Efficient Topology Calibration and Object Tracking with Distributed Pan Tilt Cameras

Norimichi Ukita, Kunihito Terashita, and Masatsugu Kidode
Nara Institute of Science and Technology
Email: ukita@ieee.org

***Abstract*—We propose a method for calibrating the topology of distributed pan tilt cameras (i.e., the structure of routes among FOVs) and its probabilistic model, which is useful for multi-object tracking in a wide area. To observe objects as long and many as possible, pan tilt control is an important issue in automatic calibration as well as in tracking. If only one object is observed by a camera and its neighboring cameras, the camera should point towards this object both in the calibration and tracking periods. However, if there are multiple objects, in the calibration period, the camera should be controlled towards an object that goes through an unreliable route in which a sufficient number of object detection results have not been observed. This control allows us to efficiently establish the reliable topology model. After the reliable topology model is established, on the other hand, the camera should be directed towards the route with the biggest possibility of object observation. We therefore propose a camera control framework based on the mixture of the reliability of the estimated routes and the probability of object observation. This framework is applicable both to camera calibration and object tracking by adjusting weight variables. Experiments demonstrate the efficiency of our camera control scheme for establishing the camera topology model and tracking objects as long as possible.**

I. Introduction

Object tracking is one of the fundamental problems in recent computer vision researches. In particular, tracking among widely distributed cameras has become a popular research issue. If fields of view (FOVs) of cameras are overlapped and their extrinsic parameters are known, object tracking can be simplified by analyzing consistencies in 3D positions of the observed objects. (e.g., using fixed cameras[1], [2], [3], omnidirectional cameras[4], and active cameras[5], [6], [7]). As is well known, many methods for calibrating the extrinsic parameters of cameras have been proposed; for example, calibration with observation results of moving objects (for synchronized[8] and asynchronized[9] cameras) and a method for improving the initial results[10]. For all of these methods, the cameras must be positioned so that an object moves through the FOVs of the cameras without going outside the FOVs. The assumption that the FOVs are overlapped makes it practically impossible to employ a number of cameras for observing wider areas.

Accordingly, camera configuration without overlapping FOVs, namely with blind spots, is necessary. In this challenging problem, the topology of FOVs and its probabilistic information (e.g., presence of a route, transit times, and object transit probabilities between FOVs) can improve object identification (see [11], [12], for example). Previously, the camera topology was given manually[11], [12]. Novel algorithms, however, provide us the camera topology automatically. Automatic calibration is desired because as the observation area grows and the number of cameras increases, the topology becomes drastically more complex. In [13], the camera topology is estimated from object tracking results among and within FOVs. However, object tracking between isolated FOVs only based on image cues (e.g., face/object recognition) is very difficult. For reliable identification among the FOVs, in [14], it is assumed that only one object moves in an environment. Otherwise, the trajectory of a moving landmark that is easily tracked (e.g., LED)[15] can be obtained even if other objects exist, or robust identification between isolated FOVs can be achieved also by employing information only of easy-identifiable objects as proposed in [16]. With these approaches, however, object transit probabilities between FOVs cannot be estimated because it can be acquired only from a number of real object trajectories. On the other hand, [17] and [18] acquire the probabilistic-topology of FOVs from a large amount of real object data, which is represented only by first and last detection results in each image; no tracking among isolated cameras is needed.

One of the next steps in calibration of widely distributed cameras is how to efficiently employ pan tilt cameras instead of fixed cameras. As proposed in [19], object tracking with distributed pan tilt cameras is crucial for efficient high-resolution observation. As similarly to tracking with fixed cameras, the camera topology of the pan tilt cameras is useful for object identification. In this paper, we 1) show that probabilistic topological calibration of pan tilt cameras can be achieved by the same way as that of fixed cameras and 2) propose how to efficiently obtain the calibration result. We also show that the camera control framework for efficient calibration can be used also for efficient object tracking with pan tilt cameras.

II. Probabilistic Topological Camera Calibration

As defined in [18], our topology model of cameras is defined by routes among the FOVs of the cameras. For object tracking through blind spots, the following object information observed in entrance and exit in FOVs is important:

IN: An event such that ‘an object is newly detected in the FOV’ is called *IN*.

OUT: An event such that ‘an object leaves the FOV at the next capturing timing’ is called *OUT*.

