Cooperative Distributed Vision:

Dynamic Integration of Visual Perception, Action, and Communication

Takashi Matsuyama

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

Abstract. We believe intelligence does not dwell solely in brain but emerges from active interactions with environments through perception, action, and communication. This paper give 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, navigation of (non-intelligent) mobile robots and disabled people. In this paper, we first define the framework of CDV and then present technical research results so far obtained: 1) fixed viewpoint pan-tilt-zoom camera for wide area active imaging, 2) active vision system for real-time moving object tracking and, 3) cooperative moving object tracking by communicating active vision agents.

1 Introduction

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 (Fig. 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.

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, navigation and guidance of (non-intelligent) mobile robots and disabled people, and cooperative mobile robots.

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.

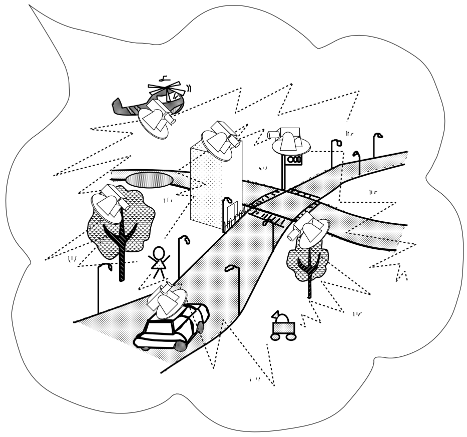


Fig. 1. Cooperative distributed vision.

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

2 Integrating Perception, Action, and Communication

2.1 Modeling Intelligence by Dynamic Interactions

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

$$Intelligence = Knowledge + Reasoning$$

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

In the 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 functional modules.

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

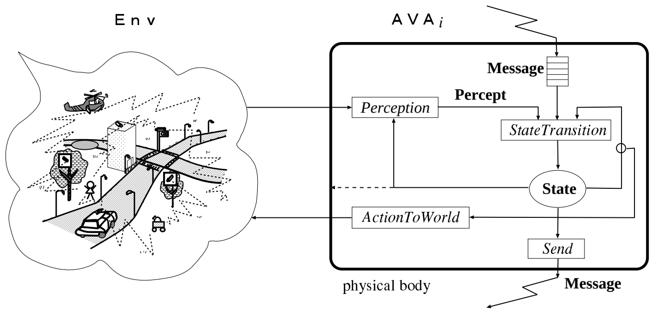


Fig. 2. Model of an Active Vision Agent.

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**	→	**Self**
Action	:	**Self**	→	**World**
Communication	:	**Self**	↔	**Others**

By integrating perception, action, and communication, various dynamic information flows are formed. In our model, reasoning implies the function which dynamically controls such flows of information. We believe intelligence does not dwell solely in brain but emerges from active interactions with environments through perception, action, and communication.

2.2 Model of Active Vision Agent

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. Then, fundamental functions of AVA$_i$ can be defined as follows (Fig. 2):

$$\begin{aligned} \mathit{Perception}_i &: \mathbf{Env} \times \mathbf{Sate}_i \mapsto \mathbf{Percept}_i && (1)\\ \mathit{Action}_i &: \mathbf{Sate}_i \mapsto \mathbf{Sate}_i \times \mathbf{Env}, && (2)\\ \mathit{Reasoning}_i &: \mathbf{Percept}_i \times \mathbf{Sate}_i \mapsto \mathbf{Sate}_i, && (3) \end{aligned}$$

where $\mathbf{Percept}_i$ stands for entities perceived by AVA$_i$.

The communication by AVA$_i$ can be defined by the following pair of message exchange functions:

$$\begin{aligned} \mathit{Send}_i &: \mathbf{Sate}_i \mapsto \mathbf{Message}_j && (4)\\ \mathit{Receive}_i &: \mathbf{Message}_i \times \mathbf{Sate}_i \mapsto \mathbf{Sate}_i, && (5) \end{aligned}$$

where $\mathbf{Message}_i$ and $\mathbf{Message}_j$ denote messages sent out to AVA$_i$ and AVA$_j$ via the communication network, respectively.

