Machine Vision and Applications manuscript No.
(will be inserted by the editor)

Norimichi Ukita

Probabilistic-Topological Calibration of Widely Distributed Camera Networks

Received: date / Accepted: date

Abstract We propose a method for estimating the topology of distributed cameras, which can provide useful information for multi-target tracking in a wide area, without object identification among the FOVs of the cameras. In our method, each camera first detects objects in its observed images independently in order to obtain the positions/times where/when the objects enter/exit its FOV. Each obtained data is tentatively paired with all other data detected before the data is observed. A transit time between each paired data and their *x*-*y* coordinates are then computed. Based on classifying the distribution of the transit times and the *x*-*y* coordinates, object routes between FOVs can be detected. The classification is achieved by simple and robust vector quantization. The detected routes are then categorized to acquire the probabilistic-topological information of distributed cameras. In addition, offline tracking of observed objects can be realized by means of the calibration process. Experiments demonstrated that our method could automatically estimate the topological relationships of the distributed cameras and the object transits among them.

Keywords Distributed camera calibration · Route between fields of view · Object tracking

1 Introduction

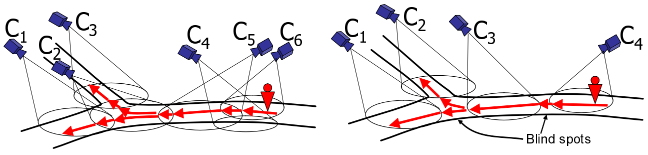
Object tracking is a fundamental technology for realizing real-world vision systems. In particular, multi-target tracking with distributed cameras is useful for enlarging observation areas and observing targets from omnidirections.

For target tracking, object identification is crucial. If fields of view (FOVs) of cameras are overlapped and their extrinsic parameters are known, object identification can be simplified by analyzing consistencies in 3D information of the observed objects. (e.g., using multiple fixed cameras[1,2], omnidirectional cameras[3], and active cameras[4,5]). Several methods for calibrating the extrinsic parameters of widely distributed cameras have been proposed; for example, initial calibration with observation results of moving objects (for synchronized cameras[6] and asynchronized cameras[7]), a method for improving the initial results[8], and calibration using landmarks with known 3D positions measured by GPS[9]. For all of these methods, the cameras must be positioned such that an object can move through the FOVs of the cameras without going outside of the overall FOV boundaries as illustrated in Fig. 1 (a). This assumption makes it practically impossible to employ a camera system for observing wider areas because numerous cameras are required.

Accordingly, camera configuration without overlapping FOVs, namely with blind spots (shown in Fig. 1 (b)), is necessary. Object identification in this environment is obviously more difficult than that in the overlapping FOVs. In this challenging problem, the topology of FOVs and its probabilistic information (e.g., presence of a route, transit times, and transit probabilities between FOVs) can improve th performance of identification (see [10,11], for example) This information is given manually in the previous methods[10,11]. Manual estimation is possible for a small number of cameras. However, as the observation area grows and the number of cameras increases, the topology becomes drastically more complex. Furthermore, automatic estimation is required to realize an online system that can update the information to cope with hardware troubles and adapt to changes in a scene. Therefore, a method for automatically estimating the topology is desired.

The above discussion covers only the camera systems with completely overlapping or isolated FOVs. In actual applications, ‘areas in which dynamic situations should be observed in detail from omnidirections by employing as many cameras as possible’ and ‘areas in which

Norimichi Ukita
8916-5 Takayama, Ikoma, Nara, Japan
Tel.: +81-743-72-5263
Fax: +81-743-72-5269
E-mail: ukita@ieee.org



(a) Overlapping configuration (b) Isolated configuration

Fig. 1 FOV configuration in a distributed camera system: Each ellipse and arrow depicts a FOV and a route, respectively. Objects go along trajectories, each of which consists of concatenated routes.

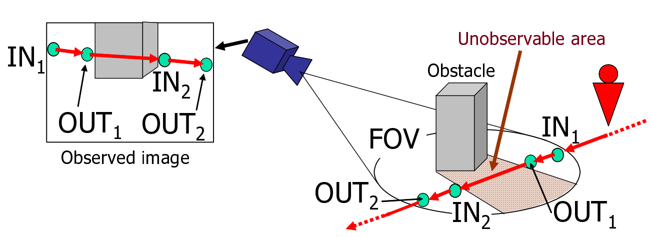


Fig. 2 IN/OUT positions and routes of a moving object: Each dot indicates an IN/OUT position.

comprehensive observation such as trajectory tracking is sufficient' are mixed in an environment. In this study, therefore, we focused on how to automatically estimate the topology of distributed cameras with overlapping and isolated FOVs. We also show that offline tracking of moving objects can be achieved in the process of the calibration scheme.

2 Issues and Goals of Calibrating Distributed Cameras

The topology of cameras can be defined by their FOVs and the routes among and within them as follows:

FOV: The FOV of a camera is defined by a 3D region visible from the camera as shown in Fig. 2. A region occluded by obstacles (e.g., 'Unobservable area' in Fig. 2) is outside the FOV of the camera even if the region is within the image boundary.

Route: Events regarding entrance and exit in FOVs are defined as follows:

IN: An event such that 'an object that was outside the FOV at the previous capturing timing is newly detected' is called *IN*.

OUT: An event such that 'an object that will leave the FOV at the next capturing timing is last detected' is called *OUT*.

In the example illustrated in Fig. 2, IN_1, IN_2, OUT_1, and OUT_2 denote the positions where an object enters and exits a FOV, respectively. Consecutive two points observed at IN/OUT events compose a route. The earlier of the two points observed is called a *beginning point*. The other is called an *end point*. That is, (1) each route is defined only by its beginning and end points, and (2) object trajectories between and within FOVs are not represented by the route information. In Fig. 2, three routes (i.e., $\overrightarrow{IN_1 \cdot OUT_1}, \overrightarrow{OUT_1 \cdot IN_2}, and \overrightarrow{IN_2 \cdot OUT_2}$, where $\overrightarrow{x \cdot y}$ means a route from the beginning point x to the end point y.) exist. The data observed at IN and OUT events (i.e., image coordinates $\boldsymbol{P}(C)$ of camera C and time T at each entrance/exit event) are called *IN data* and *OUT data*, respectively. In this research, all beginning and end points are represented by 2D coordinates. That is, route information is represented only by 2D information.

Useful information for tracking among distributed cameras can be categorized as follows:

Class-V (Relationships between FOVs): For every possible pair of cameras (denoted by C_i and C_j, where subscripts i and j denote the IDs of any cameras), the following information is provided:

V1: Presence or absence of route(s) between the FOVs of C_i and C_j. In the example illustrated in Fig.1 (b), each of 'C_1 and C_3', 'C_2 and C_3', and 'C_3 and C_4' has a route.

V2: Presence or absence of overlapping area(s) between the FOVs of C_i and C_j.

