Real-Time Cooperative Multi-Target Tracking by Communicating Active Vision Agents

Norimichi Ukita†
Graduate School of Information Science
Nara Institute of Science and Technology, Japan
ukita@is.aist-nara.ac.jp

Takashi Matsuyama
Graduate School of Informatics
Kyoto University, Japan
tm@i.kyoto-u.ac.jp

Abstract − *We present a real-time cooperative multi-target tracking system. The system consists of a group of Active Vision Agents (AVAs), where an AVA is a logical model of a network-connected computer with an active camera. All AVAs cooperatively track their target objects by dynamically exchanging object information with each other. In this paper, we address the technologies employed in the system and demonstrate their effectiveness.*

1 Introduction

Object tracking is one of the most important and fundamental technologies for realizing real-world vision systems (e.g., visual surveillance systems[2], ITS (Intelligent Transport System)[3] and so on). To realize real-time flexible tracking in a wide-spread area, we employ the idea of *Cooperative Distributed Vision*[1]. A cooperative distributed vision system consists of a group of network-connected computers with active camera(s). A group of spatially distributed active cameras enable continuous wide-area observation as well as detailed measurement of 3D object information.

In this paper, we propose a real-time cooperative tracking system that gazes at multiple objects simultaneously. The system consists of communicating *Active Vision Agents* (AVAs, in short), where an AVA is a logical model of a network-connected computer with an active camera. For real-time object tracking by multiple AVAs, we have to solve many problems (e.g., how to design an active camera for dynamic object detection[1] and how to realize real-time object tracking with an active camera[5]). In this paper, we put our focus upon how to realize a real-time cooperation among AVAs.

To implement the real-time cooperation among AVAs, we propose a three-layered interaction architecture. In each layer, parallel processes exchange different kinds of information for effective cooperation. To realize a real-time information exchange and processing, we employ the dynamic memory architecture[5]. The dynamic interaction in each layer allows the total system to track multiple moving objects under complicated dynamic situations in the real world.

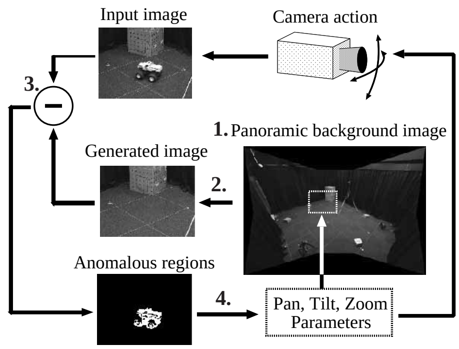


Figure 1: Object detection and tracking using an FV-PTZ camera.

2 Cooperative Multi-Target Tracking

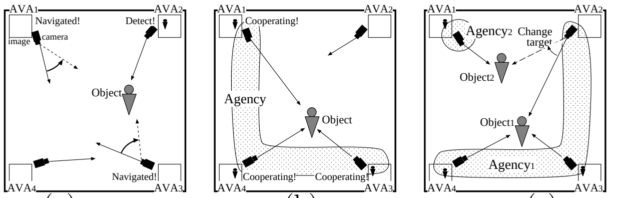
2.1 Architecture of AVA and Its Functions

Each AVA possesses a single *Fixed-Viewpoint Pan-Tilt-Zoom* (FV-PTZ) camera[1]: its projection center stays fixed irrespectively of any camera rotations and zoomings. An AVA can track the moving object as illustrated in Fig.1:

1. Generate a wide panoramic image of the scene; with the FV-PTZ camera, a wide panoramic image can be easily generated by mosaicing multiple images observed by changing pan, tilt and zoom parameters.
2. Extract a window image from the panoramic image according to the current pan-tilt-zoom parameters and regard it as the background image; the direct mapping exists between the position in the panoramic image and pan-tilt-zoom parameters of the camera.
3. Compute difference between the generated 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.

A camera is coupled to a network-connected computer. This network is not a special closed network (e.g., PC cluster) but an open network.

†'Information Infrastructure and Applications', PREST, JST.



(a) (b) (c)

Figure 2: (a) Gaze navigation, (b) Cooperative gazing, (c) Adaptive tracking.

In our system, an agent (i.e., AVA) corresponds to a single camera. All AVAs can, therefore, control their own cameras to gaze at the target. In [2, 3, 4], on the other hand, an agent is defined to correspond to the information of each object detected by the system and 2) all cameras are shared by agents, each of which manages the information of each detected object. This definition forces each agent to examine the object information detected by all cameras for tracking its target. In addition, multiple agents may control a camera inconsistently in tracking the target, if the system employs active cameras. As we can see, our definition of an agent has the advantage in that it has the one-to-one correspondence between an agent and a camera.