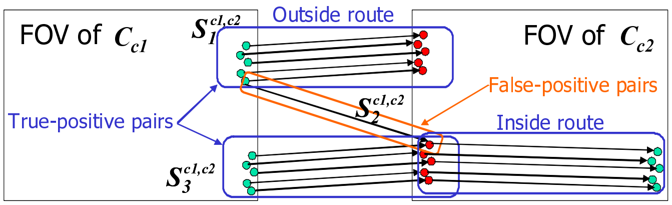
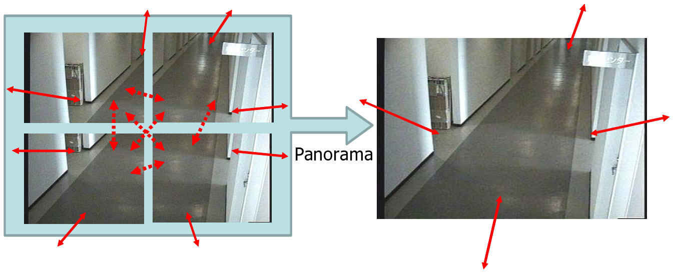


Fig. 1. True-positive route detection by vector quantization and thresholding. Small circles and lines between them indicate IN/OUT position and their pair (the beginning and end points).



(a) Multiple pan tilt images (b) Panoramic image

Fig. 2. Camera topology in pan tilt cameras.

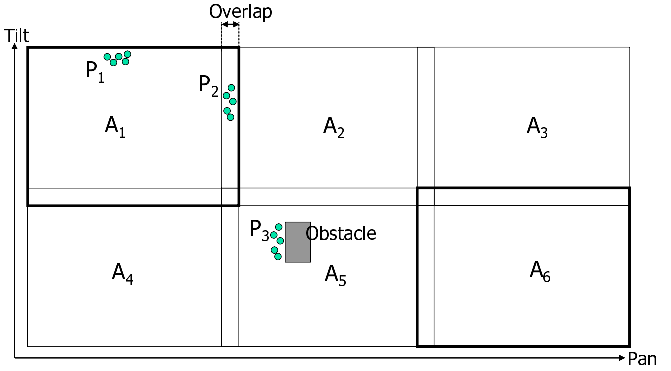


Fig. 3. Predefined pan tilt angles (FOVs) for their panoramic image. $A_1, \cdots, A_6$ indicate six partial images.

Two points observed at temporally consecutive IN/OUT events compose a route. The earlier of the two points 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 that are represented by 2D image coordinates, (2) routes are defined within a FOV (i.e., "Inside route" in Fig. 1) as well as between FOVs (i.e., "Outside route" in Fig. 1), and (3) object trajectories between the beginning and end points (e.g., straight trajectories or curve trajectories) are not represented by the route information. The data observed at IN and OUT events (i.e., image coordinates $\boldsymbol{P}(C)$ in the FOV of camera C and time T at each entrance/exit event) are called *IN data* and *OUT data*, respectively.

The basic algorithm for probabilistic topological calibration[18] using a number of IN/OUT data is as follows:

1) Each IN/OUT data is paired with all other IN/OUT data observed before its detection. In Fig. 1, each line between circles, which mean IN/OUT data, indicates a pair of IN/OUT data. Each collected pair is classified to a set, $\boldsymbol{S}^{B,E}$, where B and E denote the cameras in which the beginning and end points are observed, respectively.
2) Let $\{\boldsymbol{V}_1, \cdots, \boldsymbol{V}_{N^{B,E}}\}$ be a set of vectors, where $\boldsymbol{V}_v = (x_v^B, y_v^B, x_v^E, y_v^E, t_v)$ denote a 5D vector comprising the image coordinates of the beginning and end points and the transit time between them, and $N^{B,E}$ denote the total number of the pairs in $\boldsymbol{S}^{B,E}$. Elements in each vector are normalized between 0 and 1.
3) $\{\boldsymbol{V}_1, \cdots, \boldsymbol{V}_{N^{B,E}}\}$ are divided into several subsets, $\boldsymbol{S}_i^{B,E}$ (e.g., $\boldsymbol{S}_1^{B,E}, \boldsymbol{S}_2^{B,E}, \boldsymbol{S}_3^{B,E}$ in Fig. 1), based on similarity; LBG algorithm[21] is used.
4) The number of the vectors in each subset is then counted and its mean and standard variation (denoted by μ_b and σ_b) are computed. If the number is less than $(\mu_b - 2.5\sigma_b)$, this subset is removed (e.g., $\boldsymbol{S}_2^{B,E}$ in Fig. 1).
5) Each remaining subset corresponds to one route, in which its probabilistic model (i.e., the mean and variance of $(x_v^B, y_v^B, x_v^E, y_v^E, t_v)$) is computed.