V3: If overlapping: the extrinsic parameters (i.e., rotation and translation) between C_i and C_j.

V3 can be estimated by the above calibration methods[6–9]. On the other hand, a method for acquiring V1 and V2 has to be newly invented.

Class-R (Characteristics of routes): Variables C, P, and T with superscripts B and E denote cameras/coordinates/times of the beginning and end points, respectively. When IN/OUT data is observed in image coordinates $\boldsymbol{P}^E(C^E)$ of camera C^E at time T^E, this data must be identified with the IN/OUT data of the same object, which was observed right before T^E, in order to achieve object tracking. For this object identification, IN/OUT data of identification candidates can be reduced as follows; the identified data was detected (1) near the beginning point of the routes having the end point around $\boldsymbol{P}^E(C^E)$ and (2) at about $T^E - T$ if T is required to go through the route where the object was passed. Following stochastic representations of these constraint conditions are given as the characteristics of a route having the end point around $\boldsymbol{P}^E(C^E)$:

R1: Probability that an object detected in $\boldsymbol{P}^E(C^E)$ went through the route having the beginning point $\boldsymbol{P}^B(C^B)$.

R2: Probability that it takes an object $T^E - T^B$ to go through route $\overrightarrow{\boldsymbol{P}^B(C^B) \cdot \boldsymbol{P}^E(C^E)}$, where T^B denotes the time when the object was observed in $\boldsymbol{P}^B(C^B)$.

We call the above information the *probabilistic-topology* of cameras.

To estimate the above class-V and class-R information, valuable information can be obtained from tracking results among and within FOVs (see [12], for example). However, object identification between images observed by isolated cameras only based on image analysis (e.g., face/object recognition) is obviously very difficult. For reliable identification among FOVs, the following approaches have been proposed:

Tracking without identification: In [13], it is assumed that there is only one object moving in an environment during a training period. Otherwise, an object trajectory can be obtained by observing a moving landmark that is easily tracked (e.g., LED)[14] even if other objects exist during a training period. These approaches enable easy identification among isolated cameras. Class-R information, however, cannot be estimated because it can only be acquired by analyzing a number of real object trajectories.

Improving robustness of identification: In [15], robust identification between isolated FOVs is achieved by (1) employing information only of easy-identifiable objects and (2) matching all data of entering and exiting FOVs by using a bipartite graph. However, (1) sufficient class-R information is unavailable from object information selectively obtained by this algorithm and (2) a bipartite graph cannot express any loop route from/to the same FOV and any route between overlapping FOVs.

Accordingly, several limitations in these approaches preclude the acquisition of a sufficient amount of object data required for estimating class-R information. It should be also noted that the whole class-V information cannot be estimated even if object identification is completely established with these approaches. In fact, these approaches[13–15] can estimate only the positions of beginning/end points of routes (i.e., V1 information and partial R1 information).

While tracking among isolated FOVs is very troublesome, large numbers of sophisticated methods have been proposed for object detection and tracking within a FOV (e.g., robust foreground-object detection under non-stationary scenes[16] and occlusion-robust tracking[17]) Even with these methods, it is still difficult to completely avoid tracking errors due to occlusions and intersections of similar objects. These tracking errors make it impossible to establish correct identification. However, tracking for a short period when an object enters/exits FOVs[1] is reliable. We employ only this reliable information regarding entrance/exit into/from FOVs (i.e., IN/OUT data) for calibration of distributed cameras. Note that any object ID is not included in IN/OUT data. This means that our method can work even if multiple objects are wrongly identified within a FOV at the beginning and end points.

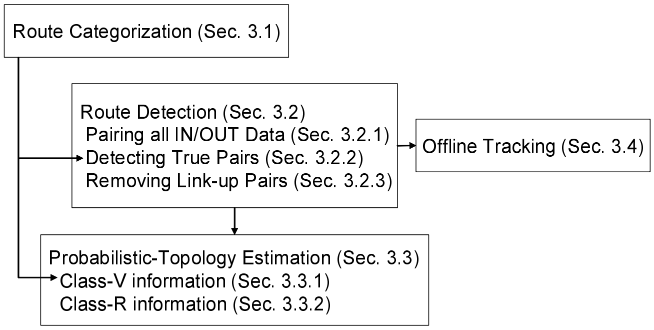

Route Categorization (Sec. 3.1)
Route Detection (Sec. 3.2)
Pairing all IN/OUT Data (Sec. 3.2.1)
Detecting True Pairs (Sec. 3.2.2)
Removing Link-up Pairs (Sec. 3.2.3)
Offline Tracking (Sec. 3.4)
Probabilistic-Topology Estimation (Sec. 3.3)
Class-V information (Sec. 3.3.1)
Class-R information (Sec. 3.3.2)

Fig. 3 Functional flow in Sec. 3.

The discussion in this section is summarized as follow:

1. No previous approach can deduce all of class-V and class-R information.
2. To acquire the complete probabilistic-topology of cameras, a sufficient amount of actual object trajectories should be analyzed.
3. It is difficult to deduce the complete probabilistic-topology of cameras from unreliable results of object tracking among isolated cameras. However, simple IN/OUT data can be acquired without errors.

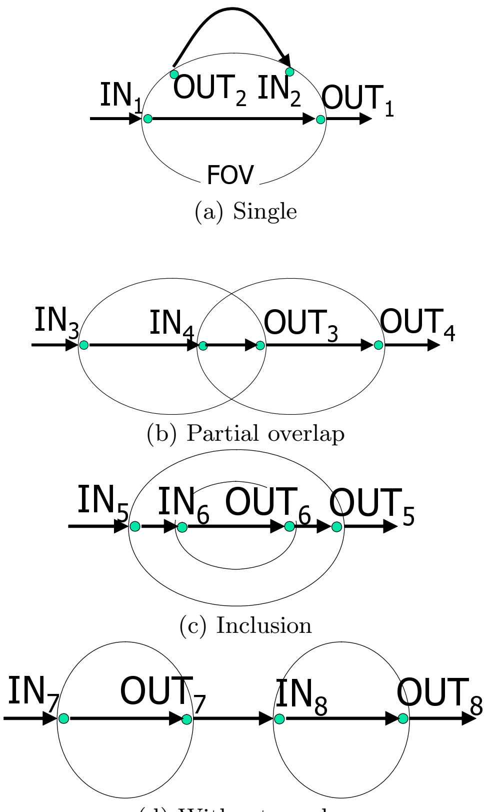
We, therefore, propose a method for estimating the probabilistic-topology of overlapping and isolated FOVs. The method has the following characteristics:

1. Can estimate all of V1, V2, R1, and R2 information.
2. Can analyze massive observation data (i.e., IN/OUT data) simultaneously for acquiring reliable probabilistic information.
3. Can work only with simple IN/OUT data for robust estimation.

