathcal{T}(\Delta P_{cam}, \Delta T_{cam}) = 0.007745 \times max\{\Delta P_{cam}, \Delta T_{cam}\} + 0.2986, \tag{17}$$

where ΔP_{cam} and ΔT_{cam} are measured in $^\circ$ and t in second.

4.2.3.3 Determining Next View Direction and Timing

By solving equations (12)$\sim$(16), we can estimate 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. Suppose $\Delta T_{cam} < \Delta P_{cam}$. Figure 23 graphically illustrates the dynamics represented by equations (12)$\sim$(16). That is, $\widehat{t}_d$ is determined by the intersection point between the straight line representing the predicted target object trajectory and the bent line representing the camera dynamics.

4.2.3.4 Coping with Uncertainties by Dynamic Zoom Control

The task of the prototype system is to track the target object keeping its silhouette captured at the center of observed images. Many uncertain factors are involved in this task:

- Target Object Motion: Since the target object moves freely, its motion cannot be estimated precisely. Moreover, the system can measure only 2D object motion.
- Camera Motion: Whereas the camera dynamics is modeled a priori, its physical motion can vary depending on its internal mechanical and electronic states.
- Image Analysis: The computed position of the target object can fluctuate due to noise and varying photographing conditions.

The system controls the zoom to cope with these uncertainties. That is, when the degree of uncertainties is low, it zooms in to acquire high resolution object images. On the other hand, when some unexpected events happen and the prediction deviates largely from observed data, the system zooms out not to loose the target. In what follows, we describe this zoom control method.

All the above-mentioned uncertainties are reflected into the prediction error of the target position, i.e. the distance between the observed target centroid and the image center. The system records such prediction errors to learn the degree of uncertainties involved in the task. To evaluate the uncertainty degree, the prediction errors should be normalized; they depend on 1) observation interval: the error becomes larger if the observation interval gets longer, and 2) target silhouette size: since the error is measured on the image plane, it gets larger when the target itself is large and/or when large zooming factor is used.

We define the instantaneous uncertainty degree at the ith observation time t_i, $\Delta UD(t_i)$, as follows:

$$\Delta UD(t_i) = \frac{POS_{error}(t_i)}{T(t_i) \times \sqrt{AREA(t_i)}}, \tag{18}$$

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

$$\Delta UD_{max} = max\{\Delta UD(t_i)\}. \quad (19)$$

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.

$$POS^{max}_{error}(t_{i+1}) = \Delta UD_{max} \times (t_{i+1} - t_i)\sqrt{AREA(t_{i+1})} \quad (20)$$

$$AREA(t_{i+1}) = \frac{AREA(t_i)}{\alpha(t_i)} \times \alpha(t_{i+1}). \quad (21)$$

We got the following observations from the experiments to model the dynamics of the zoom control mechanism of our FV-PTZ camera:

- 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 (17), which represents the dynamics of the pan-tilt control, the following zoom control method is 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 Figure 23). That is, after computing $\alpha(t_{i+1})$, the system modifies the zooming factor only by such an amount that satisfies this temporal constraint.

4.3 Performance Evaluation

To demonstrate the effectiveness of the proposed dynamic coordination method between perception and action, we conducted experiments to detect and track a radio controlled toy car. The car is manually controlled by a human; it moves around the 4m × 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. Figure 24 shows a sequence of observed images and detected target silhouettes. Figures 25 and 26 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 Figure 26 denotes the horizontal view angle, which is inversely proportional to the zooming factor.

The entire tracking period is 13.77 seconds (i.e. about 2.1 image-observation/second in average). Figure 27 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 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 $(P_{obj}(t_0), T_{obj}(t_0))$ without taking into account the object 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 object 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.