III. Panoramic Image for Efficient FOV Representation

As described in Sec. II, the camera topology is determined for a set of FOVs in [17], [18]. It is possible to prepare the camera topology for the FOVs corresponding to several pan tilt angles in each camera as shown in Fig. 2 (a), in which each pan tilt angle is regarded as a virtual camera. In this virtual camera configuration, many routes appear between the FOVs of the same camera as well as between the FOVs of different cameras. This complex virtual camera configuration might produce much errors; increasing IN/OUT data due to increasing cameras causes errors in topology calibration as shown in [18].

In our method, therefore, all of the FOVs of a pan tilt camera are integrated into one panoramic image. These real FOVs and the panoramic FOV are called *partial images* and a *panoramic image*, respectively. With the panorama generation algorithm proposed in [20], a seamless panoramic image is obtained from images observed in arbitrary pan tilt angles. With this panoramic image, only one image representation in each camera is needed for estimating the topology of multiple pan tilt cameras in the same way as [17], [18] as illustrated in Fig. 2 (b).

For panoramic image generation, any combinations of pan tilt angles (FOVs) are acceptable as long as they do not produce any clearance between the partial images. In our method, the camera is controlled towards only several predefined angles. In an example shown in Fig. 3, six partial images ($A_1, \cdots, A_6$) are used and each pair of neighbors has an overlap. This camera control gives us two advantages.

The first advantage is tracking stability. Compared with object tracking while moving the pan tilt angles, tracking in a

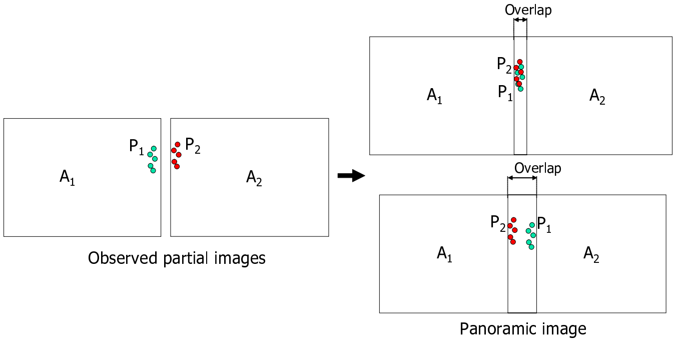


Fig. 4. Overlap between IN/OUT data detected in neighboring partial images: A_1 and A_2 are neighboring images and P_1 and P_2 are false-positive IN/OUT data detected in these images.

fixed angle is much easier.

The second advantage is wrong IN/OUT data removal. In each partial image, IN/OUT data are detected between the neighboring partial images ("P_2" in Fig. 3). These IN/OUT data are false-positive data in the panoramic image, which obstruct correct route estimation. While most true-positive IN/OUT data might be detected in the border region of the panoramic image ("P_1" in Fig. 3), some true-positive IN/OUT data might be detected inside the panoramic image if there is an obstacle in the FOV of this camera ("P_3" in Fig. 3). To remove only the false-positive IN/OUT data, IN/OUT consistency between neighboring partial images should be evaluated. Figure 4 shows how to discriminate between true and false-positive IN/OUT data. See the left and bottom right figures. If IN/OUT data P_1 are detected in the border region of A_1 but not detected in the same region of A_2 as illustrated in the bottom right figure, P_1 in A_1 should be false-positive. On the other hand, if a sufficient number of IN/OUT data are detected in the same region of the neighbors, these IN/OUT data are regarded as true-positive data. To correctly remove the false-positive IN/OUT data, they should be far away from the ones detected in the neighboring partial image. For example, if the width of the overlap between the neighbors is short, the false-positive data detected in the neighbors are mixed as shown in the top right of Fig. 4. This results in difficulty in removing the false-positive data. On the other hand, if the overlap is wide enough for separating the borders of the neighboring images as shown in the bottom right of Fig. 4, both of them can be easily removed. In our experiments, the width of the overlap was determined to be the maximum horizontal/vertical velocity of objects observed around the image border.