3 Probabilistic-Topological Calibration

In our method, routes are first detected and then their probabilistic characteristics are inferred. In this section, first of all, Sec. 3.1 introduces a route categorization. This categorization is employed in the processes for estimating the probabilistic-topological information of cameras. Sec. 3.2 describes our route detection method consisting of three steps; (1) possible candidates for existing routes are found by pairing all IN/OUT data, (2) only true pairs, each of which is a pair of the beginning and end points, are detected by dividing all the pairs into several sets based on their spatio-temporal uniformities and classifying the sets into true and false pairs, and (3) pairs, which consist of concatenated consecutive routes, are then removed from the remaining pairs. A way for estimating the probabilistic-topological information based on the result of route detection is then described in Sec. 3.3. Sec. 3.4 shows that offline object tracking can be

[1] In our experiments, IN/OUT data is produced from a tracking result for two seconds when an object is newly/last observed in a FOV.



(a) Single

(b) Partial overlap

(c) Inclusion

(d) Without overlap

Fig. 4 IN/OUT positions in FOVs (denoted by ellipses).

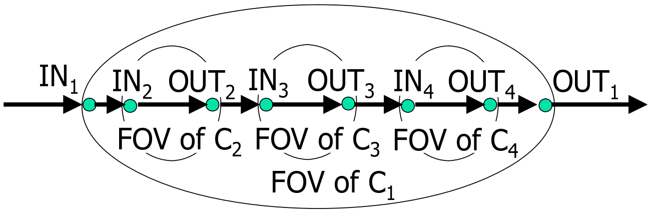


Fig. 5 Object trajectory through multiple included FOVs.

collaterally achieved by means of route detection. Fig. 3 illustrates a functional flow of the procedures described in this section.

3.1 Route Categorization for Estimating Geometric Configuration of Routes and FOVs

Figure 4 (a)∼(d) shows all possible examples of routes among overlapping and isolated FOVs. As mentioned before, two consecutive IN/OUT positions compose a route. Every route in more than two FOVs can be expressed by a combination of (a)∼(d).

From the observation of Fig. 4, all routes can be classified into five categories:

- $\overrightarrow{IN_i \cdot IN_j}$ and $\overrightarrow{OUT_p \cdot OUT_q}$, where i, j, p, and q denote any IDs, consist of two overlapping FOVs and are classified as the route type below:
  Type1 (Route through one of FOVs): A route included in one of two overlapping FOVs (e.g., $\overrightarrow{IN_3 \cdot IN_4}$, $\overrightarrow{OUT_3 \cdot OUT_4}$, $\overrightarrow{IN_5 \cdot IN_6}$, and $\overrightarrow{OUT_6 \cdot OUT_5}$ in Fig. 4).
- $\overrightarrow{IN_i \cdot OUT_j}$ exists in FOV(s) having IN_i and OUT_j and is classified into two categories as follows:
  Type2 (Route through a FOV): If a FOV observes both IN_i and OUT_j, a route goes across the FOV (e.g., $\overrightarrow{IN_1 \cdot OUT_1}$, $\overrightarrow{IN_6 \cdot OUT_6}$, $\overrightarrow{IN_7 \cdot OUT_7}$, and $\overrightarrow{IN_8 \cdot OUT_8}$ in Fig. 4).
  Type3 (Route through an overlapping area): If different FOVs observe IN_i and OUT_j, a route passes through their overlap (e.g., $\overrightarrow{IN_4 \cdot OUT_3}$ in Fig. 4).
- $\overrightarrow{OUT_i \cdot IN_j}$ exists outside FOV(s) having IN_i and OUT_j and is classified into two categories as follows:
  Type4 (Loop route): If a FOV observes both OUT_i and IN_j, a route begins and ends at the same FOV (e.g., $\overrightarrow{OUT_2 \cdot IN_2}$ in Fig. 4).
  Type5 (Route through an invisible area): If different FOVs observe IN_i and OUT_j, a route goes through the area that is invisible from them (e.g., $\overrightarrow{OUT_7 \cdot IN_8}$ in Fig. 4).

Based on the above categorization, class-V information about the FOVs which have a route between them can be obtained as follows:

V1: Pairs of FOVs with the routes categorized into Type1, Type3, and Type5 have routes between them while the routes categorized into Type2 and Type4 are in a single FOV.
V2: Pairs of FOVs with the routes categorized into Type1 and Type3 overlap. All other pairs of FOVs do not have any overlapping area.

For example, since all $\overrightarrow{IN \cdot IN}$ and $\overrightarrow{OUT \cdot OUT}$ routes are classified into Type1, a pair of FOVs with these routes have a route and an overlapping area. Accordingly, route detection is a key issue in estimating the class-V information of distributed cameras.

Note that the topology of FOVs, which do not have a route between them but have an overlapping area and an object trajectory (e.g., FOVs of $C1$ and $C3$ as illustrated in Fig. 5), should be also obtained for estimating the total topology in the system. These exceptional relationships can be obtained by categorizing *link-up routes*, each of which consists of successive routes. Since the link-up routes are also obtained using our method (described in Sec. 3.2.3), the total topology can be deduced.

In addition to estimating class-V information, the route categorization is employed also to remove false-detection routes (described in Process 5 of Sec. 3.2.2).

As described at the end of Sec. 2, all processes, including route detection, in our method are achieved by using a large amount of IN/OUT data. As a result of route detection, all the IN/OUT data are classified into each route. The classified data are then employed for estimating class-R information (described in Sec. 3.3.2).

In what follows, a practical way for probabilistic-topological calibration will be described.

3.2 Route Detection by Analyzing the Uniformities of IN/OUT data

3.2.1 Pairing All IN/OUT Data for Route Nomination

The beginning and end points of a route correspond to successive IN/OUT positions of a moving object. However, temporally successive IN/OUT data observed in the total system may be the data of different objects and do not always compose a route in an environment with multiple moving objects. That is, our problem is how to find the pairs of IN/OUT data of the same object, which are the beginning and end points of the routes, from a number of observed IN/OUT data. A pair of IN/OUT data that composes a route is called a *True Correspondence* (TC). Each of all other pairs is called a *False Correspondence* (FC).

For this problem, we focus on the following two uniformities in massive amounts of observed data:

Spatial uniformity (beginning and end points): In general, moving objects go through regular routes a number of times. In each route, therefore, a number of IN/OUT data must be observed around both its beginning and end points.

Temporal uniformity (transit time): Each of the certain kinds of objects (e.g., people walking, people jogging, and cars) needs almost the same transit time through a route, except situations such as stopping on the route. This fact results in the following prospect: If each IN/OUT data is paired with all other IN/OUT data tentatively and the time interval between each pair is calculated, time intervals between the pairs of existent beginning and end points are observed extensively. This means that routes exist between the paired IN/OUT data and it takes one of the extensively observed time intervals to go through each route.

