Real-time Cooperative Multi-target Tracking by Dense Communication among Active Vision Agents

Norimichi Ukita
Graduate School of Information Science, Nara Institute of Science and Technology
ukita@ieee.org

Abstract

We have proposed a real-time multi-target tracking system by cooperative behaviors of Active Vision Agents (AVAs), where an AVA is a logical model of a network-connected computer with an active camera. All AVAs change their roles adaptively by negotiating with each other. The previous system, however, was unable to simultaneously track target objects that are greater in number than the AVAs. In this paper, we realize the increase in number of simultaneously trackable objects by improving cooperative tracking protocols so that AVAs exchange a large amount of information with each other without increasing network traffic. This system enables the increase of trackable objects by (1) an AVA that provides information of observed objects for several agencies (i.e., a group of AVAs that track the same target) and (2) a vacant-agency that receives information of its target from AVAs tracking other objects. Experimental results demonstrate that our system can track multiple targets (greater in number than the cameras) simultaneously and robustly in real time.

1. Introduction

Object tracking is one of the most important technologies for realizing real-world vision systems (e.g., visual surveillance systems, ITS (Intelligent Transport System) and so on). Although several tracking systems have been developed, these systems have several limitations with regard to their resources and functions (e.g., the number of cameras, their characteristics, and the number of trackable objects). In [2, 3, 4], multi-target tracking with distributed fixed cameras has been proposed to enlarge the observation area. However, since only fixed cameras, which cannot control their view directions and zooming factors, are employed, the resolution of the observed image as well as the observable area are limited. These limitations regarding the system functions damage the flexibility and adaptability of the system and reduce the types of tasks and environments for which the system is applicable.

Accordingly, we have proposed a cooperative multi-target tracking system that can be applied to various tasks and can cope with dynamic situations in a complicated scene[1]. This system consists of multiple Active Vision Agents (AVAs), where an AVA is a logical model of a network-connected computer with an active camera. Spatially distributed AVAs allow us to acquire the 3D information of observed objects and track multiple targets continuously in a wide area. In this system, however, minimum information required for object identification among AVAs is exchanged among them. This insufficient communication results in the following disadvantages:

1. The upper limitation of trackable objects: The system cannot simultaneously track objects that are greater in number of than the AVAs. This is because each target must be gazed on by at least one AVA in order to track a target continuously.
2. Unstable tracking: Each AVA dumps the information of observed non-target objects while tracking its target. As a result of insufficient information of the targets, the system may lose track of them due to occlusion or other unpredicted phenomena.

For actual applications, a system with a small number of cameras for tracking many objects is desired. In addition, the system should acquire a large amount of information of each object observed from various directions in order to improve tracking stability.

In this paper, therefore, we realize the increase in number of trackable objects and tracking stability on the basis of the following ideas:

1. A target does not have to correspond to any AVA. Instead, a software agent is generated dynamically to collect the information of an object whenever an object is newly detected.
2. An AVA provides the information of all observed objects to each group of AVAs that tracks each of those objects.

2. Real-time Cooperative Multi-target Tracking

This section describes the outline of the previous system[1] in order to point out its shortcomings.

2.1. System Organization and Scheme

Each AVA consists of a Fixed-Viewpoint Pan-Tilt-Zoom camera and a network-connected computer. This network is not a special close network (e.g., PC cluster) but an open network. With this camera, an AVA can track a moving object independently. Since all AVAs work while keeping their own intrinsic dynamics, the images are observed asynchronously by them.

In our system, multiple AVAs, whose 3D positions are known, are embedded in the real world and observe a wide area. Following are an outline of the scheme of the tracking system:

1. Initially, each AVA independently searches for an object that enters the observation area.
2. If an AVA detects a target, it navigates the gazes of other AVAs toward the target based on the estimated 3D information of the target[1].
3. AVAs that gaze at the same object track the focused target cooperatively by exchanging its detected information. A group of AVAs that track the same object is called an *Agency*.
4. Depending on the target motion, each AVA dynamically changes its target.
5. When the target exits the scene, each AVA decides whether to either search for an object or join another agency depending on the situation.