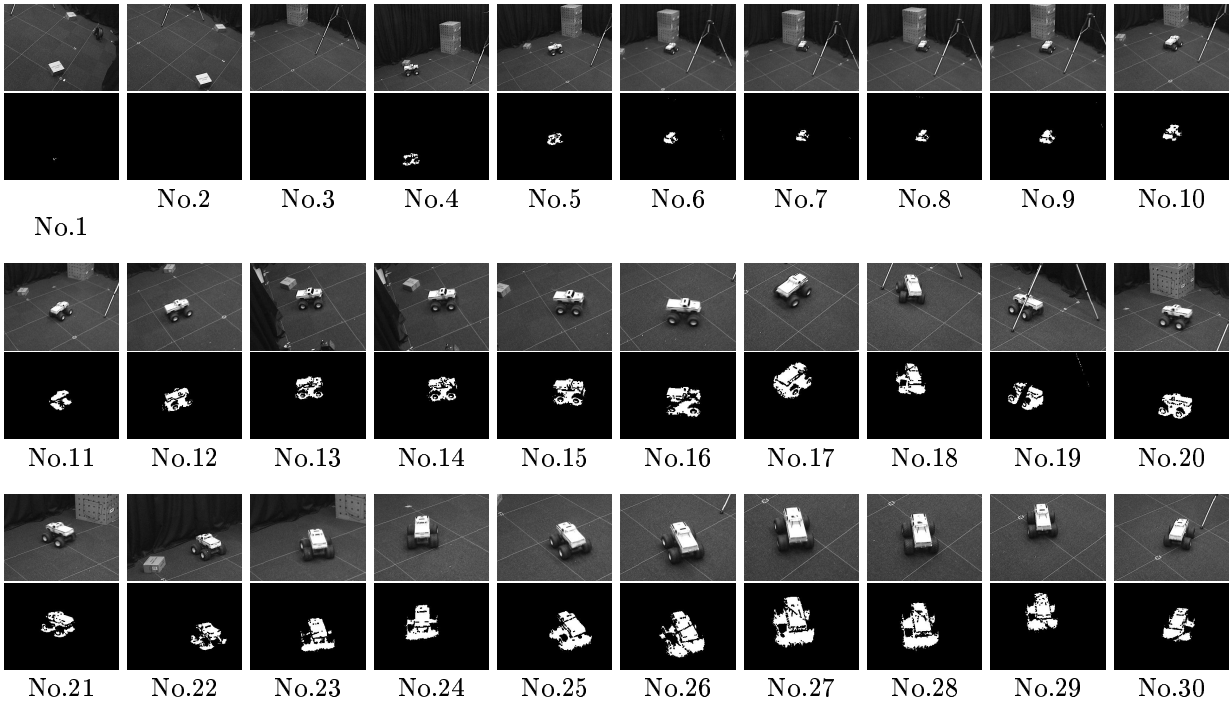


Figure 24: Images observed during tracking(Upper:input images, lower:detected object silhouette).

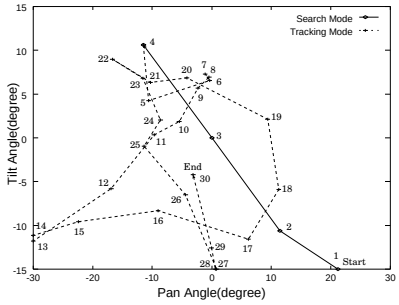


Figure 25: History of pan-tilt control.

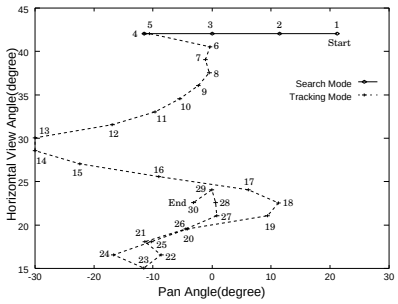


Figure 26: History of pan-zoom control.

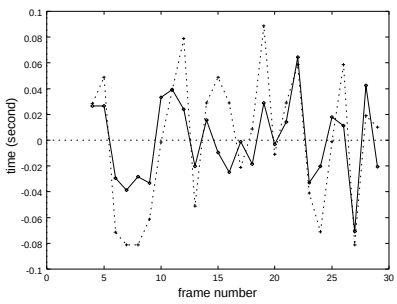


Figure 27: Dynamics of the system (see text).