2.2 Basic Scheme for Cooperative Tracking

In our system, many AVAs are embedded in the real world, and observe a wide area. With these AVAs, we realize a multi-AVA system that cooperatively tracks multiple targets. Following are the tasks of the system:

1. Initially, each AVA independently searches for an object that comes into the observation scene.
2. If an AVA detects a target, it navigates the gazes of other AVAs towards the target (Fig.2 (a)).
3. AVAs, which gaze at the same object, track the focused target cooperatively (Fig.2 (b)). A group of these AVAs is called an *Agency*.
4. Depending on the target motion, each AVA dynamically changes its target (Fig.2 (c)).
5. When the target gets out of the scene, each AVA decides whether it searches for an object again or joins another agency depending on situations.

To realize the above cooperative tracking, we have to solve the following problems:

Multi-target identification: To gaze at each target, the system has to distinguish multiple objects.

Real-time and reactive processing: To cope with the dynamics in the scene (e.g., object motion), the system has to execute the process in real time and deal with variations in the scene reactively.

Adaptive resource allocation: We have to implement a two phased dynamic resource (i.e., AVA) allocation: (1) To perform both object search and tracking simultaneously, the system has to preserve AVAs that search for new targets even while tracking targets, (2) For each target to be tracked by the AVA that is suitable for gazing at, the system has to adaptively assign AVAs to their targets.

We solve these problems with real-time cooperative communication among AVAs and agencies.

3 Task Specification

The tracking system needs to search for an object in the scene. This role is called *Search*. Once the target is detected, the system gazes at it to obtain its information. This role is called *Tracking*. In addition, the system is also required to selectively gaze at the object whose information is necessary for the given task. We specify the task of the system by the following three parameters, namely a *Task-Constraint*, an *Object-Priority* and a *Utility-Function*.

3.1 Task-Constraint

An AVA that searches for an object is called a *Freelancer*-AVA. An AVA belonging to an agency for tracking the target is called a *Member*-AVA. We realize various capabilities of the system, in terms of the combination of search and tracking as follows.

Def. 1 (Search-level, Tracking-level) *The search-level (the horizontal axis) and the tracking-level (the vertical axis) indicate the rate of AVAs that perform search and tracking, respectively.*

$$0 \leq \quad \text{Search-level } (= N_F/N_A) \quad \leq 1$$
$$0 \leq \quad \text{Tracking-level } (= N_M/N_A) \quad \leq 1$$

where N_F, N_M and N_A denote the numbers of freelancer-AVAs, member-AVAs and all AVAs, respectively.

We define the task-constraint and the current state of the system on the system state graph.

Def. 2 (Current state (S_P, T_P)) *This parameter represents the search-level (S_P) and the tracking-level (T_P) at the present time. ($S_P + T_P$) is always 1.*

Def. 3 (Task-constraint (S_C, T_C)) *This parameter represents the minimum search-level (S_C) and tracking-level (T_C), where $0 \leq (S_C + T_C) \leq 1$. The system has to keep S_C and T_C while working. This parameter is determined by a user depending on the task given to the system.*

Each AVA dynamically changes its own role between search and tracking to adapt the current state of the system to the task-constraint.

3.2 Object-Priority

The object-priority is given to each object's category that can be distinguished by the system.

Def. 4 (Object-priority P_P) *Let P_P denote the object-priority of the target of agency$_P$. The range of the object-priority is $0 \leq P_P \leq 1$.*

The number of the member-AVAs in agency$_P$ (denoted by M_P) is determined by the object-priority of the target: $M_P =$ (The total number of AVAs) $\times$ (P_P/S). Provided that S is the total sum of the object-priority $P_{1,\cdots,A}$, where A is the total number of the agencies.

3.3 Utility-Function

Each AVA can freely change its role under the restrictions given by the task-constraint and object-priority. A guideline for the adaptive role assignment is represented by what we call the utility-function. Each AVA decides its role to increase the value of the utility-function while keeping the task-constraint and object-priority. The utility-function is the sum of the following search-value and tracking-value.

- **Search-value** is determined by a fitness of each freelancer-AVA for search.
- **Tracking-value** is determined by a fitness of each member-AVA for tracking its target.

This utility-function can be designed to be adapt itself to the task given by a user [1].

4 Three-layered Dynamic Interaction for Cooperative Tracking

In our system, parallel processes cooperatively work by dynamically interacting with each other. As a result, the system as a whole works as a tracking system. By composing the system as a group of multiple processes, we can represent various complex behaviors of the total system through the interaction between processes. Designing the total system can be, therefore, reduced to designing each process. Furthermore, the states and those transitions of the system increase enormously by combining with each other. We believe that this property allows the system to cope with complicated situations in the real world.

For the system to engage in object tracking, object identification is significant. We, therefore, classify the system into three layers depending on the types of object information employed for identification. In what follows, we address the interaction in each layer.