2.2. Three-layered Dynamic Interaction for Cooperative Tracking

For object tracking, object identification is significant. We, therefore, classify the system into three layers depending on the types of object information employed for identification. Each layer corresponds to elements in the system as follows: (Fig.1):

Intra-AVA layer: An AVA (bottom in Fig.1)

Intra-agency layer: An agency (middle in Fig.1)

Inter-agency layer: The total system (top in Fig.1)

While all agents exchange information in order to decide their roles, the decision is not the optimal solution in the total system and their self-decisions may possibly be inconsistent. Our system copes with this problem using the cooperative tracking protocols that allow the agents to achieve real-time cooperation [2].

1 When only one AVA detects an object, the gaze direction of another AVA is controlled along the 3D view line that is from the camera of the AVA, which detects the object, to the object. On the other hand, when multiple AVAs detect an object, its 3D position can be computed and the gaze direction of another AVA is navigated toward the 3D point.

2 For all agents to work consistently taking into account the optimized behaviors, distributed negotiation among agents should be performed (see [7, 8], for example).

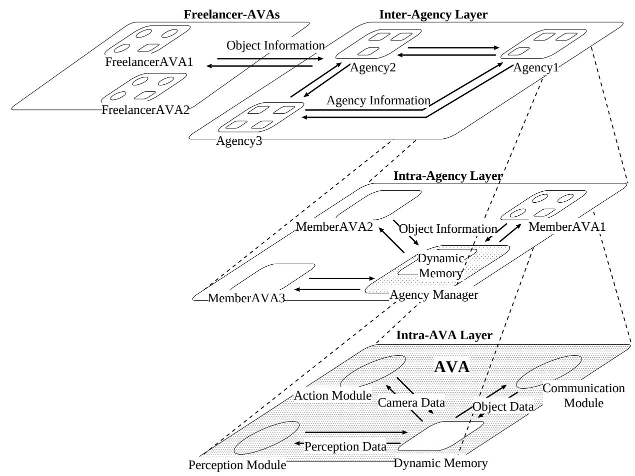


Figure 1. Three layers in the system: Intra-AVA (bottom), Intra-Agency (middle), and Inter-Agency (top) layers.

2.2.1. Intra-AVA layer

In the bottom layer in Fig.1, the perception, action, and communication modules that compose an AVA exchange the time-series information with each other via the *dynamic memory*[9] possessed by each AVA. The interaction among three modules allows the AVA to track its target independently or cooperatively.

2.2.2. Intra-Agency layer

In the middle layer in Fig.1, member-AVAs belonging to the same agency exchange the information of detected objects for the following two types of object identification.

Spatial object identification The agency has to establish object identification between the 3D view lines detected by its member-AVAs. If the distance between the 3D view lines detected by different AVAs is less than a threshold, these 3D view lines are considered as the information of the same object. The intersection of the identified 3D view lines is regarded as the 3D position of the object.

Temporal object identification To gaze at the target continuously, the agency compares the 3D trajectory of the target with the 3D positions of the objects newly computed by spatial object identification. When multiple 3D locations are obtained by spatial object identification, the agency selects the one closest to the target trajectory.

These identifications are executed by an *Agency Manager* (AM). It is an autonomous software agent working as the delegate of an agency, and performs (1) management of the dynamic memory in each agency,

(2) object identification and communication with agencies and AVAs, and (3) dynamic agency organization.

Depending on whether or not the above two types of identification are successful, the following three types of cooperative tracking protocols are activated:

Agency Formation defines the new agency generation by a freelancer-AVA. When a freelancer-AVA finds a new object, it requests object identification from existing agencies between the newly detected object and the target of each agency. If no agency establishes successful identification, the freelancer-AVA generates a new AM and joins this agency. If an agency establishes successful identification, the freelancer-AVA joins that agency.

Agency Maintenance defines the cooperative tracking in an agency. If temporal identification between the target of the agency and an object detected by member-AVA$_m$ fails, the AM reports the 3D position of the target to member-AVA$_m$ in order to navigate the gaze of member-AVA$_m$ toward the target. Nevertheless, if the failure of identification continues for a long time, the AM expels member-AVA$_m$ from the agency. If none of the member-AVAs are able to observe the target, the AM eliminates the agency.