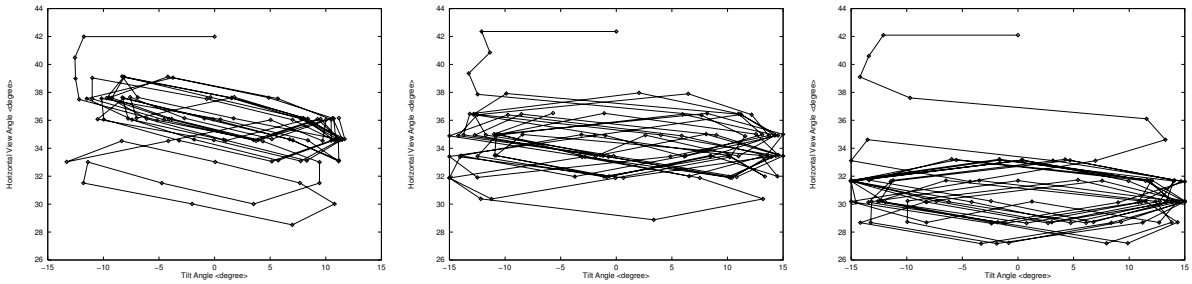
Note that all these three methods share the same zoom control method described before.

Figure 28 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 coordination method between perception and action modules is effective in moving object tracking.

5 Cooperative Object Tracking by Communicating Active Vision Agents

5.1 Introduction

This section addresses a multi-AVA system (i.e. a group of communicating AVAs) which cooperatively detects and tracks a focused target object. The task of the system is specified as follows: 1) Each AVA is equipped with the FV-PTZ camera and mutually connected



Method 1 Method 2 Method 3 (proposed method)

Figure 28: Performance evaluation: histories of tilt-zoom controls.

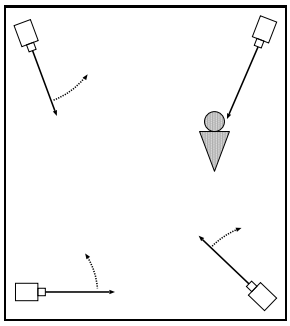


Figure 29: Gaze navigation

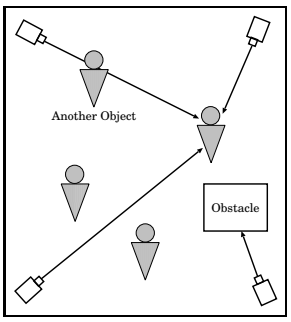


Figure 30: Cooperative gazing

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 (Figure 29). 4) All AVAs keep tracking the focused target cooperatively without being disturbed by obstacles or other moving objects (Figure 30). 5) When the target goes out of the scene, the system returns back to the initial search mode.

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.

In what follows, we design and implement a prototype system, where a major emphasis is put on how we can dynamically integrate visual perception, action, and communication. In the prototype system, the communication module in each AVA dynamically coordinates its perception and action modules to realize cooperative object tracking. Based on this scheme, cooperation protocols named *Agency Formation* and *Role Assignment* are proposed and realized by a state transition network. The network specifies *event driven asynchronous interactions* among the three modules as well as communication protocols among AVAs, through which behaviors of an AVA emerge. In this sense, this network representation can be considered as an augmentation of the dynamic memory proposed in Section 4.1. Some preliminary experimental results demonstrate the robustness and flexibility of the system.

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

As illustrated in Figure 2, there are two fundamental flows of information in a multi-AVA system: perception-action cycle and communication cycle. Since our knowledge and experience is limited, it is difficult to discuss general principles of integrating these information flows. So we took a task-oriented approach.

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 (Figure 29), 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 (Figure 30), 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.

Note that the above mentioned action and perception controls by the communication module are triggered asynchronously with the autonomous perception-action cycle in an AVA.

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

5.3 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.

5.3.1 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.