4.1 Intra-AVA layer

In the bottom layer in Fig.3, perception, action and communication modules that compose an AVA exchange time-series information with each other via the dynamic memory[5][2] possessed by each AVA. The interaction among three modules materializes the functions of the AVA.

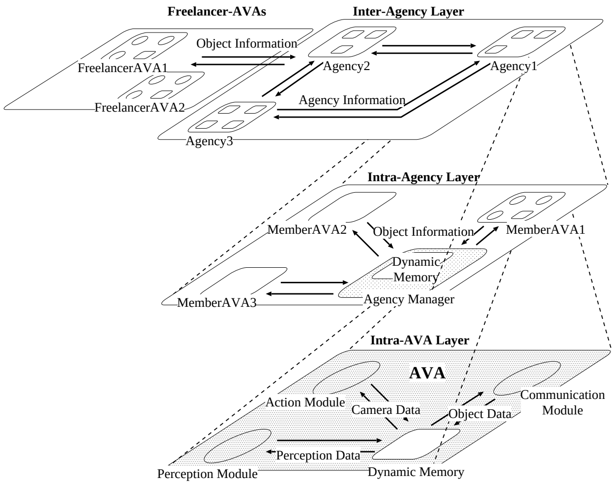


Figure 3: Three layers in the system.

(1) Perception: This module continues to capture images and detect objects in the observed image. Let the 3D view line L be determined by the projection center of the camera and the object region centroid in the observed image. When the module detects N objects at $t+1$, it computes and records into the dynamic memory the 3D view lines toward the objects (i.e., $L^1(t+1), \cdots, L^N(t+1)$). Then, the module compares them with the 3D view line toward its currently tracking target at $t+1$, $\widehat{L}(t+1)$. Note that $\widehat{L}(t+1)$ can be read from the dynamic memory whatever temporal moment $t+1$ specifies. Suppose $L^x(t+1)$ is closest to $\widehat{L}(t+1)$, where $x \in \{1, \cdots, N\}$. Then, the module regards $L^x(t+1)$ as denoting the newest target view line and records it into the dynamic memory.

(2) Action: When an active camera is ready to accept a control command, the action module reads the 3D view line toward the target (i.e., $\widehat{L}(now)$) from the dynamic memory and controls the camera to gaze at the target. As will be described later, when an agency with multiple AVAs tracks the target, it measures the 3D position of the target (i.e., $\widehat{P}(t)$) and sends it to all member AVAs, which then is written into the dynamic memory by the communication module. If such information is available, the action module controls the camera based on $\widehat{P}(now)$ in stead of $\widehat{L}(now)$.

(3) Communication: Data exchanged by the communication module over the network can be classified into two types: detected object data (e.g., $\widehat{L}(t)$ and $\widehat{P}(t)$) and messages for various communication protocols which will be described later.

4.2 Intra-Agency layer

As defined before, a group of AVAs which track the same target form an agency. The agency formation means the generation of an *Agency Manager*, which is an independent parallel process to (1) coordinate interactions among its member-AVAs and (2) commu-

[1] We give an example in Sec.5.1.

[2] With the dynamic memory, all modules can exchange their information asynchronously at any time.

nicate with other agencies and freelancer-AVAs. In our system, an agency should correspond one-to-one to a target. To make this correspondence dynamically established and persistently maintained, the following two kinds of object identification are required in the intra-agency layer (the middle layer in Fig.3).

(1) Spatial object identification

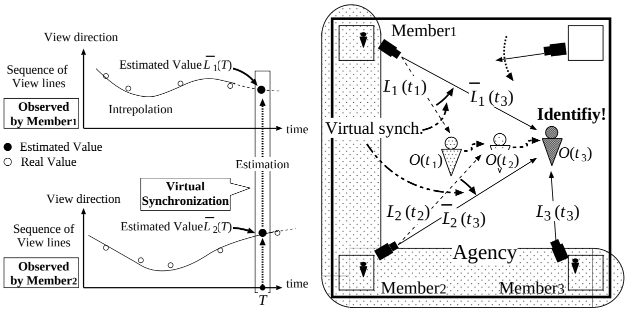
The agency manager has to establish object identification between the groups of the 3D view lines detected and transmitted by its member-AVAs. The agency manager checks distances between those 3D view lines detected by different member-AVAs and computes the 3D target position from a set of nearly intersecting 3D view lines. The manager employs what we call the *Virtual Synchronization* to virtually adjust observation timings of the 3D view lines (see 4.2.1 for details). Note that the manager may find none or multiple sets of such nearly intersecting 3D view lines. To cope with these situations, the manager conducts the following temporal object identification.