According to this criterion, if objects pass through a route at different speeds, multiple routes with different transit times are detected. In our definition of a route, since the difference of transit times is regarded as a crucial factor (i.e., R2 information) for object tracking, these routes must be detected separately.

In [18], route detection is achieved based on peak detection using independent (sequential) analyses of the above two uniformities:

1. Observed IN/OUT data are classified into groups based on proximity of their positions in each image. Each group corresponds to a beginning/end point.
2. Time intervals between IN/OUT data are voted in each pair of the groups. In each voting result, peaks are observed. If a peak of the voted time intervals exists, there is a route between the positions of the groups.

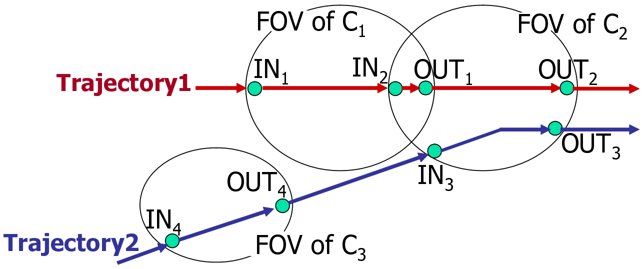


Fig. 6 FOVs and object trajectories in a scene.

In this method, however, (1) observed positions are classified in each image without linking the positions between FOVs, (2) it is very difficult to adequately classify observed IN/OUT positions because the number of classes is unknown, and (3) IN/OUT positions of different adjacent routes may be wrongly classified into one group; the class-R information estimated from this error classification is unreliable. On the other hand, our method realizes robust route detection by classification which simultaneously takes into account two uniformities.

For route detection based on the above uniformities, all tentative pairs of IN/OUT data have to be collected. With reference to Fig. 6, a procedure for collecting tentative pairs of IN/OUT data is described. Figure 6 illustrates an example with three cameras and two object trajectories. First of all, massive IN/OUT data of a number of moving objects is detected. Each IN/OUT data is then paired with all other IN/OUT data observed before its detection. The former and latter IN/OUT data are tentatively regarded as the end and beginning points of a route, respectively. However, pairs of IN/OUT data with sufficiently long intervals do not have to be collected. In the example shown in Fig. 6, since $\overrightarrow{OUT_4 \cdot IN_3}$ is the longest route, all time intervals less than the transit time of $\overrightarrow{OUT_4 \cdot IN_3}$ should be collected as tentative pairs[2]. The collected pairs of IN/OUT data are divided into several sets depending on the FOVs that observe the paired IN/OUT data. That is, if the beginning and end points of a pair are observed in cameras C^B and C^E, respectively, this pair is voted into a set of tentative pairs, $\boldsymbol{S}^{B,E}$. For N cameras, ${}_NP_2 + N$ vote sets are prepared, where ${}_iP_j$ denotes a permutation. For example, two cameras (i.e., C_1 and C_2) require four vote sets, $\boldsymbol{S}^{1,2}$, $\boldsymbol{S}^{2,1}$, $\boldsymbol{S}^{1,1}$, $\boldsymbol{S}^{2,2}$.

[2] This time threshold was determined manually in our experiments. Strictly speaking, this threshold should be determined depending on observation errors of IN/OUT data.

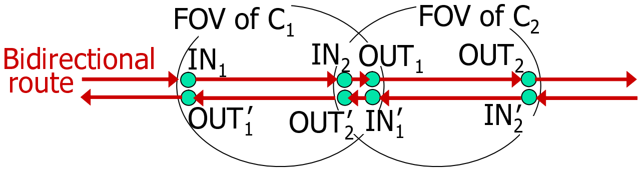
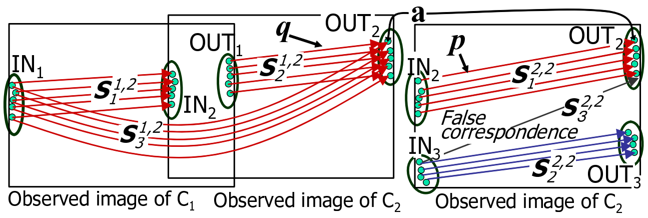


Fig. 7 Detecting an inverse route.



(a) Pairs voted into $S^{1,2}$ (b) Pairs voted into $S^{2,2}$

Fig. 8 IN/OUT data pairs displayed in two observed images: Each dot and ellipse including dots denotes an observed IN/OUT position and a beginning/end point, respectively.

In the example shown in Fig. 6, both trajectories 1 and 2 are unidirectional. If objects move in the reverse direction, every route in the reverse trajectory is detected as the route consisting of the inverted beginning and end points as follows:

- In Fig. 7, IN_1, IN_2, OUT_1, and OUT_2 in the right-pointing routes are changed to OUT'_1, OUT'_2, IN'_1, and IN'_2 in the left-pointing routes, respectively.
- Route $\overrightarrow{IN_2 \cdot OUT_1}$ of the right-pointing trajectory is detected from IN/OUT data pairs voted into $\boldsymbol{S}^{2,1}$. On the other hand, the inverse route of $\overrightarrow{IN_2 \cdot OUT_1}$ (i.e., $\overrightarrow{IN'_1 \cdot OUT'_2}$) is detected from IN/OUT data pairs voted into $\boldsymbol{S}^{1,2}$. That is, the inverse route is detected from the vote set whose beginning and end points are reversed from the vote set of the forward route.

3.2.2 Detecting TCs based on the Uniformities and Majority of IN/OUT Data Pairs

Arrows in Fig. 8 indicate several examples of tentative pairs obtained when observing the environment shown in Fig. 6. Figures (a) and (b) illustrate the pairs voted in $\boldsymbol{S}^{1,2}$ and $\boldsymbol{S}^{2,2}$, respectively. Note that

- for simplification, pairs voted into $\boldsymbol{S}^{2,1}$ and $\boldsymbol{S}^{1,1}$ (e.g., $\overrightarrow{IN_2 \cdot OUT_1}$ and $\overrightarrow{IN_1 \cdot OUT_1}$) are not displayed, and
- just a few IN/OUT data are selectively displayed in each beginning/end point although in fact massive IN/OUT data of a number of moving objects is detected.

FCs (e.g., '*False correspondence*' in Fig. 8 (b)) are included in each set $\boldsymbol{S}^{i,j}$ where i and j denote the IDs of any FOV. The goal of this section is to extract only TCs and divide them into each route. The extraction and division are achieved by classifying 5D vectors, each of which expresses one of the voted pairs of IN/OUT data:

Process 1: Let $\{\boldsymbol{V}_1, \cdots, \boldsymbol{V}_{N^{i,j}}\}$ be a set of vectors, where $\boldsymbol{V}_i = (x_i^B, y_i^B, x_i^E, y_i^E, t_i)$ denote a 5D vector comprising the image coordinates of the beginning and end points and the transit time between them, and $N^{i,j}$ denote the total number of the ballots in $\boldsymbol{S}^{i,j}$.