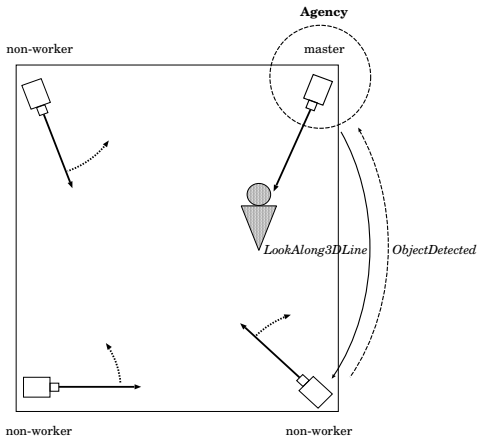


Figure 31: Agency formation.

Based on this consideration, we represent the target object by an *agency*, a group of those AVAs that are observing the target at the current moment. 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.

5.3.2 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.

5.3.2.1 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 (Figure 31). 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.

5.3.2.2 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 (Figure 31).

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.

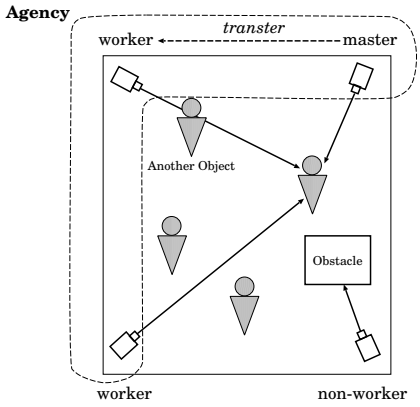


Figure 32: Role assignment.

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 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 in the next section.

5.3.2.3 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.

5.3.3 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 (Figure 32).

The transition from/to worker to/from 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 (Figure 32).

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.

5.4 Implementation by a State Transition Network

Figure 33 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 this sense, this network representation can be considered as an augmentation of the dynamic memory proposed in Section 4.1. This network, however, was designed based on the current task specification and we have not yet established any formal method to prove its dynamics. What we want to show here is the feasibility of cooperative object tracking by a multi-AVA system, which is an important step toward cooperative distributed vision.