(2) Temporal object identification

The manager records the 3D trajectory of its target, with which the 3D object position(s) computed by spatial object identification is compared. That is, when multiple 3D locations are obtained by spatial object identification, the manager selects the one closest to the target trajectory. When spatial object identification failed and no 3D object location was obtained, on the other hand, the manager selects such 3D view line that is closest to the target trajectory. Then the manager projects the target 3D position onto the selected view line to estimate the new 3D target position. Note that when an agency contains only a single AVA, neither spatial nor temporal object identifications succeed and hence the member-AVA just conducts appearance-based 2D tracking by itself.

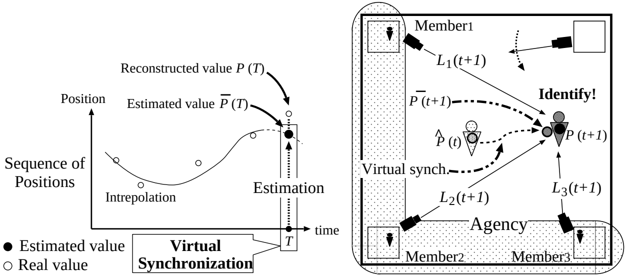
4.2.1 Virtual Synchronization for Spatial Identification

Since AVAs capture images autonomously, member-AVAs in an agency observe a target at different moments. Furthermore, the message transmission over the network introduces unpredictable delay between the observation timing by a member-AVA and the object identification timing by the agency manager. Other distributed systems which consist of autonomous cameras coped with this problem as follows: In [2], the newest information gathered from each camera is considered to be observed at the same time. In [4, 6], each object information includes its time stamp. The system regards the information observed at t_i and t_j, where $|t_i - t_j|$ is small enough, as simultaneous information. Such approximate methods, however, break down under complicated situations and network congestion. To solve this problem, we introduce the dynamic memory into an agency manager, which enables the manager to virtually synchronize any asynchronously observed/transmitted data. We call this function *Virtual Synchronization.*



(a) (b)

Figure 4: Virtual synchronization for spatial object identification: (a) Read values from the dynamic memory, (b) Spatial identification.



(a) (b)

Figure 5: Virtual synchronization for temporal object identification: (a) Read values from the dynamic memory, (b) Temporal identification.

Fig.4 shows the mechanism of the virtual synchronization. All 3D view lines computed by each member-AVA are transmitted to the agency manager, which then records them into its internal dynamic memory. Fig.4 (a), for example, shows a pair of temporal sequences of 3D view line data (indicated by white points in the figure) transmitted from member-AVA$_1$ and member-AVA$_2$, respectively. When the manager wants to establish spatial object identification at T, it can read the pair of the synchronized 3D view line data at T from the dynamic memory (i.e., $\bar{L}_1(T)$ and $\bar{L}_2(T)$ in Fig.4 (a), both of which are indicated by black points in the figure). In the actual example shown in Fig.4 (b), the manager reads $\bar{L}_1(t_3)$ and $\bar{L}_2(t_3)$ from its dynamic memory to adjust observation timings of three member-AVAs.

4.2.2 Virtual Synchronization for Temporal Identification

For temporal object identification, an agency manager has to compare the 3D position of its target at t with the 3D positions of the detected objects$_{1,\ldots,N}$ at $t+1$. The result of object identification is, however,

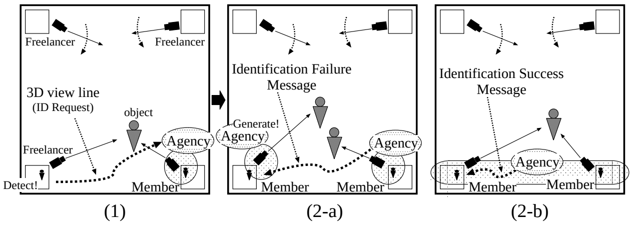


Figure 6: Agency formation protocol.

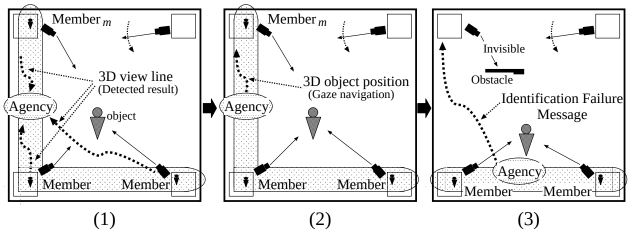


Figure 7: Agency maintenance protocol.

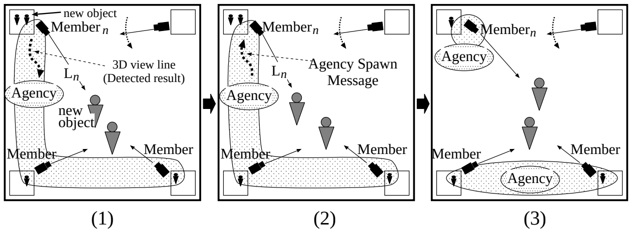


Figure 8: Agency spawning protocol.

unreliable because the object information obtained at different moments is compared with each other.