Agency Spawning defines the generation of a new agency from an existing one. After temporal object identification, the AM may find such a 3D view line(s) that does not correspond to the target; this represents the detection of a new object by its member-AVA. Let L_n denote such a 3D view line detected by AVA$_n$. The AM then requires the other agencies to compare L_n with their own targets for object identification. If no identification is successful, the AM orders member-AVA$_n$ to generate a new agency and join it.

2.2.3. Inter-Agency layer When an AM newly obtains the 3D position of its target, it broadcasts the position to the others. Another AM, which receives this information, compares it with the 3D position of its target to check object identification. Depending on the result of this object identification, either of the following cooperative tracking protocols is activated.

Agency Unification is activated when the result of the inter-agency object identification is successful. It performs the merging procedure of two agencies, both of which happen to track the same object due to failure of object identification.

Agency Reconstruction is activated when the result of inter-agency object identification is unsuccessful. It performs the dynamic interchange of member-AVAs between agencies. The AM determines "which AVA belongs to which agency" taking into account the given task specification described in [1].

An AM communicates not only with other AMs but also with all freelancer-AVAs (top row in Fig.1) to achieve the following functions:

- Maintain consistency of the one-to-one correspondence between an agency and its target: Only if no agency tracks the object detected by a freelancer-AVA, the system can generate a new agency.
- Investigate the state of the system: When a message from a freelancer-AVA is received, the number of freelancer-AVAs can be checked. Similarly, when a message from an AM is received, the number of member-AVAs can be checked.

In our system, communication between freelancers and AMs is implemented by broadcast messages in order to maintain continuous communication despite AVAs change their roles dynamically and AMs generate and disappear dynamically. Although the information of detected objects for object identification should be exchanged frequently, the interval of sending this broadcast message can be short; for example, one second in our experimental system. Moreover, periodic broadcasting allows the system to know the dynamic participation and secession of an AVA and an AM (e.g., a newly added AVA and a halt of an AVA/AM due to hardware trouble).

3. Augmented Cooperative Tracking Protocols

3.1. Dense Communication through Supporter-AVAs

In the previous system[1], each AVA can belong only to one agency and send information of detected objects only to the agency. That is, the information of objects detected by that AVA is not provided to other agencies. Therefore, even if the AVA detects the target of another agency, the information of the target is not provided to the corresponding agency. The previous system is designed as mentioned above in order to implement the following functions:

- An AVA can belong to only one agency: An AVA has its active camera for continuously observing high-resolution images of its target. If an AVA belongs to multi agencies, they may control the camera inconsistently in tracking their targets.
- A member-AVA sends the information of detected objects only to its agency: To suppress network load, a member-AVA sends the information of detected objects only to the agency that definitely

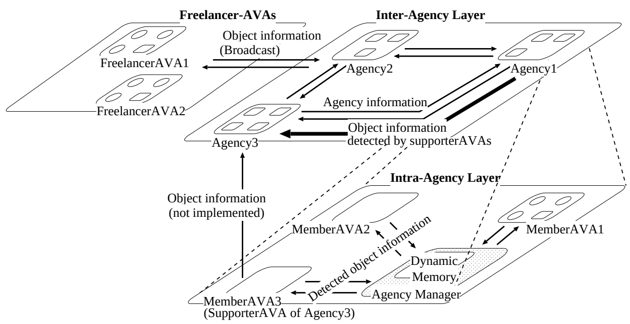


Figure 2. Communications in the Intra-agency and Inter-agency layers.

requires the information, namely, the agency that it belongs to.

If the system employs static sensors (e.g., [2, 3, 4]), the former problem is not raised. This is because their sensing ranges are fixed and every agency can share them simultaneously. In the system with active (i.e., pan-tilt-zoom) cameras for observing high-resolution images of targets, on the other hand, a sophisticated mechanism for resource allocation is required. We realize this mechanism by the cooperative tracking protocols and this is an essential difference between our system and other similar systems.