In Figure 33, 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

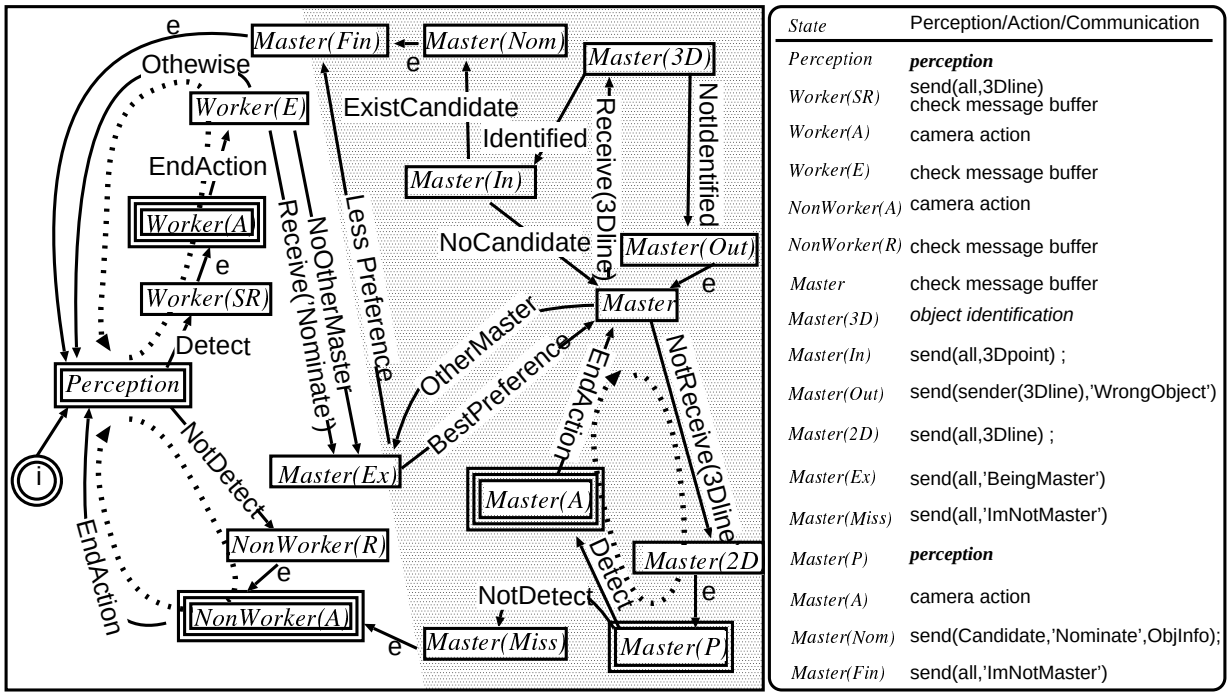


Figure 33: State transition network for the cooperative object tracking.

dynamics and dynamic behaviors of an AVA are fabricated by state transitions. The network specifies the following behaviors.

[Becoming the master]

1. Starting at state i, AVA_i first executes perception at *Perception*.

2. If an object is not detected, its state is changed to $Nonworker(A)$ via $NonWorker(R)$, where the message buffer is checked to see if any object information has been broadcast. At $Nonworker(A)$, the camera action is executed according to the object information received or the prespecified search method. Then it goes back to *Perception*.

3. If an object is detected, the state is changed to $Worker(A)$ via $Worker(SR)$, where the 3D view line toward the object is broadcast and the message buffer is checked to see if other AVAs have detected objects already. After executing the camera action at $Worker(A)$, the state is changed to $Worker(E)$, where the message buffer is checked.

   (a) If the existing master AVA nominates AVA_i as the successor or no master AVA exists, then AVA_i takes over the master authority or tries to generate the agency representing the detected object by changing its state to $Master(Ex)$. Then it waits for a prespecified period to check if other AVAs declare 'BeingMaster.' When time is out, it takes the master authority to move to *Master* or goes back to *Perception* via $Master(Fin)$ depending on its priority.

   (b) Otherwise, go back to *Perception*.

Note that a pair of dotted loops starting from *Perception* represent the perception $\rightarrow$ communication $\rightarrow$ action cycles performed by worker and nonworker AVAs.

[Working as the master]

At *Master*, AVA_i first checks the message buffer.

1. If other AVA(s) have declared 'BeingMaster', then go back to $Master(Ex)$ for the conflict resolution based on the priority.

2. If other AVAs have detected objects and

their 3D view line data have been received, then AVA_i executes the object identification procedure at $Master(3D)$.

(a) If the object is identified, AVA_i includes such AVAs into the agency that detected the identified object, and computes the 3D position of the object, which then is broadcast at $Master(In)$. Then, AVA_i tries to transfer the master authority to other member AVAs.

i. If the more reliable AVA exists, then AVA_i transfers the master authority at $Master(Nom)$ and goes back to *Perception* via $Master(Fin)$.

ii. Otherwise, AVA_i returns back to $Master$ again.

(b) Otherwise, AVA_i returns back to $Master$ via $Master(Out)$.

3. Otherwise, AVA_i moves to $Master(P)$ via $Master(2D)$, and executes perception to find the object by itself.

(a) If it is detected, AVA_i goes back to $Master$ via $Master(A)$, where the camera action is executed according to the perceived object data.

(b) Otherwise,
AVA_i moves to $NonWorker(A)$ via $Master(Miss)$.

Note that the dotted loop starting from $Master$ represent the communication → perception → action cycle performed by the master AVA.

As is obvious from the above description, 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.

5.5 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 × 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 to the target person. Then, both went out from the scene and after a while, a new person came into the scene.

Figure 34 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 (Figure 35).

Figure 36 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.

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.

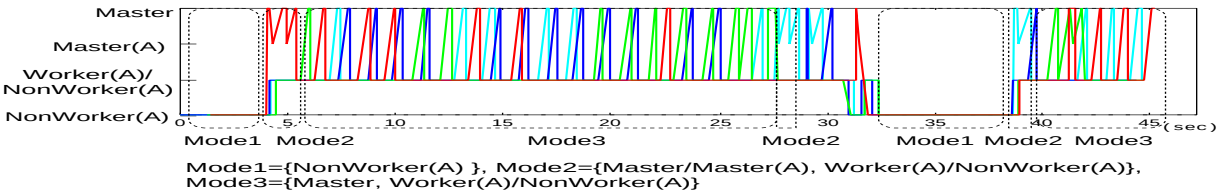


Figure 36: State transition histories of the four AVAs.