Process 2: Elements in each vector are normalized so that all of their ranges are from 0 to 1.

Process 3: $\{\boldsymbol{V}_1, \cdots, \boldsymbol{V}_{N^{i,j}}\}$ are divided into several subsets based on LBG algorithm[19]: In our method, the division is continued until the mean of distances between all vectors in each subset and the codevector (i.e., mean vector) of the subset is less than a predefined threshold of the mean distance. The threshold is manually set small enough to be able to discriminate between the vectors corresponding to FCs and similar TCs. After the division, the norm of the difference between each vector and the codevector is calculated. Vectors having the norm that is more/less than $(\mu_v +/- 2.5\sigma_v)$, where μ_v and σ_v denote the mean and the standard deviation of all the norms, respectively, are eliminated. As a result, all remaining vectors in each subset are classified into the sets of TCs or FCs. Note that the important requirement is removing all FCs. Therefore, even if the vectors of a route that should be in the same subset are divided into different subsets, this has no negative effect on employing the acquired information for tracking. This is because all of the divided subsets corresponding to the same route are integrated to estimate class V and R information of each route (described in Sec. 3.3).

Process 4: The processes mentioned above generate several subsets from a set, $\boldsymbol{S}^{i,j}$. In the example shown in Fig. 8, '$\boldsymbol{S}_1^{1,2}$, $\boldsymbol{S}_2^{1,2}$, and $\boldsymbol{S}_3^{1,2}$' and '$\boldsymbol{S}_1^{2,2}$, $\boldsymbol{S}_2^{2,2}$, and $\boldsymbol{S}_3^{2,2}$' are generated from $\boldsymbol{S}^{1,2}$ and $\boldsymbol{S}^{2,2}$, respectively. The number of the ballots is then calculated in each subset. If the number is less than $(\mu_b - 2.5\sigma_b)$, where μ_b and σ_b denote the mean and the standard deviation of the numbers of the ballots, respectively, the subset with few ballots is considered to be the set of FCs and it is eliminated.

Process 5: Several remaining FCs can be removed based on route categorization described in Sec. 3.1. The beginning and end points of a Type1 route must be observed by different FOVs. Therefore, Type1 routes, having the beginning and end points in the same FOV, are removed.

The above processes are performed for every set of the tentative pairs (i.e., every $\boldsymbol{S}^{i,j}$). After all the processes, all subsets with FCs are removed and only subsets with TCs remain. These subsets are called *False Corresponded Routes* (FCRs) and *True Corresponded Routes* (TCRs), respectively. The TCRs between the cameras C^B and C^E are denoted by $\boldsymbol{R}^{B,E} = \left\{R_i^{B,E} | i \in \{1, \cdots, N^{B,E}\}\right\}$, where $R_i^{B,E}$ denotes one of the detected TCRs between C^B and C^E and $N^{B,E}$ denotes the number of the routes between C^B and C^E. By means of the above mentioned TCR detection processes, the pairs of IN/OUT data are selectively grouped into the subsets (i.e., all the sets remaining after Process 5), each of which shows the information with regard to one of the TCRs (i.e., $R_i^{B,E}$ in $\boldsymbol{R}^{B,E}$). Each subset of the pairs grouped into $R_i^{B,E}$ is expressed by $\boldsymbol{TCS}_i^{B,E}$.

3.2.3 Removing Link-up Routes

The TCRs, $\boldsymbol{R}^{B,E}$, consist not only of routes but also link-up routes introduced in Sec. 3.1. In the example illustrated in Fig. 8, $\overrightarrow{IN_1 \cdot OUT_2}$, which is detected as a TCR, is a link-up route formed by concatenating $\overrightarrow{IN_1 \cdot IN_2}$, $\overrightarrow{IN_2 \cdot OUT_1}$ (not displayed in Fig. 8 but detected from $\boldsymbol{S}^{2,1}$) and $\overrightarrow{OUT_1 \cdot OUT_2}$. All link-up routes can be removed based on cooccurence of the detected TCRs:

Process 1: Let C^{B_i} and C^{B_j} denote any pair of cameras, both of which have routes towards camera C^E, including $C^{B_i} = C^E$ and/or $C^{B_j} = C^E$.

Process 2: Let $\boldsymbol{TCS}_p^{B_i,E}$ have a voted pair with the end IN/OUT data, which is also the end IN/OUT data of a pair included in $\boldsymbol{TCS}_q^{B_j,E}$. In Fig. 8, two voted pairs p and q have the same OUT data (indicated by a) at their end points. If the transit time of p is longer than that of q, $R_p^{B_i,E}$ might be a link-up route consisting of $R_q^{B_j,E}$ and other route(s).

Process 3: To verify whether $R_p^{B_i,E}$ is really a link-up route involving $R_q^{B_i,E}$, voted pairs in $\boldsymbol{TCS}_p^{B_i,E}$ and $\boldsymbol{TCS}_q^{B_j,E}$, that have the same relationship with that of p and q, are checked. First of all, the number of the voted pairs in $\boldsymbol{TCS}_p^{B_i,E}$ and $\boldsymbol{TCS}_q^{B_j,E}$, which have the same IN/OUT data at their end points, is counted. The rate of the counted number in the smaller one of the two sets (i.e., $\boldsymbol{TCS}_p^{B_i,E}$ or $\boldsymbol{TCS}_q^{B_j,E}$) is then calculated. If the rate is over a predefined threshold[3], $R_p^{B_i,E}$ is regarded as a link-up route and removed from $\boldsymbol{R}^{B_i,E}$.

These processes are performed such that every camera is regarded as camera C^E in the above processes.

Finally, all routes are detected by the procedure described in Sec. 3.2.

3.3 Estimating the Probabilistic-Topology

3.3.1 Acquiring Class-V Information based on Categorizing the TCRs

With the processes described in Sec. 3.2, all routes and link-up routes (i.e., TCRs) among the FOVs of all cameras are detected. As described in Sec. 3.1, therefore, V1 and V2 can be estimated by comparing the types of the detected TCRs with the five categories.