This problem can be also solved with the dynamic memory. Let $\widehat{P}(t)$ denote the 3D target trajectory recorded in the dynamic memory and $\{P_i(T)|i = 1,\cdots,M\}$ the 3D positions of the objects identified at T. Then the manager (1) reads $\widehat{P}(T)$ (i.e., the estimated target position at T) from the dynamic memory, (2) selects the one among $\{P_i(T)|i = 1,\cdots,M\}$ closest to $\widehat{P}(T)$, and (3) records it into the dynamic memory as the target position.

Figure 5 shows an example of temporal object identification with the virtual synchronization. The 3D position $P(t+1)$ is reconstructed at $t+1$. The agency manager then estimates the 3D position of the target at $t+1$ (i.e., $\bar{P}(t+1)$), and compares $\bar{P}(t+1)$ with $P(t+1)$.

The following three communication protocols are activated depending on the success or failure of the above mentioned temporal object identification.

4.2.3 Agency Formation Protocol

An *Agency Formation* protocol defines (1) the new agency generation by a freelancer-AVA and (2) the participation of a freelancer-AVA in an existing agency.

When a freelancer-AVA finds a new object, it requests from the existing agencies object identification between the newly detected object and the target of each agency (Fig.6, (1)). Depending on whether or not the result of this identification is successful, the freelancer-AVA works as follows:

- **When no agency established a successful identification,** the freelancer-AVA that finds the new object starts a new agency manager and joins this agency (Fig.6, (2-a)).
- **When an agency established a successful identification,** the freelancer-AVA joins the agency that has made successful identification, if the task-constraint can be kept (Fig.6, (2-b)).

4.2.4 Agency Maintenance Protocol

An *Agency Maintenance* protocol defines (1) the cooperative tracking, (2) the continuous maintenance of an agency and (3) the elimination of an agency.

After an agency is generated, the agency manager continues spatial and temporal object identifications for cooperative tracking (Fig.7, (1)). If temporal object identification between the target of the agency and the object detected by member-AVA$_m$ fails, the agency manager reports the 3D position of the target to member-AVA$_m$. This information navigates the gaze of member-AVA$_m$ towards the target (Fig.7, (2)). Nevertheless, if the failure of identification continues for a long time, the agency manager puts member-AVA$_m$ out of the agency (Fig.7, (3)).

If all member-AVAs are unable to observe the target, the agency manager forces them to be freelancer-AVAs and eliminates itself. To maintain the acquired object information, the agency manager records its object information into the database before eliminating itself. This information will be read by the newly generated agency if their targets are the same object.

4.2.5 Agency Spawning Protocol

An *Agency Spawning* protocol defines the new agency generation from an existing agency.

After spatial and temporal object identifications, the agency manager may find such a 3D view line(s) that does not correspond to the target. Let L_n denote such 3D view line detected by member-AVA$_n$ (Fig.8, (1)). The agency manager requires other agencies to compare L_n with their targets for object identification. If none of the identification is successful (namely, there is not an agency that tracks the newly detected object), the agency manager orders member-AVA$_n$ to generate a new agency (Fig.8, (2)). Member-AVA$_n$ then joins a new agency (Fig.8, (3)).

4.3 Inter-Agency layer

The fundamental task of an agency is to keep tracking its own target. In order to keep tracking the target in the complicated wide area, agencies need to adaptively exchange their member-AVAs with each other. To realize the adaptive reconstruction of the agency, the information about targets and member-AVAs are exchanged between agencies (the top layer in Fig.3).

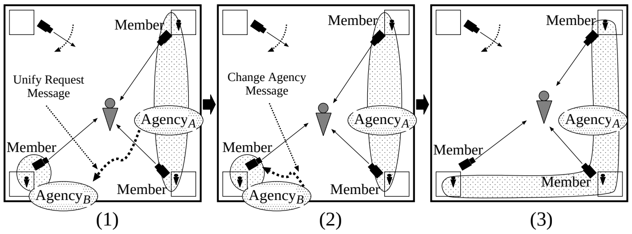


Figure 9: Agency unification protocol.

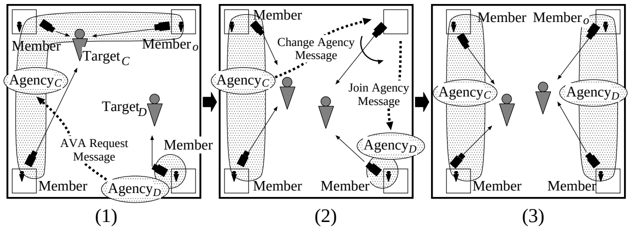


Figure 10: Agency restructuring protocol.