IV. Camera Control Strategy for Efficient Trajectory Acquisition

The greater the number of IN/OUT data becomes, the more accurate and robust the estimated camera topology gets. However, a pan tilt camera might possibly fail to observe a moving object that goes inside the FOV of the panoramic image but outside the current partial image. Similarly, assume that an object exits from a FOV that has a route(s) with another FOV. We call these FOVs having the routes *neighboring FOVs*. Even if this object enters the latter FOV, the object might be possibly lost depending on the pan tilt angles of this camera. To avoid these problems, efficient camera control that allows us to obtain IN/OUT data as much as possible is required. In our method, this camera control is achieved based on the following three functions:

- Function-1: Search of new objects
- Function-2: Tracking in a panoramic image
- Function-3: Tracking through blind spots

In what follows, these three functions are introduced (Sec. IV-A, IV-B, and IV-C) and then how to control the pan tilt cameras with these functions is described in Sec. IV-D.

A. Function-1: Search of New Objects

If no object and OUT data is currently detected in the FOV of a camera (denoted by C_c) and its neighboring FOVs, C_c is controlled so that its partial image is located where a new object is likely to appear.

With obtained IN/OUT data, the possibility of object detection in each partial image can be computed. Then each partial image I_i is observed at the following interval so that the interval is proportioanl to the number of IN/OUT data plus a minimum duration:

$$d_i = (D - ND_0)\frac{N_i^O}{\sum_{i=1}^{N} N_i^O} + D_0, \tag{1}$$

where D_0 is the minimum observation duration in each partial image and N_i^O denote the number of IN/OUT data obtained in I_i, except that $d_i = D_0$ at an initial state (i.e., when no IN/OUT data is obtained).

B. Function-2: Tracking in a Panoramic Image

If any object is currently observed in the FOV of a camera, this camera can track it by controlling the pan tilt angles in order to detect its OUT data. Notice that only predefined pan tilt angles shown in Fig. 3 are acceptable also for this tracking. While tracking among isolated FOVs is difficult, large numbers of successful methods have been proposed for object detection and tracking within a FOV (e.g., robust detection under non-stationary scenes[22] and occlusion-robust tracking[23]).

C. Function-3: Tracking through Blind Spots

It is impossible to visually track an object moving through blind spots between FOVs. However, not only for tracking with the camera topology but also for efficiently obtaining IN/OUT data for topological calibration, object transit probability estimated from the probabilistic camera topology is useful even if a sufficient amount of IN/OUT data are not obtained yet. The transit probability of object o that leaves the FOV of camera B is estimated as follows as defined in [18]:

1) Assume that camera E has a route(s) with camera B in which OUT data is detected. In each route between B and E (denoted by r) the mean (x, y) coordinates

of the beginning and end points (denoted by $\boldsymbol{\mu}_r^B, \boldsymbol{\mu}_r^E$) and their covariance matrices (denoted by $\boldsymbol{\Sigma}_r^B, \boldsymbol{\Sigma}_r^E$) are calculated.

2) Let $\boldsymbol{R}^{\cdot,E} = R_1^{\cdot,E}, \cdots, R_{N^{\cdot,E}}^{\cdot,E}$ be 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 $R_r^{\cdot,E}$ is $\boldsymbol{P}^E(C^E)$ is calculated by substituting $\boldsymbol{\mu}_r^E$ and $\boldsymbol{\Sigma}_r^E$ of $R_r^{\cdot,E}$ and $\boldsymbol{P}^E(C^E)$ for the equation of the Gaussian below:

$$Q(\boldsymbol{P}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \;=\; \frac{1}{2\pi|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left((\boldsymbol{P}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{P}-\boldsymbol{\mu})\right) \quad (2)$$

Let S be the total sum of $Q(\boldsymbol{P}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ multiplied by the number of pairs, namely,

$$S = \sum_{i=1}^{N^{\cdot,E}} Q(\boldsymbol{P}^E(C^E); \boldsymbol{\mu}_r^E, \boldsymbol{\Sigma}_r^E) Nv_r,$$

where Nv_r denote the number of IN/OUT data pairs classified into r-th route. Then, $\left(Q(\boldsymbol{P}^E(C^E); \boldsymbol{\mu}_r^E, \boldsymbol{\Sigma}_r^E) Nv_r\right)/S$ can be considered to be the probability that the end point of route $R_r^{\cdot,E}$ is regarded as the position of new detection at $\boldsymbol{P}^E(C^E)$ (denoted by $P(\boldsymbol{P}^E(C^E), R_r^{\cdot,E})$).

3) Let $\boldsymbol{R}^{B,E} = R_1^{B,E}, \cdots, R_{N^{B,E}}^{B,E}$ be a subset of $\boldsymbol{R}^{\cdot,E}$, which has the beginning point at the FOV of C^B. 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 (2).