3.3.2 Acquiring Class-R Information from the Categorized Routes

In each route R_i, where i denotes the ID of routes, the mean (x, y) coordinates of the beginning and end points (denoted by $\boldsymbol{\mu}_i^B, \boldsymbol{\mu}_i^E$) and their covariance matrices (denoted by $\boldsymbol{\Sigma}_i^B, \boldsymbol{\Sigma}_i^E$) are calculated. With the mean coordinates, the covariance matrices, and the number of pairs voted into R_i (denoted by Nv_i), the probability that an object was lastly observed at $\boldsymbol{P}^B(C^B)$ before it is newly detected at $\boldsymbol{P}^E(C^E)$ (this probability is denoted by $P_{R1}(\boldsymbol{P}^B(C^B), \boldsymbol{P}^E(C^E))$) can be estimated as follows:

1. Let $\boldsymbol{R}^{\cdot,E} = R_1^{\cdot,E}, \cdots, R_{N^{\cdot,E}}^{\cdot,E}$ denote all routes with the end point in the FOV of C^E, where $N^{\cdot,E}$ is the number of these routes. The probability that the end point of $R_i^{\cdot,E}$ is $\boldsymbol{P}^E(C^E)$ at which an object is newly detected is calculated by substituting $\boldsymbol{\mu}_i^E$ and $\boldsymbol{\Sigma}_i^E$ of $R_i^{\cdot,E}$ and $\boldsymbol{P}^E(C^E)$ for the equation below:
$$Q(\boldsymbol{P}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left((\boldsymbol{P}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{P}-\boldsymbol{\mu})\right) \qquad (1)$$
Let S be the total sum of $Q(\boldsymbol{P}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ multiplied by the number of pairs, namely,
$$S = \sum_{i=1}^{N^{\cdot,E}} Q(\boldsymbol{P}^E(C^E); \boldsymbol{\mu}_i^E, \boldsymbol{\Sigma}_i^E) Nv_i.$$
Then, $\left(Q(\boldsymbol{P}^E(C^E); \boldsymbol{\mu}_i^E, \boldsymbol{\Sigma}_i^E) Nv_i\right) / S$ can be considered to be the probability that the end point of route $R_i^{\cdot,E}$ is regarded as the position of new detection at $\boldsymbol{P}^E(C^E)$ (denoted by $P(\boldsymbol{P}^E(C^E), R_i^{\cdot,E})$).
2. Let $\boldsymbol{R}^{B,E} = R_1^{B,E}, \cdots, R_{N^{B,E}}^{B,E}$ denote a subset of $\boldsymbol{R}^{\cdot,E}$, which has the beginning point at the FOV of C^B. In a similar way to the above process, the probability that the beginning point of $R_j^{B,E}$ is $\boldsymbol{P}^B(C^B)$ (this probability is denoted by $Q(\boldsymbol{P}^B(C^B); \boldsymbol{\mu}_j^B, \boldsymbol{\Sigma}_j^B)$) is calculated by equation (1).
3. The total sum of Q multiplied by P of the same route is the probability that means R1:
$$P_{R1}(\boldsymbol{P}^B(C^B), \boldsymbol{P}^E(C^E)) = \sum_{x=1}^{N^{B,E}} P(\boldsymbol{P}^E(C^E), R_x^{\cdot,E}) Q(\boldsymbol{P}^B(C^B); \boldsymbol{\mu}_j^B, \boldsymbol{\Sigma}_j^B)$$

[3] In our extensive experiments, the rate in a link-up route and its segment is over 90% while that in two independent routes is less than 10%. The method for removing link-up routes is, therefore, very robust.

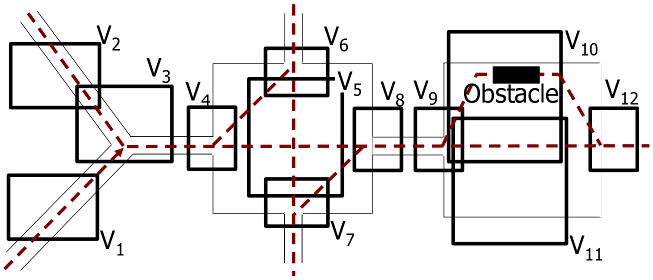
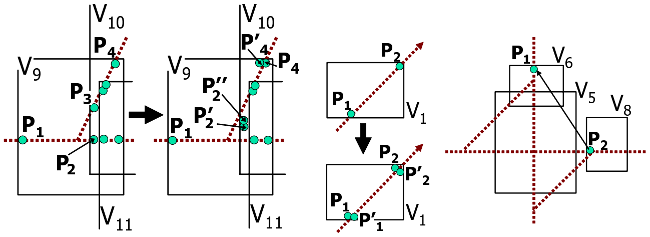


Fig. 9 View from above of the scene for simulation experiments.



(a) Experiment 1 (b)
Experiment 2 (c) Experiment 3

Fig. 10 Examples of the rise and fall of detected routes.

With this procedure, R1 regarding every pair of the beginning and end points can be estimated.

R2 can be estimated from the mean and variance of the transit times in the IN/OUT pairs voted in $\boldsymbol{R}_i$ as follows. $Q(T;\mu_i^{B,E},\sigma_i^{B,E})$ calculated by equation (1) indicates the probability that an object spends T for crossing route R_i. That is, $Q(T;\mu_i^{B,E},\sigma_i^{B,E})$ means R2.

When a tracking system uses class-R information, two probabilities, P_{R1} and P_{R2}, are multiplied. The product is regarded as the probability that an object leaves the FOV of C^B from $\boldsymbol{P}^B(C^B)$ and directly reaches the FOV of C^E at $\boldsymbol{P}^E$ and its transit time is T.

3.4 Offline Object Tracking from IN/OUT Data Pairs in TCRs

The pairs of IN/OUT data are voted in Sec. 3.2. Each pair grouped into a set of TCs is considered to comprise successive IN/OUT data of the same object, which make a route. This means that the voted pairs in a set of TCs provide the results of object identification between and within FOV(s). Accordingly, offline tracking can be achieved in the process of route detection. Note that

- false-positive identifications are detected when two or more objects pass through a route side by side and go through the near beginning/end points of different routes almost simultaneously, and
- correct identifications cannot be established when an object passes through an irregular trajectory at an irregular interval.

4 Experimental Results

4.1 Simulation Experiments

We conducted simulation experiments to confirm the robustness of the proposed method against three undesired phenomena. An experimental scene is illustrated in Fig. 9. Each rectangle V_i and dotted line indicates a FOV (i.e., its area observed by 640×480 pixel) and an object trajectory, respectively. While only the trajectory from V_1 to V_3 is a one-way path, objects move bidirectionally on all other paths. In this environment, the number of routes is 78 (37 bidirectional routes and 4 unidirectional routes) if all observation conditions are ideal such that objects move just along the dotted line and their trajectories are observed without any image noise.

Experiment 1: Fluctuate the coordinates of IN/OUT positions: This causes the fluctuations of the beginning and end points of each route and destroys their uniformity.

Experiment 2: Fluctuate the velocities of objects: This causes the fluctuations of the transit time of each route and violates its uniformity.

Experiment 3: Increase the number of moving objects simultaneously in the environment: This increases FCs.

The results of the experiments are presented in Table 1 with the following data:

number of routes: '+' and '−' indicate the rise and fall of the number of detected routes from the ideal number, respectively. Underlined values indicate the number of remaining FCRs.

successful tracking: Success rates (%) of offline tracking are shown.