An agency that has received this information from another agency (agency$_i$) compares the 3D position of its own target with that of agency$_i$'s target. This object identification is not reliable if these 3D positions are observed at different moments. This problem can be solved with the virtual synchronization; with the 3D positions of its target recorded as time-series data in the dynamic memory, the agency manager can synchronize the 3D position of its target with the received 3D position of another object.

Depending on the result of object identification between agencies, the following two protocols are activated.

4.3.1 Agency Unification Protocol

An *Agency Unification* protocol defines the merging procedure of the agencies, both of which happen to track the same object. This protocol is achieved when the result of object identification between the agencies is successful. Figure 9 shows an example.

Followings are actual examples of situations that cause the agency unification.

- When an agency considers multiple objects in the scene as a single object because of the identification failure.
- When a single object is regarded as multiple objects because of the identification failure, and then multiple agencies are formed for the same object.

That is, this protocol is required to cope with the failures of object identification and discrimination.

4.3.2 Agency Restructuring Protocol

An *Agency Restructuring* protocol defines the dynamic interchange of member-AVAs between agencies. This protocol is achieved when the result of object identification between the agencies is unsuccessful. The agency manager performs the agency restructuring taking into account the following two factors:

- The number of the member-AVAs is determined by the object-priority of the target.
- Under the restriction about the number, each agency is attended by AVAs, which are suitable for gazing at the target, based on the utility-function.

We have various factors in determining the aptitude of each AVA for tracking, namely the criterion for the agency restructuring. A user can settle down this criterion depending on the task given to the system.

In an example illustrated in Fig.10, agency$_D$ requests a member-AVA from agency$_C$, and then agency$_C$ transfers member-AVA$_o$ to agency$_D$.

4.3.3 Communication with Freelancer-AVAs

An agency manager communicates with freelancer-AVAs as well as with other managers (the top layer in Fig.3). As described in the agency formation protocol in Sec.4.2.3, a freelancer-AVA activates the communication with agency managers when it detects an object. To determine whether or not generate a new agency based on the agency formation protocol, a freelancer-AVA communicates with agency managers when it detects an object. An agency manager, on the other hand, sends to freelancer-AVAs its target position when the new data are obtained. Then, each freelancer-AVA decides whether it continues to be a freelancer-AVA or joins into the agency depending on the task specification.

4.4 Soundness of the System

In the proposed system, all events happened in the real world are characterized by the result of object identification. Therefore, by verifying the types of the protocols executed depending on the result of each object identification, we can confirm the necessity and sufficiency of the protocols for multi-target tracking.

All the protocols are activated by an agency, and object identification is established when the agency received the object information from freelancer-AVAs, member-AVAs and other agencies. Table 1 shows the types of the protocols that are activated according to the relations between the type of the received object information and the result of object identification. As we can see, the protocols are designed just enough in accordance with the situations in the real world.

5 Experiments

We experimented to verify the effectiveness of the proposed system. We employed ten AVAs. Each AVA was implemented on a network-connected PC (PentiumIII 600MHz × 2) with an active camera (SONY EVI-G20), where the perception, action, and communication modules as well as agency managers were realized as UNIX processes. The internal clocks of all the PCs were synchronized by Network Time Protocol to realize the virtual synchronization. With this architecture, the perception module can capture images and detect objects in the observed image at about 0.1[sec] intervals on average. Figure 13 (a) shows the camera layout The external camera parameters were calibrated.

Table 1: Protocols activated depending on the result of object identification.

Received object information	Identification success	Identification failure
3D view lines of objects from a freelancer-AVA	Agency Formation	Agency Formation
3D view line of a target from a member-AVA	Agency Maintenance	Agency Maintenance/Spawning
3D view lines of non-targets from a member-AVA	Agency Maintenance	Agency Spawning
3D point of a target from an agency	Agency Unification	Agency Restructuring

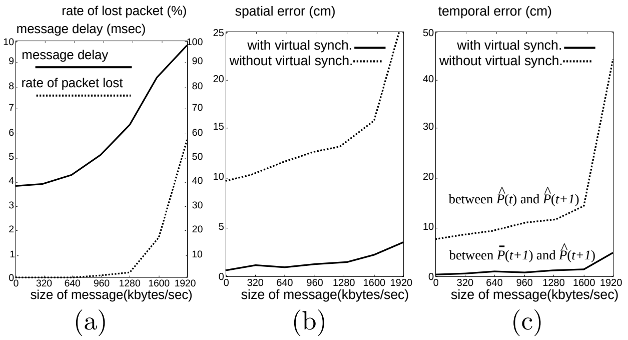


Figure 11: (a) Delay of the message (solid line) and Rate of lost packets (dotted line), (b) Error in spatial identification, (c) Error in temporal identification.