4) The total sum of Q multiplied by P of the same route is the probability, P_p, that an object was lastly observed at $\boldsymbol{P}^B(C^B)$ before it is newly detected at $\boldsymbol{P}^E(C^E)$:

$$P_p(\boldsymbol{P}^B(C^B), \boldsymbol{P}^E(C^E)) = \sum_{x=1}^{N^{B,E}} P(\boldsymbol{P}^E(C^E), R_x^{\cdot,E}) Q(\boldsymbol{P}^B(C^B); \boldsymbol{\mu}_j^B, \boldsymbol{\Sigma}_j^B)$$

5) As similarly to equation (2), the gaussian probability below is computed:

$$\begin{aligned} P_t &= Q(T; \mu_r^{B,E}, \sigma_r^{B,E}) && (3) \\ &= \frac{1}{\sqrt{2\pi\sigma^{B,E}}} \exp\left(-\frac{(T-\mu^{B,E})^2}{2\sigma^{B,E^2}}\right) && (4) \end{aligned}$$

P_t indicates the probability that an object spends T for crossing route R_r.

At each moment, the point in which $P_p P_t$ has the maximum value is regarded as the end point in which object o is most likely to appear in camera E. If multiple OUT data are observed in the neighboring FOVs of camera E, $P_p P_t$ is computed for every OUT data and the maximum one is selected at each moment.

D. Camera Control Decision by Object Observability and Route Uncertainty

Our objective in the calibration period is to efficiently obtain useful IN/OUT data. In the calibration period, the useful IN/OUT data are detected in the beginning/end points of uncertain routes in which a sufficient number of IN/OUT data are not observed. To increase the certainty of every route, each camera should be controlled to observed IN/OUT data in these unreliable routes. After a sufficient amount of IN/OUT data are obtained, namely during the tracking period, the camera system should observe objects as many and long as possible.

In both of the calibration and tracking periods, if no object and recent OUT data is not detected in the FOV of a pan tilt camera and its neighboring FOVs, this camera is controlled based on function-1.

If one or more objects is moving through inside and/or outside routes in the FOV of a camera, C_c, one of them is selected and this C_c is controlled so that its IN/OUT data is detected. If the camera 1) is currently observing any object and/or 2) detected IN data of any object previously and its current P_t (Formula (4)) is above a threshold (this means that the object is probably still within the FOV), it is regarded that an object(s) is moving through an inside route(s) of C_c. On the other hand, if any neighboring FOVs of C_c detected OUT data previously and its current P_t is above a threshold, it is regarded that an object(s) is moving through an outside route(s) connecting to C_c. In our method, the target object is selected based on the certainty of each route and the observation probability of IN/OUT data. That is, more IN/OUT data should be detected in uncertain routes and the camera should be directed towards the beginning/end point of the route in which any object is most likely to appear. The normalized score of the uncertainty of each route is expressed by the formula below:

$$\begin{aligned} U_r &= \frac{u_r}{max}, && (5) \\ u_r &= \frac{\sigma_{r_x}^{B^2} + \sigma_{r_y}^{B^2} + \sigma_{r_x}^{E^2} + \sigma_{r_y}^{E^2} + \sigma_{r_t}^2}{Nv_r}, && (6) \end{aligned}$$

where max denotes the maximum u_r during the calibration period. The observation probabilities are computed for all possible routes of every detected object: the probability is denoted by $P(o, r)$, where o and r denote the IDs of the object and the route, respectively. The probabilities of previously detected objects moving through inside and outside routes are computed using all collected IN/OUT data based on function-3. On the other hand, the probabilities of objects currently observed in the FOV are set to be 1. Finally, the pan tilt camera C_c is controlled at each moment towards the route having the maximum value of the following weighted sum of $P(o, r)$ and $U(r)$:

$$S(o, r) = w_p P(o, r) + w_u U(r), \quad (7)$$

where w_p and w_u denote the weight variables. Note that $S(o, r)$ is changed with time because $P(o, r)$ is determined depending on the difference between the current time and the disappearance time of object o. That is, each camera is controlled momentarily depending on observation results. For example, let o_{max} and r_{max} have the maximum $S(o, r)$ at time t. If object o_{max} is not observed currently, the camera is controlled towards the end point of route r based on function-3. If object o_{max} is observed currently, the camera tracks it based on function-2. If $S(o, r)$ is lower than a threshold, the camera is controlled for search based on function-1. The same camera control manner is applied in every camera at every time.

