# Virtual Synchronization for Real-Time Multi-Target Tracking by Asynchronous Distributed Cameras

Norimichi Ukita† and Takashi Matsuyama‡
†Graduate School of Information Science, Nara Institute Science and Technology
‡Graduate School of Informatics, Kyoto University
ukita@is.aist-nara.ac.jp and tm@i.kyoto-u.ac.jp

## 1. Introduction

Object tracking is one of the most important and fundamental technologies for various real-world vision systems. To realize real-time flexible tracking in a wide-spread area, we employ a group of network-connected computers with an active camera, whose logical model is called an *Active Vision Agent* (AVA).

A three-layered interaction architecture with AVAs for tracking was proposed in [2]. Autonomous reactive behaviors of AVAs without synchronization allow the total system to cope with dynamic situations in the scene and achieve complex tasks. For the system to integrate information observed by different AVAs, however, all the information should be synchronized. To solve these two contrary problems, we propose the virtual synchronization mechanism employing the dynamic memory architecture proposed in [3]. In addition, by employing multiple pan-tilt-zoom cameras and exchanging time-series data among them in real time, we realize active-camera controls for wide-area observation and acquiring high-resolution images of observed moving objects.

## 2. Three-layered Dynamic Interaction for Cooperative Tracking by Active Vision Agents

In this section, we summarize the basic scheme of the system proposed in [2].

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. The FV-PTZ camera allows us to generate a wide panoramic image of the scene by mosaicing multiple images observed by changing pan, tilt and zoom parameters. We can extract background images taken with arbitrary combinations of the pan-tilt-zoom parameters from the panoramic image. An AVA can, therefore, detect an anomalous region by the background subtraction during widely observing the scene. Thus the tracking by an AVA is achievable by changing the gazing direction to the detected region in the image.

In our system, many AVAs are embedded in a wide area. Following are the tasks of the system:

1. If an AVA detects a target, it navigates the gazes of other AVAs towards the target (Fig.1 (a)).

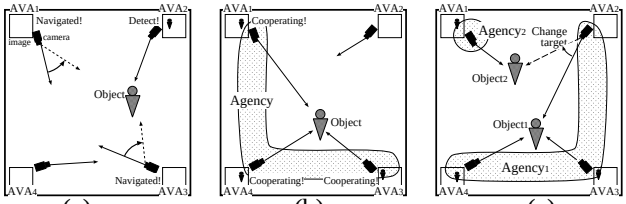



**Figure 1. Basic scheme for cooperative tracking: (a) Gaze navigation, (b) Cooperative gazing, (c) Adaptive tracking.**

2. AVAs that track the same object form a group called an agency and track the focused target cooperatively (Fig.1 (b)).

3. Depending on the target motion, each AVA dynamically changes its target (Fig.1 (c)).

We realize the above tasks with real-time cooperative communication among AVAs and agencies.

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.

### 2.1. Intra-AVA layer

In the bottom layer (intra-AVA layer), perception, action and communication modules exchange time-series information with each other via the dynamic memory[3] [1] possessed by each AVA. The interaction among three modules materializes the functions of an AVA.

**(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

[1] With the dynamic memory, all modules can exchange their information asynchronously at any time. The details are discussed in Sec.4.

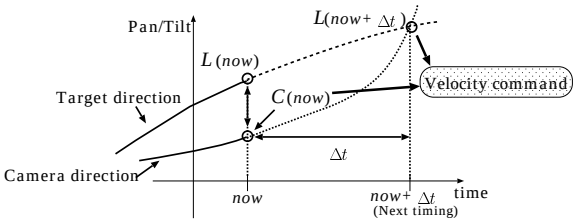



**Figure 2. Prediction-based camera control.**

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:** We employ the prediction-based control method as well as information exchange without synchronization. When an active camera is ready to accept a control command, the action module reads the 3D view line towards the target ($\widehat{L}(now)$) and the view line of the camera (denoted by $C(now)$) from the dynamic memory. The module then determines the next pan-tilt velocity command so that $\widehat{L}(now + \Delta t)$ coincides with $C(now + \Delta t)$ at the next command timing ($now + \Delta t$) as shown in Fig.2. $\Delta t$ should be determined by intensive experiments.

Note that the action module can control the camera only according to its intrinsic dynamics and perform smooth camera motions. These properties are superior to the sequential camera control proposed in [1], where a single process continues stop-and-sensing observations.

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.

## 2.2. 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 coordinate interactions among its member-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 the system).

**(a) 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. 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 identification.

**(b) 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 identifications succeed and hence the member-AVA just conducts appearance-based 2D tracking by itself.