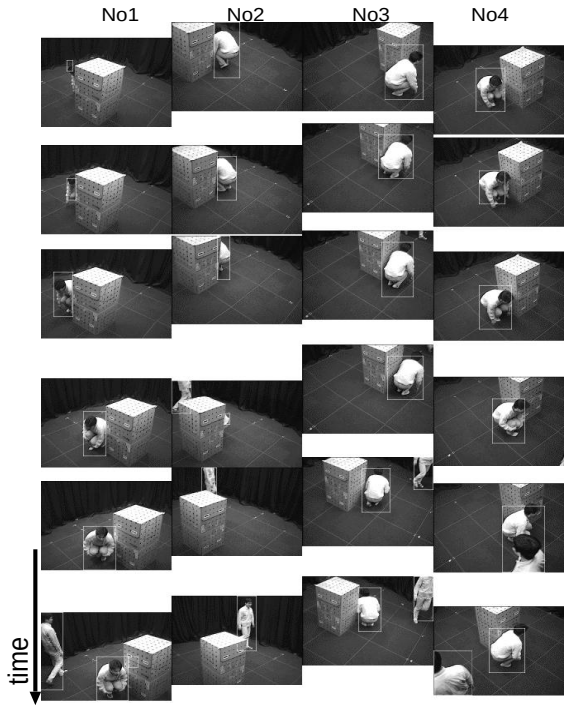


Figure 34: Partial image sequences observed by four cameras. The vertical length of an image represents 0.5 sec.

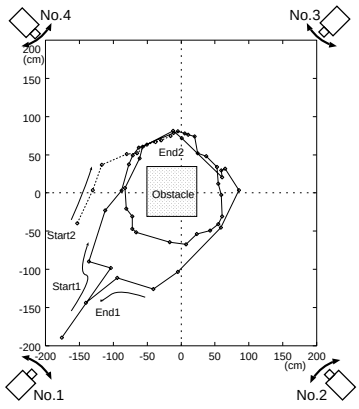


Figure 35: 3D target motion trajectories.

5.6 Discussions

In the prototype system, we introduced several timeout facilities to avoid deadlocks and realize real time object tracking. Moreover, the priority based conflict resolution was incorporated to guarantee the uniqueness of the master authority. The experimental results demonstrated that these functions worked well. We are now studying the following improvements:

- We should introduce a more systematic design method to specify the dynamics of communicating AVAs.
- Multiple AVAs should be 'synchronized' at 1) the state level for mutual cooperation (i.e. dynamic role assignment) and 2) the perception level for simultaneous object observation. To implement these synchronizations, we need to develop a wide spectrum of communication methods including high speed networks, real time communication methods, and cooperation protocols.
- The system should be augmented to be able to track multiple objects simultaneously. To realize this multi-target tracking, we have to augment the agency formation protocol and introduce a negotiation protocol between multiple agencies.

6 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 moving object detection and tracking by cooperative observation stations: 1) Fixed-Viewpoint Pan-Tilt-Zoom (FV-PTZ) camera for wide-area active imaging, 2) Real time object detection and tracking by an FV-PTZ camera, and 3) Cooperative object track-

ing by communicating active vision agents.

To improve the performance of these systems, we are now studying
1) Robust background subtraction method which can work in non-stationary indoor and outdoor scenes[20].
2) Implementation of the dynamic memory proposed in Section 4.1 using a SIMD real time video image processor, with which flexible and real time event driven asynchronous interactions between visual perception and camera action can be realized.
3) Representation and analysis methods of real time cooperation protocols among active vision agents.