Multiple objects may be observed even while a member-AVA controls its camera to gaze at its target. By employing the information of all the observed objects, the performance of object identification (i.e., tracking stability) can be improved.In this paper, we augment the system so that a member-AVA can provide the object information to multiple agencies while only a single agency, which the member-AVA belongs to, can control the camera of the member-AVA. A member-AVA providing the object information to agency$_s$, which is not the member's agency, is called a *Supporter*-AVA of that agency$_s$. This agency$_s$ is called a *Supported*-agency of the supporter-AVA.

Next, we describe how to send the object information from a supporter-AVA to its supported-agency.

An AM is a software agent, which is generated and halted dynamically. It is necessary for another agent to know the information regarding one-to-one communication with an AM via a network (e.g., hostname and port numbers) in order to exchange information with the AM excepting the case of broadcasting. It should be noted that one-to-one communication is demanded in order to reduce network load instead of broadcasting if the message is necessary only to an AM. For efficient message exchange among distributed agents, a shared memory system or a tuple-space system can be employed (see [5, 6], for example). For real-time tracking with active cameras, however, reactive communication is required to follow target motions. Therefore, every agent in our system exchanges information through active message transmission without a memory space.

Every member-AVA is informed of the information required for one-to-one communication with its AM when it becomes the member of the agency, because it has to send the information of detected objects to its AM. In addition, all AMs are required to communicate with each other through one-to-one communication in order to perform the cooperative tracking protocols in the inter-agency layer. As mentioned earlier, all agencies broadcast messages at regular intervals. By including the information for one-to-one communication into the broadcast message, one-to-one communication between agencies can be realized. The above two types of one-to-one communications ("communication between a member-AVA and its AM (intra-agency layer)" and "communication between agencies (inter-agency layer)," which are indicated as *Detected object information* and *Agency information*, respectively, in Fig. 2) are inevitable functions for our cooperative tracking with the three-layered architecture and have been implemented in the previous system[1]. To realize newly required communication between a supporter-AVA and its supported-agency, one-to-one communication between them is necessary. However, the direct communication between them as well as object identification and agency reformation based on the communication results in the following problems:

- All member-AVAs must receive messages from all AMs and handle the information for communicating with them dynamically. This redundant process is avoidable if all member-AVAs broadcast the information of detected objects to all AMs. The broadcast, however, causes high traffic congestion. Note that a member-AVA communicates only with its AM in the previous system[1].
- If each member-AVA establishes object identification independently and moves between agencies based on the result of the identification, the information regarding the target and the agency organization may be inconsistent among the member-AVAs in an agency and its AM. To avoid this inconsistency in the information managed by the agency as well as network congestion, a member-AVA should entrust negotiation with other agencies to its AM.

Therefore, the information from a supporter-AVA to its supported-agency is transmitted via the AM of the supporter-AVA as indicated by "Object information detected by supporter-AVAs" in Fig. 2.

A freelancer may also observe a target object that is tracked by an agency. As mentioned in Sec. 2.2.3, a freelancer-AVA broadcasts the information of detected objects and every AM receives this information as indicated by "Object information (Broadcast)" in Fig. 2. That is, every agency can receive the object information transmitted by all freelancer-AVAs.

Accordingly, the difference in communication mechanisms between the new system and the previous one illustrated in Fig. 1 is merely the forwarding of object information from an AM to another AM (indicated by a thick line in Fig. 2). Section 3.5 will describe the details of how to realize the functions of a supporter-AVA.

3.2. Tracking with Vacant-agencies

In contrast to the previous system, supporter-AVAs enable each agency to obtain a large amount of information of its target, which can improve tracking stability. However, the system cannot simultaneously track targets that are greater in number of than the AVAs under the following conditions: (1) there is a one-to-one correspondence between a target and an agency and (2) each agency has at least one member-AVA. Here we consider the necessity of these two conditions for our objective:

1. An agency (AM) manages all information of its target object exclusively and performs object identification of the target by itself. The one-to-one correspondence is, therefore, essential.
2. An agency that tracks an object that is accorded higher priority should have several member-AVAs, and conversely, a small number of AVAs should track an object that is accorded lower priority. This function has been realized by the object-priority introduced in [1]. In the previous system, at least one member-AVA belongs to an agency for gazing at its target continuously even if the agency tracks a target that is given the lowest priority. However, continuous observation of a target that is given a fairly low priority is not necessarily required; intermittent observation by AVAs tracking other targets is sufficient. All agencies, therefore, do not have to possess a member-AVA.