Based on the functional definitions given above, we can derive the following observations:
1. $Action_i$ in (2) can be decomposed into

$$InternalAction_i : \mathbf{Sate}_i \mapsto \mathbf{Sate}_i, \quad (6)$$

$$ActionToWorld_i : (\mathbf{Sate}_i \mapsto \mathbf{Sate}_i) \mapsto \mathbf{Env}. \quad (7)$$

The former implies pure internal state changes, such as pan-tilt-zoom controls of an active camera, while the latter means those state transitions whose side-effects incur state changes of **Env**, e.g. mechanical body-arm controls of a robot.
2. The internal state transition is caused by the (internal) action, the perception followed by the reasoning, and/or the message acception. Thus we can summarize these processes into

$$StateTransition_i : \mathbf{Percept}_i \times \mathbf{Message}_i \times \mathbf{Sate}_i \mapsto \mathbf{Sate}_i. \quad (8)$$

3. The above described model merely represents static functional dependencies and no dynamic properties are taken into account. A straightforward way to introduce dynamics into the model would be to incorporate time variable t into the formulae. For example, equation (8) can be augmented to

$$StateTransition_i(\mathbf{Percept}_i(t), \mathbf{Message}_i(t), \mathbf{Sate}_i(t)) = \mathbf{Sate}_i(t + \Delta t). \quad (9)$$

This type of formulation is widely used in control systems. In fact, Asada[2] used linearized state equations to model vision-based behaviors of a mobile robot. We believe, however, that more flexible models are required to implement the dynamics of an AVA;
1) Asynchronous Dynamics: Communications between AVAs are asynchronous in their nature.
2) Conditional Dynamics: Cooperations among AVAs require conditional reactions.

In what follows,
1. we first introduce *Fixed Viewpoint Pan-Tilt-Zoom (FV-PTZ) camera*, with which camera actions can be easily correlated with perceived images. (Section 3).
2. Then, a real-time object tracking system with an FV-PTZ camera is presented, where a novel dynamic interaction mechanism between perception and action is proposed (Section 4).
3. Finally, we present a cooperative object tracking system, where a state transition network is employed to realize asynchronous and conditional dynamics of an AVA, i.e. dynamic interactions among perception, action, and communication modules. (Section 5).

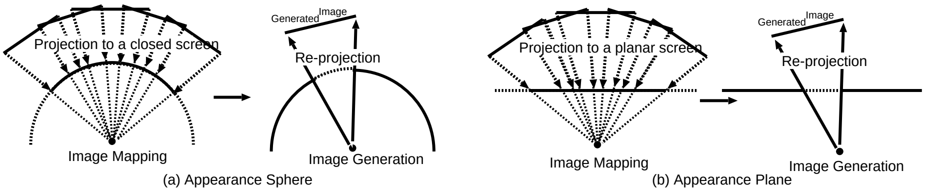


Fig. 3. Appearance sphere and plane.

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

The pan-tilt-zoom camera control can be modeled by $InternalAction_i$ in (6), which transforms AVA$_i$'s internal state from $\mathbf{State}_i(before)$ to $\mathbf{State}_i(after)$. This state change is reflected on $\mathbf{Percept}_i$ by $Perception_i$ in (1). As is well know in Computer Vision, however, it is very hard to find the correlation between $\mathbf{State}_i(before) \rightarrow \mathbf{State}_i(after)$ and $\mathbf{Percept}_i(before) \rightarrow \mathbf{Percept}_i(after)$; complicated 3D → 2D geometric and photometric projection processes are involved in $Perception_i$ even if **Env** is fixed.

We devised a sophisticated pan-tilt-zoom camera, with which camera actions can be easily correlated with perceived images.