5.1 Designing Utility-Function

We designed the utility-function as follows:

Search-value of freelancer-AVA$_f$: Let W_f denote the area size of the floor that is visible from AVA$_f$. The search-value of AVA$_f$ (denoted by V_{S_f}) is determined as follows: $V_{S_f} = \alpha_S \times W_f$, where α_s is a constant that is determined so that V_{S_f} is well-balanced with the following tracking-value.

Tracking-value of member-AVA$_m$: Let D^n_m denote the 3D distance between the camera of AVA$_m$ and the target of agency$_n$, and A^n_m denote the angle between the central direction of AVA$_m$'s view angle and the direction from the camera to the target object. The tracking-value of AVA$_m$ (denoted by $V_{T^n_m}$) is determined as follows: $V_{T^n_m} = (1)/(D^n_m) \times (1)(A^n_m)$.

5.2 Performance Evaluation

We conducted experiments using the systems with/without the virtual synchronization. To verify the effectiveness of the virtual synchronization against not only the asynchronized observations but also the network congestion, we broadcasted vain packets over the network to adjust the network load.

The system tracked two computer-controlled mobile robots. Both the robots repeated a straight-line motion at a speed of 50[cm/sec] in the observation scene.

Figure 11 (a) shows the network conditions when the size of the vain messages is changed. The error of spatial identification in Fig.11 (b) denotes the average distance between the reconstructed 3D position and the 3D view lines detected by member-AVAs. The error of temporal identification in Fig.11 (c) denotes the average distance between the 3D positions of the same target, each of which are reconstructed/estimated at different times (i.e., $\widehat{P}(t)/\bar{P}(t+1)$ and $\widehat{P}(t+1)$).

As we can see, the virtual synchronization helps both spatial and temporal object identifications, especially in the case of bad network conditions.

5.3 Verifying Communication Protocols

In the next experiment, the system tracked two people. Object$_1$ first came into the scene. Next, object$_2$ came into the scene. Both objects then moved freely.

Followings are the given task specification.

Task-constraint Search-level=0.1. Tracking-level=0.9.

Object-priority The values of all objects were 1.0.

The upper part of Fig.12 shows the partial image sequences observed by AVA$_2$, AVA$_5$ and AVA$_9$. The images on the same column were taken at almost the same time. The regions enclosed by black and gray lines in the images show the detected regions of object$_1$ and object$_2$, respectively. Each figure in the bottom of Fig.12 shows the role of each AVA and the agency organization at such a moment when the upper images in the same column were observed. White circles denote freelancer AVAs, while black and gray circles indicate member AVAs belonging to agency$_1$ and agency$_2$, respectively. Black and gray squares indicate computed locations of object$_1$ and object$_2$ respectively.

The system worked as follows.

a: Each AVA searched for an object independently.

b: AVA$_5$ first detected object$_1$, and agency$_1$ was formed.

c: All AVAs except for AVA$_5$ were tracking object$_1$ as the member-AVAs of agency$_1$, while AVA$_5$ was searching for a new object as a freelancer-AVA.

d: AVA$_5$ detected object$_2$ and generated agency$_2$.

e: The agency restructuring balanced the numbers of member-AVAs in agency$_1$ and agency$_2$.

f: Since no AVA could distinguish two objects, the agency unification merged agency$_2$ into agency$_1$.

g: When the targets got apart, agency$_1$ detected a 'new' target. Then, it activated the agency spawning protocol to generate agency$_2$ again for target$_2$.

h: Object$_1$ was going out of the scene.

i: After agency$_1$ was eliminated, all the AVAs except for AVA$_4$ tracked object$_2$.