Based on the above discussion, we newly define an agency without any member-AVA, which is called a *vacant-agency*. The vacant-agency acquires the information of its target from freelancer-AVAs and member-AVAs in other agencies.

To introduce supporter-AVAs and vacant-agencies in our system, the agency formation and maintenance protocols have to be augmented and a new protocol

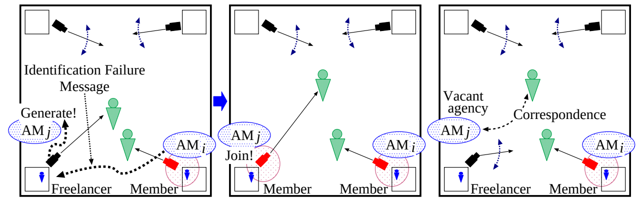


Figure 3. Augmented agency formation.

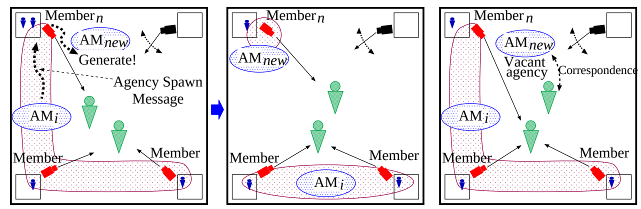


Figure 4. Augmented agency spawning.

for forwarding object information via an AM, called the *agency support* protocol, is required.

3.3. Augmented Agency Formation

When an object is newly detected, the previous system works as follows: if a new agency cannot be generated due to an insufficient number of freelancer-AVAs, the newly detected object is not tracked, or one of the existing agencies halts to generate a new agency for tracking the newly detected object.

The proposed system, on the other hand, can theoretically generate an infinite number of vacant-agencies. When a freelancer-AVA finds a new object, it requests all agencies to identify the object with their targets. After this request, the freelancer-AVA works as follows:

1. If an agency establishes successful identification, no agency is generated and the subsequent behaviors are identical to those of the previous system[1].

2. If no agency establishes successful identification, the freelancer-AVA invariably generates a new agency (indicated by AM_j in Fig. 3 (1)). After generating the agency, the freelancer-AVA determines whether to join the agency (Fig. 3 (2-a)) or to continue searching for objects as a freelancer-AVA (Fig. 3 (2-b)) depending on the ratio of freelancer-AVAs to member-AVAs, which is given in advance by a user. If it remains a freelancer-AVA, the newly generated agency works as a vacant-agency.

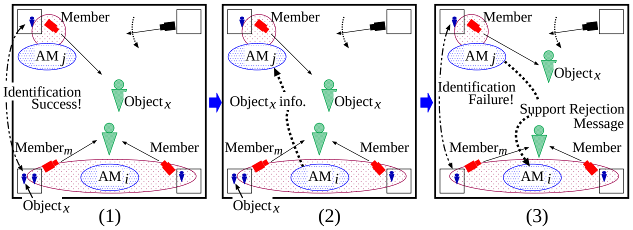


Figure 5. Agency support.

3.4. Augmented Agency Spawning

The agency spawning protocol is modified in a manner similar to that of the agency formation protocol: the system invariably generates a new agency when an object is newly detected.

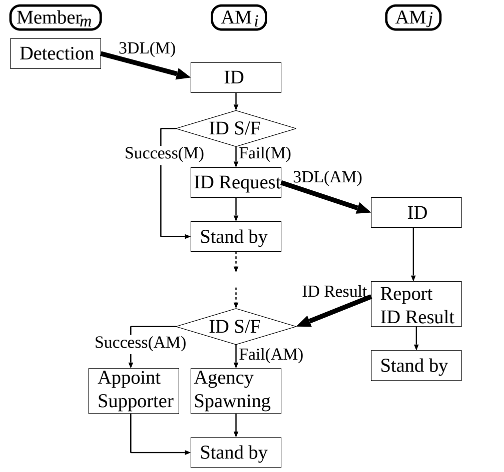
If an AM (denoted by AM_i in Fig. 4) finds that one of the objects detected by its member-AVA_n (denoted by object_{new}) is not tracked by any agency, AM_i generates a new agency (denoted by AM_{new} in Fig. 4) for tracking object_{new} (Fig. 4 (1)). Member-AVA_n determines whether to join agency_{new} (Fig. 4 (2-a)) or to remain in agency_i (Fig. 4 (2-b)) according to the given task specification introduced in [1].