3.1 Fixed Viewpoint Pan-Tilt-Zoom Camera

In general, a pan-tilt camera includes a pair of geometric singularities: 1) the projection center of the imaging system and 2) the rotation axes. In ordinary active camera systems, no deliberate design about these singularities is incorporated and 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 (unknown) 3D scene geometry.

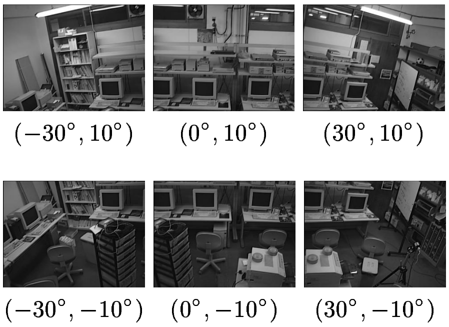
Our idea to overcome the above problem is simple but effective:

1. Make pan and tilt axes intersect with each other.
2. Place the projection center at the intersecting point.
3. Design such a zoom lens system whose projection center is fixed irrespectively of zooming.

We call the above designed active camera the *Fixed Viewpoint Pan-Tilt-Zoom camera* (in short, FV-PTZ camera)[3].

3.2 Image Representation for the 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. These variations are caused by the movement of the image plane, which can be rectified by projecting observed


$(-30°, 10°)$ $(0°, 10°)$ $(30°, 10°)$

$(-30°, -10°)$ $(0°, -10°)$ $(30°, -10°)$

Fig. 4. Observed images with (pan, tilt).

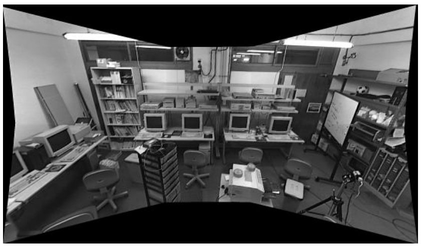


Fig. 5. Generated APP image.

images onto a common virtual screen. On the virtual screen, the projected images form a seamless wide panoramic image.

For the rectification, we can use arbitrarily shaped virtual screens:
APS: When we can observe the 360° panoramic view, a spherical screen can be used (Fig. 3 (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. 3 (b)). The panoramic image on the planar screen is called *APpearance Plane* (APP in short).

As illustrated in the right sides of Figs. 3(a)(b), 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.

We developed a sophisticated camera calibration method for an off-the-shelf active video camera, SONY EVI G20, which we found is a good approximation of an FV-PTZ camera ($-30° \leq \text{pan} \leq 30°$, $-15° \leq \text{tilt} \leq 15°$, and zoom: $15° \leq$ horizontal view angle $\leq 44°$) [4]. Figs. 4 and 5 show a group of observed images and the APP panoramic image synthesized.

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

As shown in (1), (2), and (3), $Perception_i$ and $Action_i$ are mutually dependent on each other and their integration has been studied in Active Vision and Visual Servo[5]. To implement an AVA system, moreover, we have to integrate three modules with different intrinsic dynamics:

Visual Perception : video rate periodic cycle
Action : mechanical motions involving variable (large) delays
Communication : asynchronous message exchanges, in which variable delays are incurred depending on network activities.

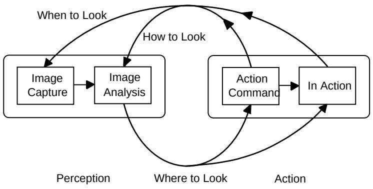


Fig. 6. Information flows in dynamic vision.

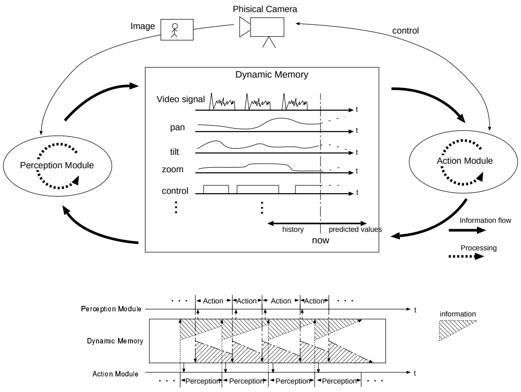


Fig. 7. Dynamic memory architecture.