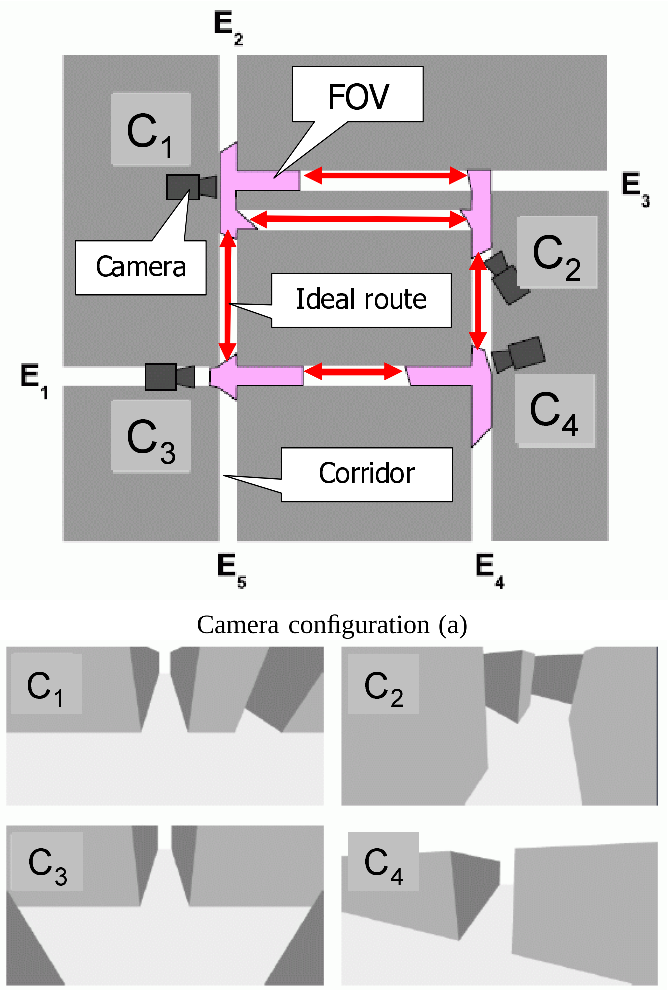
w_p and w_u are adjusted depending on reliability of the obtained probabilistic topology. Accordingly, during a calibration period, w_p and w_u should be low and high, respectively. In a tracking period, on the other hand, w_p and w_u should be high and low, respectively. Therefore, our camera control scheme is applicable to tracking with the probabilistic camera topology as well as to efficient calibration of the probabilistic camera topology.

V. Experiments

We conducted comparative experiments to confirm the efficiency of our proposed method. Both for calibration and tracking, two methods were evaluated: our proposed method and a method using simple search described in IV-A and intra-FOV visual tracking described in IV-B. To evaluate the results of different methods in the same situation in each trial, 3D simulation environments were used: simple and complex camera configurations, small and huge environments, and different object trajectories.

First of all, two small camera configurations shown in Fig. 5 and 6 were calibrated. Configuration (b) was more complex than (a) because 1) one point in a FOV connected to other two FOVs and two points in another FOV and 2) an inside route existed due to an obstacle. Objects entered the environment from one of $E_1, \cdots, E_5$ randomly, selected its direction randomly at each corner, and left from one of $E_1, \cdots, E_5$ randomly. Object trajectories and velocities were also fluctuated with gaussian noise. The mean velocity was determined so that an object moved from E_2 to E_5 in 12 seconds. At each moment, two objects were simultaneously moving in the environment. With the above manners, three kinds of object trajectories, each of which was analyzed by both methods, were generated. The resolution of each camera was assumed to be 640×480 pixel. The ranges of its pan tilt angles were $\pm 30^{\cdot}$ and $\pm 15^{\cdot}$, respectively. The predefined pan tilt angles were 12 directions (4 pan angles $\times$ 3 tilt angles). The weight variables in Formula (7) were determined as follows: $w_p = 0.2$ and $w_u = 0.8$ for the calibration period and $w_p = 1.0$ and $w_u = 0.0$ for the tracking period. The tracking period began when the normalized uncertainty score U_r was less than 0.05 in every route.