3.5. Agency Support

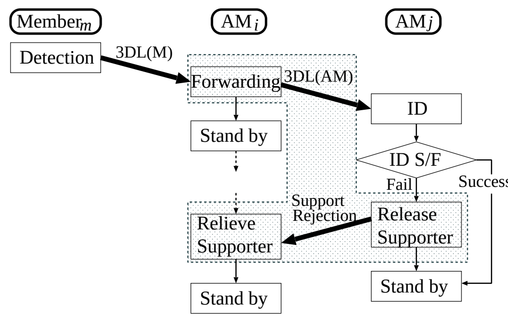
This protocol defines how object information detected by member-AVAs is sent to multiple agencies through the AM to which the member-AVAs belong.

To identify a member-AVA that can work as a supporter-AVA for another agency, the AM has to establish identification between the objects detected by its member-AVAs and the targets of other agencies. This identification can be activated by sending the information of the detected objects from the AM to the other AMs. In an example in Fig. 6 (a), object information "3DL(M)" that is sent from member-AVA_m to AM_i is forwarded to the other AMs (e.g., AM_j) by "3DL(AM)" in order to request identification from the other agencies. This identification procedure is defined in the agency spawning protocol. Then, the result of the identification (denoted by "ID Result" in Fig. 6) is sent to AM_i from AM_j. If object_x detected by member-AVA_m is identified with the target of agency_j, AM_i assigns member-AVA_m to a supporter-AVA of agency_j (Fig. 5 (1) and "Appoint Supporter" in Fig. 6). Hereafter, AM_i forwards the object information detected by member-AVA_m to AM_j (Fig. 5 (2) and "Forwarding" in Fig. 6) until member-AVA_m withdraws from agency_i, the message indicating that the supporter is not required ("Support Rejection") is sent from AM_j to AM_i, or agency_j disappears.

When AM_j receives the forwarded information of detected objects from AM_i (i.e., "3DL(AM)" in Fig. 6 (b)), AM_j establishes object identification between these objects and its target (shown by "ID" in Fig. 6 (b)). If member-AVA_m does not observe the target of agency_j and then this identification is not successful (shown by "Fail" in 6 (b)), AM_j sends the message of "Support Rejection" to AM_i (Fig. 5 (3) and Fig. 6) while AM_j controls the gaze direction of the AVA that lost track of the target if it is working as a member-AVA of AM_j.



(a) Appointing a supporter-AVA



(b) Forwarding the information of detected objects and relieving an AVA of a supporter

Figure 6. Process flow diagram of the agency support: Thin and thick arrows represent state transition and communication flows.

3.6. Theoretical Study on the Communication Cost

The number of communications for exchanging object information, which are produced whenever a member-AVA detects objects, is represented by the equation below:

$$1 + (1 - P_{td}) + 2P_{td}N_a + 2P_{nd}N_a + P_{sa}N_a, \quad (1)$$

where these terms denote the following:

1: Detected object information sent from a member-AVA to its AM (denoted by "3DL(M)" in Fig. 6 (a)(b)).

$1 - P_{td}$**:** Gaze navigation for a member-AVA by its AM. P_{td} denotes the probability that identification between the target and one of objects detected by the member-AVA is successful.

$2P_{td}N_a$**:** Target information exchanged in the inter-agency layer (i.e., from an AM to other AMs) and the reply including the result of object identification. N_a denotes the number of agencies.

$2P_{nd}N_a$**:** Information of an object newly detected by a member-AVA, which is sent from an AM to other AMs, (denoted by "3DL(AM)" in Fig. 6 (a)) and the reply including the result of object identification (denoted by "ID Result" in Fig. 6 (a)). P_{nd} denotes the probability that a member-AVA detects a new object.