4.1 Dynamic Memory

We are proposing a novel scheme named *Dynamic Vision*, whose distinguishing characteristics are as follows.
1. 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 (Fig. 6):

Where to Look : Geometric camera motion planning based on image analysis
When to Look : Image acquisition timing should be determined depending on the camera motion as well as the target object motion, because quick camera motion can degrade observed images (i.e. motion blur).
How to Look : Depending on camera parameters (e.g. motion speed, iris, and shutter speed), different algorithms and/or parameter values should be used to realize robust image processing, because the quality of observed images is heavily dependent on the camera parameters.

2. The system dynamics is represented by a pair of parallel time axes, on which the dynamics of perception and action modules are represented respectively (See the lower diagram in Fig. 7). That is, the modules run in parallel dynamically exchanging data.

To implement a dynamic vision system, the *dynamic memory architecture* illustrated in Fig. 7 has been proposed, where perception and action modules share what we call the *dynamic memory.* It records temporal histories of state variables such as pan-tilt angles of the camera and the target object location. In addition, it stores their predicted values in the future (dotted lines in the figure). Perception and action modules are implemented as parallel processes which dynamically read from and write into the memory according to their own intrinsic dynamics.

While the above architecture looks similar to the "whiteboard architecture" proposed in [6], the critical difference rests in that the dynamic memory main-

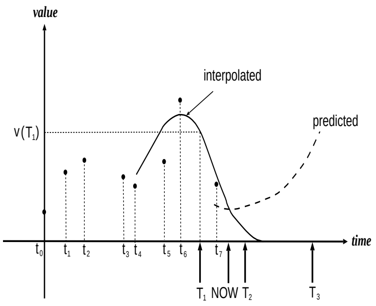


Fig. 8. Representation of a time varying variable.

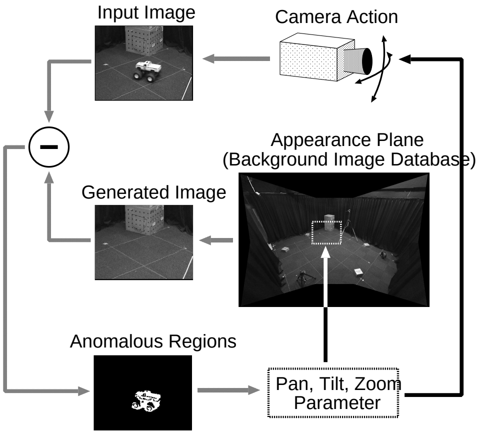


Fig. 9. Basic scheme for object tracking.

tains dynamically varying variables whose temporal periods are *continuously* spanning from the past to the future. That is, each entity in the dynamic memory is associated with the following temporal interpolation and prediction functions (Fig. 8):

1. *Temporal Interpolation*: Since the perception and action modules can write data only intermittently, the dynamic memory interpolates such discrete data. With this function, a module can read a value at any temporal moment, for example at T_1 in Fig. 8.
2. *Future Prediction*: Some module runs fast and may require data which are not written yet by another module (for example, the value at T_3 in Fig. 8). The dynamic memory generates predicted values based on those data so far recorded. Note that multiple values may be defined by the interpolation and prediction functions, for example, at NOW and T_2 in Fig. 8. We have to define the functions to avoid such multiple value generation.

With these two functions, each module can get any data along the time axis freely without waiting (i.e. wasting time) for synchronization with others. That is, the dynamic memory integrates parallel processes into a unified system while decoupling their dynamics; each module can run according to its own dynamics without being disturbed by the others. Moreover, prediction-based processing can be easily realized to cope with various delays involved in image processing and physical camera motion.