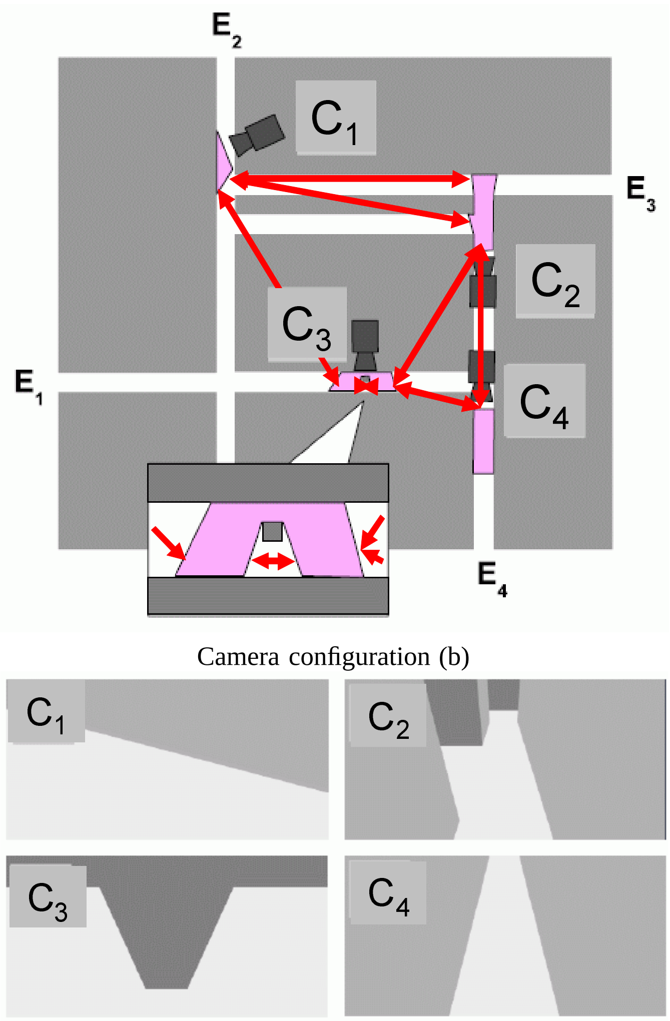
Under the above environment, the performance of the topology calibration was evaluated in terms of 1) the effectiveness of false-positive IN/OUT data removal described in Sec. III and 2) the efficiency of our proposed camera control. Figure 7 shows one of the estimated results (routes) in camera configuration (a) by our method without/with false-positive removal. In the figure, small red and blue dots indicate true-positive and false-positive IN/OUT data, respectively, and lines between the red points indicate estimated routes. It can be confirmed that incorrect routes were not detected by our method because false-positive points were removed correctly. Figure 8 shows the temporal histories of the mean value of normalized uncertainty scores (U_r in Formula (6)) in each trial in camera configuration (a). These results were obtained by camera control without the probabilistic camera topology, which is equal to the initial search scheme of our method described in Sec. IV-A, and our proposed method. The horizontal axis indicates a lapse time in the environment (not a computational time). Each line shows the mean value of all routes in each trial. Our method could obtain a sufficient amount of IN/OUT data, which got less than $U_r = 0.05$ in all routes, 1.4 times as fast as the simple search scheme. Figure 9 shows the histories of routes estimated by our method in camera configurations (a) and (b) in one of the trials. It can be confirmed that wrong routes were removed and correct routes were established over time. In (b), however, several false-positive and false-negative routes remained around the obstacle in the FOV of C_3. The magnified images around the



Camera configuration (a)

Observed images in (a)

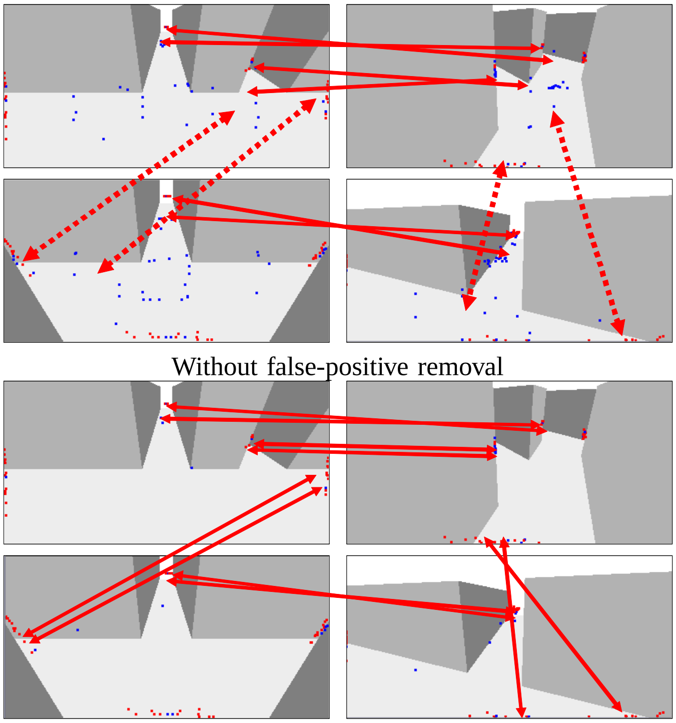
Fig. 5. Camera configurations (a).