$P_{sa}N_a$**:** Detected object information forwarded from an AM to a supported-AM (denoted by "3DL(AM)" in Fig. 6 (b)). P_{sa} denotes the probability that each supporter-AVA detects the target of its supported-AM.

In the proposed system, "Appoint Supporter" in Fig. 6 (a) and a shaded region in Fig. 6 (b) are added to the previous system[1]. Therefore, only $P_{sa}N_a$ is added for the agency support protocol. Furthermore, the magnitude relation among the probabilities is $P_{nd} < P_{sa} << P_{td}$ in most cases. Accordingly, we can confirm that the increase of communications for the agency support protocol is quite smaller than the amount of communications in the previous system.

4. Experiments

We conducted experiments to verify the effectiveness of the proposed system. Each AVA was implemented on a network-connected PC (Xeon 1.5GHz $\times$ 2) with an active camera (SONY EVI-G20), where the perception, action, and communication modules as well as AMs were realized as UNIX processes. The communication module sends messages using UDP via a 100M bps network. In addition, the internal clocks of all the PCs were synchronized by NTP[10] to realize the virtual synchronization[9]. With this architecture, the perception module can capture images and detect objects in the observed image at approximately 0.1 sec intervals on an average.

To confirm the effectiveness of the augmented cooperative tracking protocols (i.e., (1) tracking objects greater in number than the AVAs and (2) improving tracking stability), we conducted experiments under the following conditions:

- Track four objects using three AVAs.
- The number of freelancer-AVAs and member-AVAs is 0 and 3, respectively.

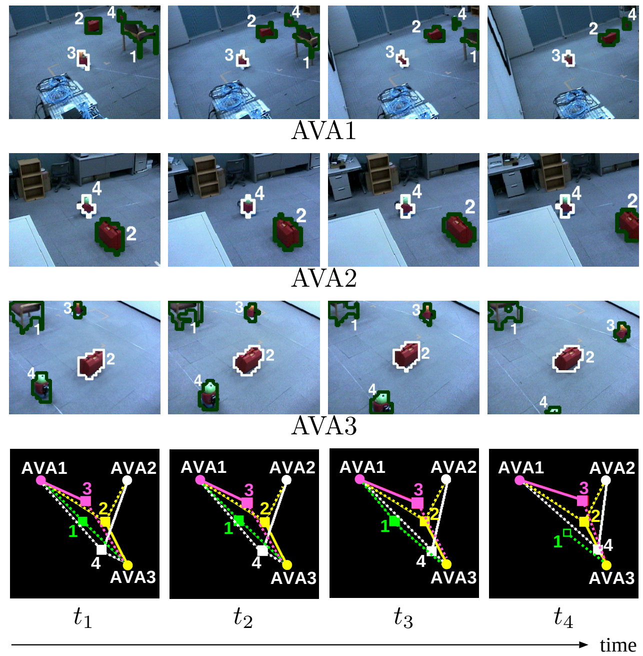


Figure 7. Upper: partial image sequences (0.5 sec intervals on average). Lower: agency organizations and the target positions. Each number indicates the ID of each target.

To verify the actual bandwidth requirements, we measured the amount of messages in the system. In the experiments, the size of a message exchanged among agents was about 6k bytes and the average amount of messages for tracking an object with an AVA was about 60k bps (48k bps in the previous system[1]) in the total system. Since the amount of messages is roughly proportional to the numbers of AVAs and targets, the bandwidth of a common network system is sufficient for the proposed system.

The upper part of Fig. 7 shows the partial image sequences observed by each AVA. The images on the same column were taken at almost the same time. The regions enclosed by lines in the images show the detected object regions and the region with a white line indicates the target at each moment. The lower part of Fig. 7 shows the role of each AVA and the agency organization at the moment when the same column of images in the upper part were observed. The circles and squares indicate the positions of AVAs and the computed locations of targets, respectively. Each line segment between an AVA and an object indicates that the AVA provides the information of the object to the agency that tracks it. The solid and dotted lines imply that the AVA provides the information of corresponding objects to each agency as a member and a supporter, respectively. For example, AVA_1 detected all objects excepting at t_1, t_2 and t_3 and worked as a supporter-AVA of every agency while working as a member-AVA of the agency tracking $object_3$. Since $object_1$ exited the visual field of AVA_1 at t_4, AVA_1 ceased being a supporter-AVA of the agency tracking $object_1$. We can confirm that the information of all objects detected by each AVA is provided to corresponding agencies and employed to estimate 3D information of each target. As a result, three AVAs could track four target objects simultaneously. That is, the system could continue acquiring the information of objects that are greater in number of that the AVAs.