The upper rows in tables (1), (2), and (3) denote the variance of observed (x, y) coordinates, the variance of object velocities, and the mean value of objects observed in each image at each moment, respectively. In order to verify only the influence of each varying parameter, each experiment was conducted under the conditions below except the varying parameter: 'the variance of coordinates is 1', 'the variance of object speed is 1', and 'one object is in a FOV at each moment'. Note that the success rate of tracking is shown only in the results of experiment 3; in experiments 1 and 2, all objects could be easily identified because only one object was observed at each moment.

With a sufficient number of IN/OUT data (about 10 thousands × the number of simultaneously observed objects), our proposed method was executed. Figure 10 shows examples of the results:

Fall in experiment 1 (Fig. (1)): IN data voted into the end points of two ideal routes from V_9 to V_{10} (i.e.,

Table 1 Changes in the number of routes.

variance of coordinates	0	2	4	8	16	32
number of routes	±0	±0	±0	+2,−2	+2,−11	+2,−12

(1) Fluctuation of detected positions

variance of velocities	0	2	4	8	16	32
number of routes	±0	±0	±0	+1	+5	+11

(2) Fluctuation of object velocity

number of objects	1	2	4	8	16	32
number of routes	±0	±0	±0	±0	$\underline{+1}$	$\underline{+10}$
successful tracking (%)	100	100	100	99	94	86

(3) Frequency of object detection

$\overrightarrow{P_1 \cdot P_2}$ and $\overrightarrow{P_1 \cdot P_3}$ in the left figure of Fig. 10 (1)) were observed with overlap around P_2 and P_3. These two route then merged into $\overrightarrow{P_1 \cdot P_2'}$ in the right-hand figure.

Rise in experiment 1 (Fig. (1)): Several observed trajectories from P_3 to P_4 did not pass through V_{11} due to fluctuation. As a result, $\overrightarrow{P_2'' \cdot P_4'}$ in the right-hand figure were newly generated.

Rise in experiment 2 (Fig. (2)): $\overrightarrow{IN_{P_1'} \cdot OUT_{P_2'}}$ in the lower figure of Fig. (2) derived from $\overrightarrow{IN_{P_1} \cdot OUT_{P_2}}$ due to the fluctuation in the time intervals between the beginning and end points.

Rise in experiment 3 (Fig. (c)): Since a number of IN/OUT data was observed for a short time, FCR $\overrightarrow{P_1 \cdot P_2}$, whose beginning and end points were both IN data, remained in the result of route detection. Due to this error, the overlap between V_6 and V_8 was detected by mistake.

From these results, we can see the following observations regarding the characteristics of the proposed method:

Experiment 1: The fluctuation in the coordinates of IN/OUT positions yields both the rise and fall of detected routes.

Experiment 2: The fluctuation in the velocities of objects yields only the rise of detected routes.

Experiment 3: The increase of objects yields the rise of FCRs. The results are robust as long as the number of objects does not increase drastically.

While the rise and fall of detected routes change depending on a threshold of LBG for classifying 5D vectors, the characteristics mentioned above are invariable. All the rise and fall of routes except those of FCRs are induced by the rise and fall of the grouped IN/OUT data due to fluctuations of IN/OUT positions. These rises and falls of the detected routes, as well as those depending on thresholding in LBG, do not have an adverse effect on tracking with the estimated topology of FOVs. The FCRs, on the other hand, violate the topology of FOVs because narrowing down the possible trajectories of target objects may fail.

It can be also confirmed that offline tracking based on route detection is very robust except in the case of a large number of objects.

4.2 Experiments with Real Distributed Cameras

We conducted experiments using a real system with 12 cameras ($C_1 \sim C_{12}$). The cameras captured 320x240 pixel RGB images at 1 sec intervals. The experimental environment and the image examples are shown in Fig. 11. While the images were observed during daytime hours, many office workers and students engaged in everyday activities in the environment. All the people walked in the environment at various speeds. At each moment when at least one object was observed in one of 12 cameras, 2.7 persons on average were observed simultaneously in the environment.

Object detection was implemented with a simplified version of [16]. All detected pixels were then grouped into each object region based on connectivity of the detected pixels. Each detected object region is identified with object regions in the previous frame based on proximity. The numbers of IN/OUT data observed by $C_1 \sim C_{12}$ were 7238, 7910, 11789, 13782, 12376, 6792, 7067, 7856, 8178, 12574, 12456, and 12786, respectively. Some of the detection and tracking results included errors, for example, two or more persons were regarded as one person. However, these errors had less harmful impact on our method because IN/OUT data is free from object IDs.

From all the above IN/OUT data, 130 routes (59 bidirectional and 12 unidirectional routes). The mean of pairing IN/OUT data voted into each TCR was 2139. Examples of the detected routes are shown in Fig. 12. Arrows and ellipses indicate the detected routes and the distributions of their beginning and end points, respectively. The width of the arrow is in proportion to the number of pairing IN/OUT data voted into the arrow. Although the beginning and end points are generated for each route, their ellipses, which are close to each other, are merged in the figure for viewability.

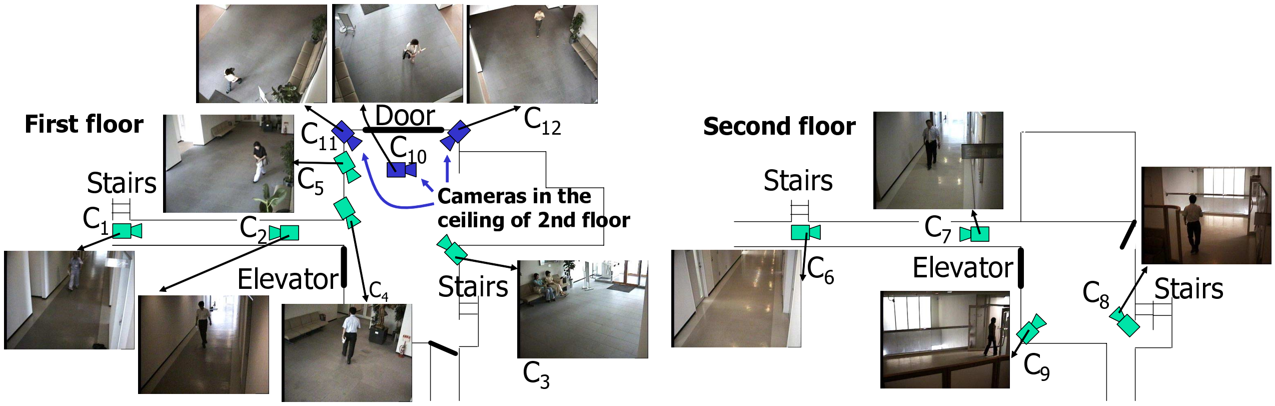


Fig. 11 View from above of the scene and image examples (Left: 1st floor, Right: 2nd floor).