In addition to the topics addressed in this paper, the project has done and is studying a wide spectrum of researches:
- Multi-focus camera for real time 3D range sensing[21]
- Multi-target motion analysis by cooperative agents[22]
- Human behavior recognition by multiple cameras[23]
- Scenario-based camera work design for intelligent TV studio[24]
- Remote lecturing systems by cooperative distributed vision[25]
- Visual behavior learning for cooperative soccer robots[26]

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] Poole, D., Mackworth, A., and Goebel, R.: Computational Intelligence, Oxford University Press, 1998.

[2] 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.

[3] Yamazawa K., Yagi Y. and Yachida M.: Obstacle Detection with Omnidirectional Image Sensor HyperOmni Vision, Proc. of ICRA, pp.1062 - 1067, 1995.

[4] Peri V. N. and Nayar S. K.: Generation of Perspective and Panoramic Video from Omnidirectional Video, Proc. of IUW, pp.243 - 245, 1997.

[5] Murray,D. and Basu,A.: Motion Tracking with an Active Camera, IEEE Trans. of PAMI, Vol. 16, No. 5, pp. 449-459, 1994.

[6] 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.

[7] Wada T. and Matsuyama T.: Appearance Sphere: Background Model for Pan-Tilt-Zoom Camera, Proc. of ICPR, Vol. A, pp. 718-722, 1996.

[8] 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.

[9] Greene N.: Environment Mapping and Other Applications of World Projections, CGA, 6 (11), pp. 21-29, 1986.

[10] Chen S.E.: QuickTime VR – An Image-Based Approach to Virtual Environment Navigation, Proc. of SIGGRAPH'95, pp. 29-38, 1995.

[11] Murase, K., Wada, T., and Matsuyama, T.: Moving Object Detection by a Rotating Camera, Proc. of MIRU'98, pp.I425-I430, 1998 (in Japanese).

[12] Aloimonos, Y. (ed.): Special Issue on Purposive, Qualitative, Active Vision, CVGIP: Image Understanding, Vol.56, No.1, 1992.

[13] Aloimonos, Y. (ed.): Active Perception, Lawrence Erlbaum Associates Publisher, 1993

[14] Weiss, L.E., Sanderson, A.C., and Neuman, C.P.: Dynamic Sensor-Based Control of Robots with Visual Feedback, IEEE Trans., Vol.RA-3, No.5, pp.404-417, 1987.

[15] Brown, C.M.: Gaze Control with Interactions and Delays, IEEE Trans., Vol.SMC-20, No.1, pp.518-527, 1990.

[16] Ben-Ari, M.: Principles of Concurrent Programming, Prentice-Hall, 1982.

[17] Nakai, H.: Robust Object Detection Using A-Posteriori Probability, Tech. Rep. of IPSJ, SIG-CV90-1, 1994 (in Japanese).

[18] Grimson, E.: A Forest of Sensors, Proc. of VSAM Workshop, 1997.

[19] Davis, L.: Visual Surveillance and Monitoring, Proc. of VSAM Workshop, 1997.

[20] 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).

[21] Hiura, S. and Matsuyama, T.: Depth Measurement by the Multi-Focus Camera, Proc. of CVPR, pp.953-959, 1998

[22] Yoshida, N., Sayo, T., and Yamamoto, K.: Multitarget Motion Analysis by Cooperative Multiagent System, Proc. of 2nd CDV Workshop, 1998.11.

[23] Wada, T. and Matsuyama, T.: Appearance Based Behavior Recognition by Event Driven Selective Attention, Proc. of CVPR, pp. 759-764, 1998

[24] Tokai, S. and Matsuyama, T.: Scenario Based Multi-Camera Work Planning, Proc. of 2nd CDV Workshop, 1998.11.

[25] Kameda, Y., Taoda, T., and Minoh, M. : High Speed 3D Reconstruction by Video Image Pipeline Processing and Division of Spatio-Temporal Space, IAPR Workshop on Machine Vision Applications, 1998.12

[26] Uchibe, E., Asada, M., and Hosoda, K.: Behavior Coordination for a Mobile Robot Using Modular Reinforcement Learning, Proc. of IROS'96, pp.1329-1336, 1996.

COPY

Image Understanding Workshop, Monterey, CA. 1998.11