4.2 Prototype System Development

To embody the idea of Dynamic Vision, we developed a prototype system for real-time moving object detection and tracking with the FV-PTZ camera [7]. Fig. 9 illustrates its basic scheme:

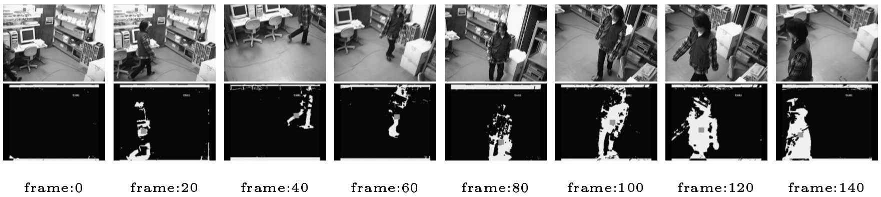


Fig. 10. Partial image sequence of tracking (Upper: captured images, Lower: detected target regions).

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.

Fig. 10 illustrates a partial sequence of human tracking by the system. Fig. 11 illustrates object and camera motion dynamics which were written into and read from the dynamic memory:

1. The action module reads pan-tilt angles (P',T') from the camera and writes them as CAM data into the dynamic memory.
2. When necessary, the perception module reads the CAM data from the dynamic memory: i.e. (Cp(t),Ct(t)) in the figure. Note that since the latter module runs faster, the frequency of reading operations of (Cp(t),Ct(t)) is much higher than that of writing operations of (P',T') by the former. (Compare two upper graphs.)
3. Then, the perception module conducts the object detection as illustrated in Fig. 9, whose output, i.e. the detected object centroid (Op(t),Ot(t)), is written back to the dynamic memory as OBJ data.
4. The action module, in turn, reads the OBJ data to control the camera.

Fig. 12 shows the read/write access timing to the dynamic memory by the perception and action modules. Note that both modules work asynchronously keeping their own intrinsic dynamics. Note also that the perception module runs almost twice faster than the action module (about 66msec/cycle).

These figures verify that the smooth real-time dynamic interactions between the perception and action modules are realized without introducing any interruption or idle time for synchronization.

5 Cooperative Object Tracking by Communicating Active Vision Agents

5.1 Task Specification

This section addresses a multi-AVA system which cooperatively detects and tracks a focused target object. The task of the system is as follows: 1) Each AVA

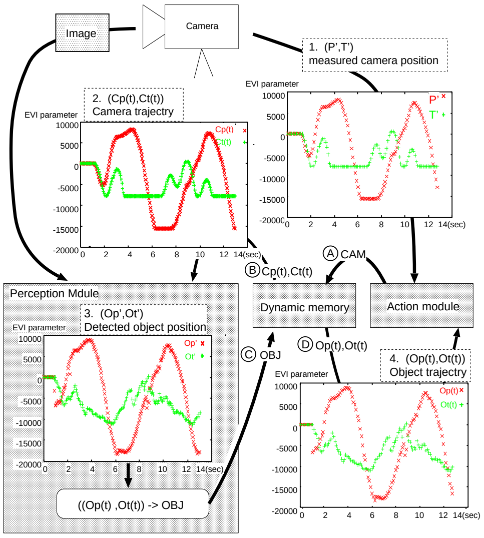


Fig. 11. Dynamic data exchanged between the perception and action modules (Large amplitude: pan, Small amplitude: tilt).

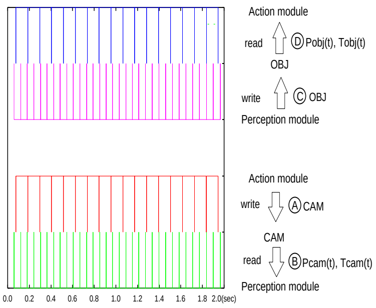


Fig. 12. Access timing to the dynamic memory by the perception and action modules (Upper: Object information, Lower: Camera information).

is equipped with the 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. 13). 4) All AVAs keep tracking the focused target cooperatively without being disturbed by obstacles or other moving objects (Fig. 14). 5) When the target goes out of the scene, the system returns back to the initial search mode.