Next, we conducted experiments with the previous system[1] and the proposed system in the same situation; multiple objects repeated several motions. The number of AVAs that provide the object information to each agency, namely, the transitions of the number of member-AVAs in each agency, were investigated while each system worked for a minute. In general, with an increase in the number of AVAs gazing at an object, the system can observe the object from various directions. Such diverse observations can improve the accuracy of estimating 3D information and enable continuous gazing at the target without being disturbed by obstacles or other moving objects. The mean values in the previous and proposed systems were 0.6 and 2.4, respectively. In addition, we evaluated the success rates of spatial and temporal object identifications:

- When multiple objects are observed by a member-AVA, its AM identifies the target among the observed objects. The success rates of this spatial object identification in the previous and proposed systems were 69% and 93%, respectively.
- Since the system can compute the sequential 3D positions of each target at short time intervals, the computed trajectory should be smooth. However, when a small number of AVAs observe an object that is occluded by other objects in the observed images, its sequential positions computed by the system sway suddenly due to the errors not only of spatial identification but also of 3D position estimation. This results in the error in temporal identification. The success rates of temporal identification in the previous and proposed systems were 74% and 96%, respectively.

Note that in these evaluations, the success rate of identification in the previous system was especially low because a small number of AVAs were employed. Accordingly, we can confirm the superiority of the proposed system in terms of tracking stability.

5. Concluding Remarks

This paper proposed a real-time cooperative multi-target tracking system with multiple active cameras. The proposed system is endowed with the following augmentations by a dense and flexible communication mechanism for exchanging the information of detected objects between agents:

- Improvement of tracking stability by diverse observations
- Elimination of the restriction on the number of simultaneously trackable objects

With these augmentations, our tracking system can be applied to extensive real-world vision systems.

This study is supported by PRESTO program of Japan Science and Technology Agency (JST) and the Grant-in-Aid for Scientific Research (No.15700157).

References

[1] N. Ukita and T. Matsuyama: "Real-time Multi-target Tracking by Cooperative Distributed Active Vision Agents," in *Proc. of AAMAS 2002*, pp.829–838, 2002.

[2] L. Lee, R. Romano, and G. Stein, "Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame," *IEEE Transaction on PAMI*, Vol.22, No.8, pp.758–767, 2000.

[3] B. Horling, R. Vincent, R. Mailler, J. Shen, R. Becker, K. Rawlins, and Victor Lesser, "Distributed Sensor Network for Real Time Tracking," in *Proc. of the 5th ICAA*, pp.417–424, 2001.

[4] K. C. Ng, H. Ishiguro, M. M. Trivedi, and T. Sogo, "An Integrated Surveillance System-Human Tracking and View Synthesis using Multiple Omni-Directional Vision Sensors," *Image and Vision Computing*, Vol.22, No.7, pp.551–561, 2004.

[5] J. Protic, M. Tomasevic, and V. Milutinovic, "Distributed Shared Memory: Concepts and Design," IEEE Computer Society, 1997.

[6] W. Vasconcelos, *et al*, "Rapid prototyping of large multi-agent systems through logic programming," Annals of Mathematics and Artificial Intelligence, Vol. 41, No. 2–4, pp.135–169, 2004.

[7] K. Hirayama and M. Yokoo, "An approach to over-constrained distributed constraint satisfaction problems: Distributed hierarchical constraint satisfaction," in *Proc. of ICMAS*, pp.135–142, 2000.

[8] R. Mailler, V. Lesser, and B. Horling, "Cooperative Negotiation for Soft Real-Time Distributed Resource Allocation," in *Proc. of AAMAS 2003*, pp.576–583, 2003.

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

[10] D. L. Mills, " Internet time synchronization: the network time protocol," *IEEE Transaction of Communications*, Vol.39, No.10, pp.1482–1493, 1991.