Depending on the success or failure of the above mentioned temporal object identification, AVAs form an agency, maintenance its organization and disperse according to three kinds of communication protocols defined in [2].

## 2.3. Inter-Agency layer

The fundamental task of an agency is to keep tracking its own target. In order to keep tracking the target in a 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 the system). 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.

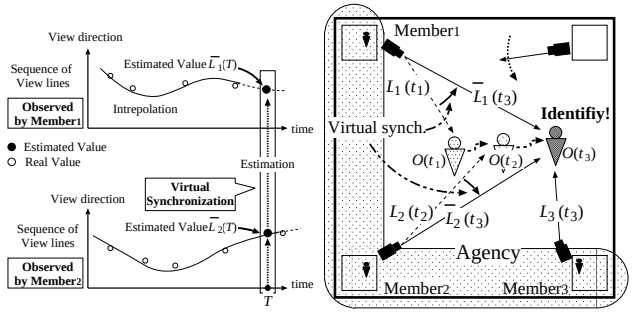
Depending on the result of object identification between agencies, they change their organizations according to two kinds of communication protocols defined in [2].

# 3. Virtual Synchronization for Identification

As described in Sec.2, object information observed at different moments is compared with each other for identification. Furthermore, the message transmission over the network introduces unpredictable delay between the observation timing by an AVA and the object identification timing by the agency manager. The result of object identification is, therefore, unreliable because asynchronized object information is compared with each other. To solve this problem, we propose the following dynamic interaction methods.

**(a) Spatial object identification**

In many multi-camera systems (see [4], for example), the object information detected at $t_i$ and $t_j$, where $|t_i - t_j|$




(a) (b)

**Figure 3. Virtual synchronization for spatial object identification: (a) Read values from the dynamic memory, (b) Spatial identification.**

is small enough, is considered to be observed simultaneously. 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* by the dynamic memory.

Fig.3 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.3 (a), for example, shows a pair of temporal sequences of 3D view line data 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.3 (a)). In the actual example shown in Fig.3 (b), the manager reads $\bar{L}_1(t_3)$ and $\bar{L}_2(t_3)$ from its dynamic memory to adjust observation timings.

**(b) Temporal object identification**

The virtual synchronization is also effective in temporal object identification. 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.

**(c) Object identification between agencies**

Since all agency managers work autonomously, the 3D positions of their targets are reconstructed at different moments and compared with each other for object identification. This problem can be solved with the virtual synchronization in the same way as temporal object identification in the intra-agency layer. 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.

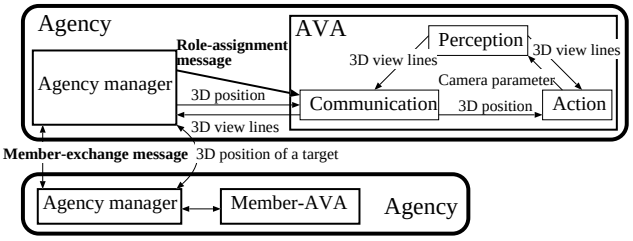
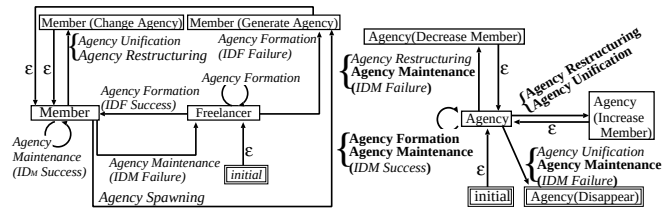



**Figure 4. Elements of communications at and between the three layers.**




(a)State transition of an AVA (b)State transition of an agency

**Figure 5. All the state transitions are caused by communication protocols except for an automatic transition $\epsilon$ that occurs immediately.**

## 4. Soundness of the Dynamic Interaction

In each layer, multiple parallel processes 1) dynamically exchange their information with each other for cooperation and 2) adaptively change their states. The elements of communications are abstracted in Fig.4. These dynamic interaction and state transition have to be realized without any deadlock and delay.

**(1) Intra-AVA layer:** Perception, action and communication modules exchange their information through the dynamic memory in the intra-AVA layer. A reader module runs in parallel to a writer module and tries to read from the dynamic memory the value of the variable at a certain moment according to its own dynamics. Since all information in the dynamic memory is time-series data, namely targets' trajectories and camera parameters (i.e., pan-tilt angles and zooming factor), the dynamic memory can interpolate the value at the specified moment $T$ from its neighboring recorded discrete values when no value is recorded at $T$. Each module can, therefore, obtain information of another module immediately and asynchronously at any time.