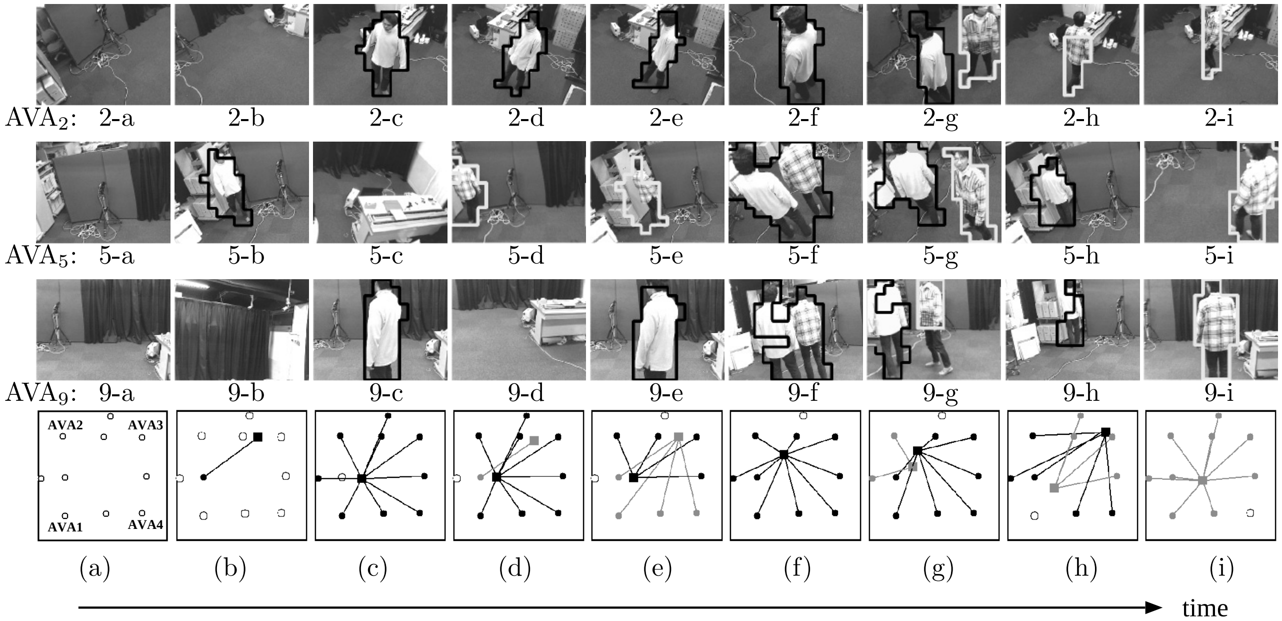


Figure 12: Upper: Partial image sequences, Lower: The role of each AVA and the agency organization

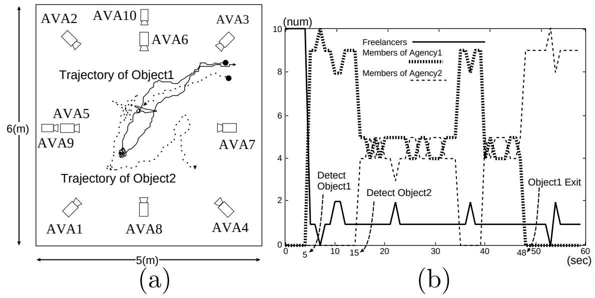


Figure 13: (a) Trajectories of the targets, (b) The number of AVAs that performed each role.

Figure 13 (a) shows the trajectories of the targets reconstructed by the agencies. Figure (b) shows the dynamic population changes of freelancer AVAs, AVAs tracking target$_1$ and those tracking target$_2$.

As we can see, the dynamic cooperations among AVAs and agencies worked well and enabled the system to keep tracking multiple targets.

6 Concluding Remarks

This paper proposed a real-time cooperative multi-target tracking system with multiple active cameras. The system has the following properties:

- Parallel processes dynamically interact with each other, which results in the system that works as a whole for cooperative tracking.
- The system is classified into three layers to establish various types of object identification.

  Intra-AVA: Perception, action and communication modules work together as a single AVA by dynamically interacting with each other.

  Intra-Agency: AVAs in an agency exchange object information to track the target.

  Inter-Agency: To adaptively reform agencies taking into account targets' motions, agencies mutually exchange their information.
- Employing the dynamic memory realized the dynamic interactions in each layer without synchronization.

These properties allow the system to be adaptable to complicated dynamic situations in the real world.

This work was supported by PREST program of JST and the Grant-in-Aid for Scientific Research (No.13224051).

References

[1] T. Matsuyama, "Cooperative Distributed Vision - Dynamic Integration of Visual Perception, Action and Communication -", *Proc. of Image Understanding Workshop*, pp.365–384, 1998.

[2] A. Nakazawa, H. Kato, S. Hiura and S. Inokuchi, "Tracking multiple people using distributed vision systems", *Proc. of ICRA*, pp.2974–2981, 2002.

[3] S. Nishio and Y. Ohta, "Tracking of Vehicles at an Intersection by Integration of Multiple Image Sensors", *Proc. of MVA*, pp.321–324, 1992.

[4] B. Horling, *et al*, "Distributed Sensor Network for Real Time Tracking", *Proc. of the 5th ICAA*, pp. 417–424, 2001.

[5] T. Matsuyama, *et al*, "Dynamic Memory: Architecture for Real Time Integration of Visual Perception, Camera Action, and Network Communication", *Proc. of CVPR*, pp.728–735, 2000.

[6] G. P. Stein, "Tracking from Multiple View Points: Self-calibration of Space and Time", *Proc. of CVPR*, Vol. I, pp.521–527, 1999.