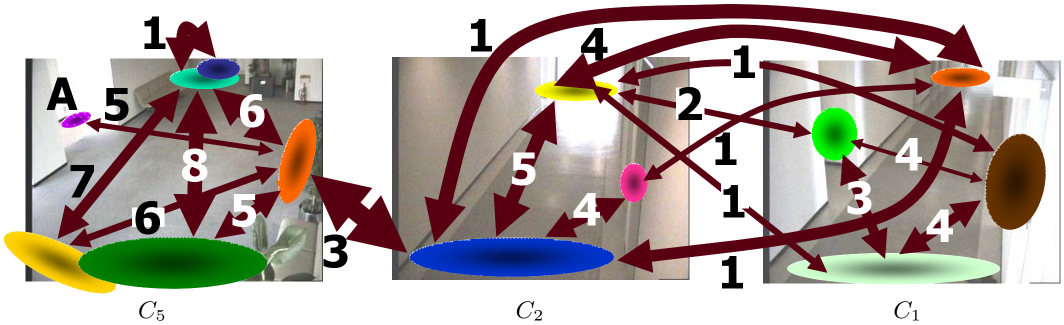


Fig. 12 Examples of detected routes.

From information of the detected routes, the following pairs of FOVs are considered to have routes and overlapping areas:

With routes (V1): C_1-C_2, C_1-C_2, C_2-C_4, 'all possible pairs of C_3, C_4, C_5, C_{10}, C_{11}, and C_{12}', C_6-C_7, C_7-C_8, C_7-C_9 and C_8-C_9.

With overlapping areas (V2): C_1-C_2, 'all possible pairs of C_3, C_4, C_5, C_{10}, C_{11}, and C_{12}', C_6-C_7, and C_8-C_9.

Some examples of the estimated class-R information are as follows:

R1: The mean and variance of point 'A' in C_5 of Fig. 12 were (56.1, 71.8) and (4.1, 2.2), respectively.

R2: An integer with each arrow refers to the mean of transit times (seconds) in each route.

Finally, the effectiveness of offline tracking was verified. 300 object transits between FOVs were selected manually from the observed sequences. By comparing them with the results of offline tracking, two incorrect identifications were found. These were caused by errors in detecting object regions from observed images. For example, one of the errors was caused when an object moved through the upper part of C_5 of Fig. 12. Since its transit time was very short and the capturing interval was long (i.e., 1 sec.), object detection and tracking in this route were liable to fail.

5 Concluding Remarks

We proposed the probabilistic-topological calibration of widely distributed cameras. With the proposed method, (1) the relationships among the FOVs of the cameras and (2) the probabilistic representations of the beginning and end points and the time intervals between the points can be estimated.

We are studying the following improvements:

- Automatic thresholding in dividing IN/OUT data pairs for more robust route detection.
- FCRs in the same FOV can be removed by checking the results of continuous tracking within the FOV.
- In the experiments shown in this paper, every trajectory was restricted in a narrow area (e.g., a passage way) and could be simply expressed as a line. If each FOV is wider, on the other hand, the routes of freely moving objects in a large open area have large variations of their beginning and end points. A new solution may be required for this situation.

Acknowledgements This work is supported by the PRESTO program of Japan Science and Technology Agency (JST).

References

1. Q. Cai and J. K. Aggarwal, "Tracking Human Motion in Structured Environments using a distributed camera system," *PAMI*, Vol.21, No.11, pp.1241–1247, 1999.
2. A. Mittal and L. S. Davis, "M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene," *IJCV*, Vol.51, No.3, pp.189–203, 2003.
3. 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," *IVC*, Vol.22, No.7, pp.551–561, 2004.
4. N. Ukita and T. Matsuyama, "Real-time Cooperative Multi-target Tracking by Communicating Active Vision Agents," *CVIU*, Vol.97, No.2, pp.137–179, 2005.
5. R. Collins, O. Amidi, and T. Kanade, "An Active Camera System for Acquiring Multi-View Video," in *Proc. of ICIP*, pp.517-520, 2002.
6. A. Azarbayejani and A. Pentland, "Real-time Self-calibrating Stereo Person Tracking Using 3-D Shape Estimation from Blob Features," in *Proc. of ICPR*, Vol.3, pp.627–632, 1996.
7. X. Chen, J. Davis, and P. Slusallek, "Wide Area Camera Calibration Using Virtual Calibration Objects," in *Proc. of CVPR*, Vol.2, pp.5200-527, 2000.
8. L. Lee, R. Romano, G. Stein, "Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame," *PAMI*, Vol.22, No.8, pp.758–767, 2000.
9. R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for Cooperative Multisensor Surveillance," in . *Proc. of the IEEE*, Vol.89, No.10, pp.1456–1477, 2001.
10. H. Pasula, S. Russell, M. Ostland, and Y. Ritov, "Tracking Many Objects with Many Sensors," in *Proc. of IJCAI*, pp.1160–1171, 1999.
11. V. Kettnaker and R. Zabih, "Bayesian Multi-camera Surveillance," in *Proc. of CVPR*, pp.253–259, 1999.
12. J. E. Boyd, J. Meloche, and Y. Vardi, "Statistical Tracking in Video Traffic Surveillance," in *Proc. of ICCV*, pp.163–168, 1999.
13. O. Javed, Z. Rasheed, O. Alatas, M. Shah, "KNIGHT: A Real Time Surveillance System for Multiple and Non-overlapping Cameras," in *Proc. of ICME*, pp.649–652, 2003.
14. J. Davis and X. Chen, "Calibrating Pan-Tilt Cameras in Wide-area Surveillance Networks," in *Proc. of ICCV2003*, pp.144–149, 2003.
15. O. Javed, Z. Rasheed, K. Shafique, and M. Shah, "Tracking Across Multiple Cameras With Disjoint Views," in *Proc. of ICCV*, pp.952–957, 2003.
16. K. Toyama, J. Krumm, B. Brumitt and B. Meyers: "WallFlower: Principles and Practice of Background Maintenance", in *Proc. of ICCV*, pp.255–261, 1999.
17. J. Vermaak, A. Doucet, and P. Perez: "Maintaining Multi-Modality through Mixture Tracking," in *Proc. of ICCV*, pp.1110–1116, 2003.
18. D. Makris, T. Ellis, and J. Black, "Bridging the Gaps between Cameras," in *Proc. of CVPR*, Vol.2, pp.205–210, 2004.
19. Y. Linde, A. Buzo, and R. M. Gray: "An Algorithm for Vector Quantizer Design", *IEEE Trans. on Communications*, Vol.28, No.1, pp.84–95, 1980.

Norimichi Ukita received the B.E. and M.E. degrees in information engineering from Okayama University, Japan, in 1996 and 1998, respectively, and the Ph.D degree in Informatics from Kyoto University, Japan, in 2001. He is currently an assistant professor at the graduate school of information science, Nara Institute of Science and Technology (NAIST), Japan. He is also a research scientist of Precursory Research for Embryonic Science and Technology, Japan Science and Technology Agency (JST). His main research interests are object detection and tracking, active camera control, and cooperative distributed vision. He has received the best paper award from the IEICE in 1999.