All FV-PTZ cameras are calibrated and the object detection and tracking by each AVA is realized by the same method as described in Section 4.

5.2 Cooperative Object Tracking Protocol

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

Agency Formation Protocol 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. 13). 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.

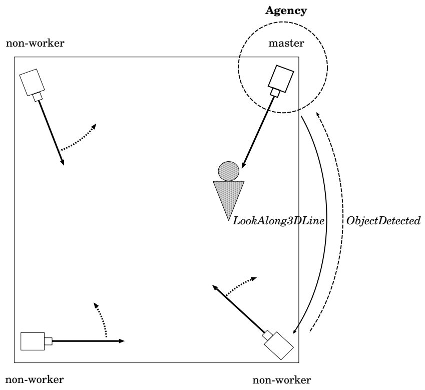


Fig. 13. Agency formation.

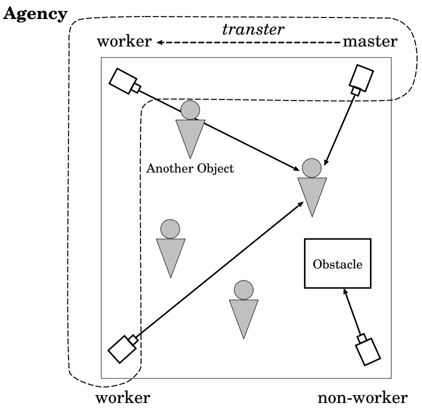


Fig. 14. Role assignment.

Suppose AVA$_i$ has generated an agency.

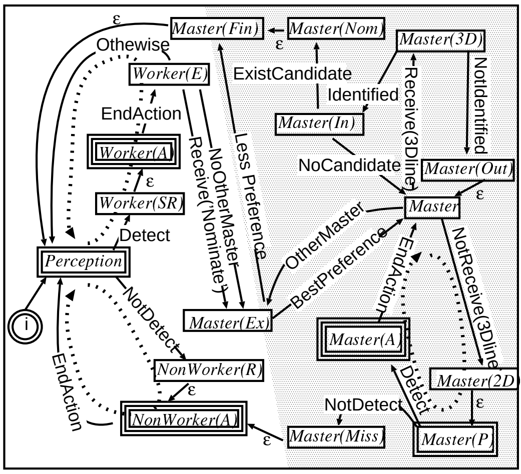
Gaze Navigation :First, AVA$_i$ broadcasts the 3D line, L_i, defined by the projection center of its camera and the object centroid in the observed image. Then, the other AVAs search for the object along this 3D line by controlling their cameras respectively (Fig. 13).

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 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.

Role Assignment Protocol 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 are called *worker* AVAs and AVAs outside the agency *non-worker* AVAs (Fig. 14).

When an AVA first generates the agency, it immediately becomes the master. The master AVA conducts the object identification described before to allow



State	Perception/Action/Communication
Perception	perception
Worker(SR)	send(all,3Dline) check message buffer
Worker(A)	camera action
Worker(E)	check message buffer
NonWorker(A)	camera action
NonWorker(R)	check message buffer
Master	check message buffer
Master(3D)	*object identification*
Master(In)	send(all,3Dpoint) ;
Master(Out)	send(sender(3Dline),'WrongObject')
Master(2D)	send(all,3Dline) ;
Master(Ex)	send(all,'BeingMaster')
Master(Miss)	send(all,'ImNotMaster')
Master(P)	perception
Master(A)	camera action
Master(Nom)	send(Candidate,'Nominate',ObjInfo);
Master(Fin)	send(all,'ImNotMaster')

Fig. 15. State transition network for the cooperative object tracking.

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. 14).

The prototype system employs a simple method: the master AVA transfers the authority to such member AVA whose object observation time is the latest, since the latest object information may be the most reliable.

5.3 Prototype System Development

Fig. 15 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.

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.