In addition, the band-width among modules is high enough to avoid delay because all the modules in an AVA are implemented by threads in a single PC.

**(2) Intra-agency layer:** Fig.5 (a) shows the state transition of an AVA. Data exchanged among an agency manager and its member AVAs is classified into two types: detected object data and messages for communication protocols. Since the former data is exchanged through the dynamic memory possessed by each agency manager, 1) asynchronous message transmission from a member-AVA to its agency man-

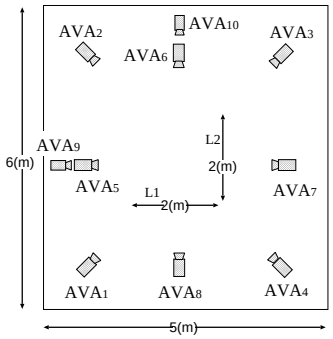



**Figure 6. Experimental environment.**

**Table 1. Errors in identifications [cm].**

| | with | without |
|---|---|---|
| *Spatial identification* | | |
| A | 1.1 | 3.2 |
| B | 1.4 | 4.1 |
| C | 2.0 | 10.3 |
| *Temporal identification* | | |
| A | 0.4 | 2.9 |
| B | 0.7 | 3.8 |
| C | 1.2 | 18.6 |

ager is guaranteed and 2) reliable object identification in the intra-agency layer is realized[2] . The delay of the latter data, on the other hand, may incurs invalid control of a member-AVA: for example, a member AVA may receive a message from its former agency manager due to network congestion. A member-AVA, therefore, communicates only with its current agency manager not to be affected inconsistently by multiple agencies

**(3) Inter-agency layer:** Fig.5 (b) shows the state transition of an agency. Depending on the result of inter-agency object identification, various messages are exchanged between agencies based on inter-agency communication protocols. To avoid a conflict between different interactions, 1) each agency activates a protocol only with a single agency at a certain moment and 2) timeout process is utilized to cope with a message delay, dynamic agency generation and elimination, and other unpredictable factors.

Thus, dynamic interactions in each layer can be reactively realized without inconsistency and deadlock.

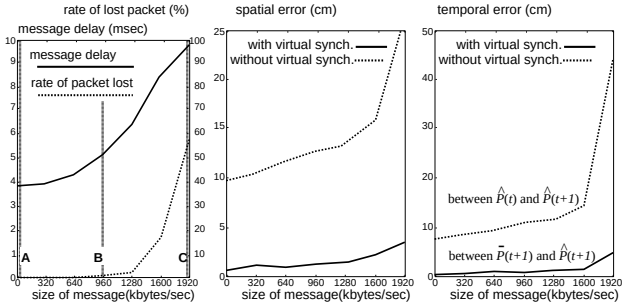
## 5. Experiments

We experimented to verify the effectiveness of the proposed method. We employed ten AVAs. Each AVA is implemented on a network-connected PC (PentiumIII 600MHz $\times$ 2) with an active camera (SONY EVI-G20), where the perception, action, and communication modules as well as agency managers are realized as UNIX processes. In addition, the internal clocks of all the PCs are synchronized by Network Time Protocol to realize the virtual synchronization. Figure 6 illustrates the camera layout in the room.

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

[2] Depending on the band-width among AVAs and agency managers, the performance of the dynamic memory changes (described in Sec.5).




(a) (b) (c)

**Figure 7. (a) Delay of the message (solid line) and Rate of lost packet (dotted line), (b) Error in spatial object identification, (c) Error in temporal object identification.**

a speed of 50[cm/sec] in the observation scene. L1 and L2 in Fig.6 respectively show the trajectories of the robots.

Figure 7 (a) shows variations of network conditions when the size of the vain messages is changed. The error of spatial identification in Fig.7 (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.7 (c) denotes the average distance between the 3D positions of the same target, each of which are reconstructed/estimated at different moments (i.e., $\widehat{P}(t)/\bar{P}(t+1)$ and $\widehat{P}(t+1)$). Table 1 shows the deviations of the error distance in three types of the network conditions. A, B, and C in Fig.7 (a) denote the conditions at which the deviations were computed.

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

## 6. Concluding Remarks

This paper proposed the virtual synchronization for real-time cooperative multi-target tracking with multiple active cameras. Employing the dynamic memory realized the dynamic interactions in each layer without synchronization. As a result, the system is endowed with a high reactiveness.

This work is partly supported by PREST program of JST.

## 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] N. Ukita and T. Matsuyama, "Real-time Cooperative Multi-target Tracking by Communicating Active Vision Agents", *Proc. of ICPR2002*, Vol.2, pp.14–19, 2002.

[3] 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.

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