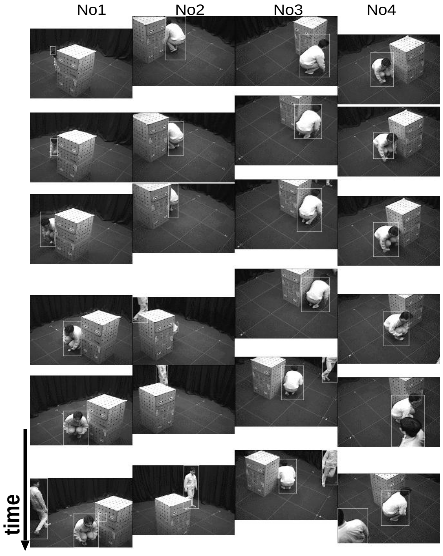


Fig. 16. Partial image sequences observed by four cameras. The vertical length of an image represents 0.5 sec.

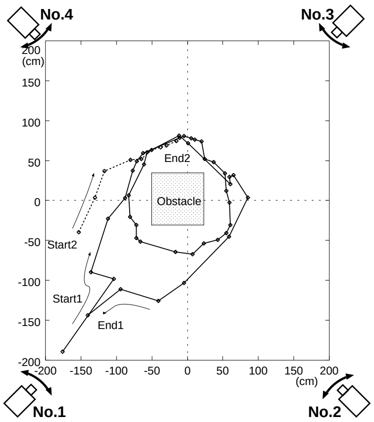


Fig. 17. 3D target motion trajectories.

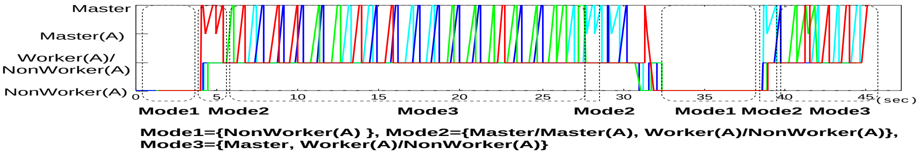


Fig. 18. State transition histories of the four AVAs.

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

We conducted experiments to verify the system performance. Two persons walked around a large box located at the center of the room (5m × 6m). Four FV-PTZ cameras (i.e. four AVAs) 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.

Fig. 16 illustrates partial image sequences observed by the 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 cameras are almost synchronized. This is because the master AVA broadcasts the 3D view line 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. 17).

Fig. 18 illustrates the state transition histories of the four AVAs. We can see that the system exhibits well coordinated behaviors as designed:

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.

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 active vision agents. Currently, we are studying

1. Robust object detection in complex natural scenes
2. Communication protocols for cooperative multi targets tracking
3. Application system developments such as remote lecturing and intelligent TV studio systems.

For detailed activities of the CDV project, see our homepage at
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).

References

1. Poole, D., Mackworth, A., and Goebel, R.: Computational Intelligence, Oxford University Press, 1998.
2. 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.
3. Wada, T. and Matsuyama, T.: Appearance Sphere: Background Model for Pan-Tilt-Zoom Camera, Proc. of ICPR, Vol. A, pp. 718-722, 1996.
4. Wada, T., Ukita, N., and Matsuyama, T.: Fixed Viewpoint Pan-Tilt-Zoom Camera and its Applications, Trans. IEICE, Vol.J81D-II, No.6, pp.1182-1193, 1998 (in Japanese).
5. Aloimonos, Y. (ed.): Active Perception, Lawrence Erlbaum Associates Publisher, 1993

6. Shafer, S.A., Stentz, A., and Thorpe, C.E.: "An Architecture for Sensor Fusion in a Mobile Robot," Proc. of IEEE Conf. on Robotics and Automation, pp.2002-2011, 1986.
7. Murase, K., Hiura, S., and Matsuyama, T.: Dynamic Integration of Perception and Action for Real-Time Object Detection and Tracking, IPSJ SIG-CVIM115-20, 1999 (in Japanese).

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