Camera configuration (b)

Observed images in (b)

Fig. 6. Camera configurations (b).



Without false-positive removal

With false-positive removal

Fig. 7. Comparative results without/with false-positive IN/OUT data removal in camera configuration (a). Arrows indicate the estimated routes. Dotted arrows indicate false-positive routes.

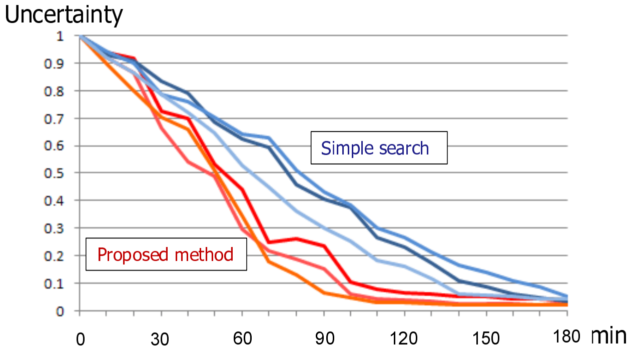


Fig. 8. Temporal histories of the uncertainty scores in camera configuration (a).

obstacle were shown in the upper row of Fig. 10. These errors were caused because tracking within the FOV of a panoramic image was disturbed by the obstacle. Assume that an object is moving from left to right in the panoramic image of C_3 as illustrated in the bottom row of Fig. 10. Its OUT data might be detected at the left boundary of the obstacle. After this OUT data detection, the pan tilt angles of C_3 are controlled based on function-1 or function-3. Unfortunately, based on function-1 (i.e., search) in a raster scan manner, the pan tilt angles are directed towards the top left of the panoramic image. Therefore, the route between the obstacle could not be detected in our experiments. To solve this problem, camera control in a search scheme should be determined not only in a routine manner but also in a random manner as similar to Monte Carlo simulated annealing.

Next, a number of cameras in a huge environment were calibrated. The huge environment was prepared by connecting the small environments (i.e., camera configurations (a) and (b) shown in Figures 5 and 6); by connecting E_1, E_2, E_3, and E_4 of camera configuration (a) to E_3, E_4, E_1, and E_2 of camera configuration (b), respectively, and vice versa, recursively. Camera configuration (a) was located in the center and then seven small environments were connected to each of E_1, E_2, E_3, and E_4 in the center; in total 116 cameras in 29 small environments. In each calibration process, all of the 116 cameras were evaluated equally without any prior with regard to the configuration of the small environments. Figure 11 shows the temporal histories of uncertainty scores in each trial by the simple search scheme and our method. As similarly to the results in camera configuration (a), our method could

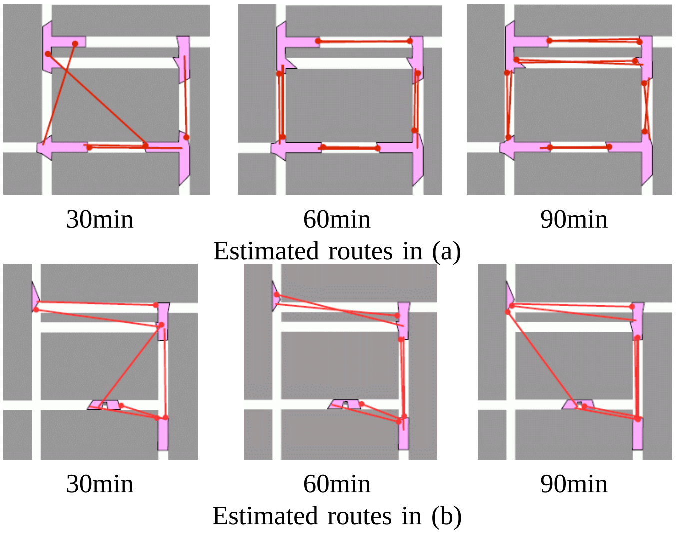


Fig. 9. Temporal histories of routes estimated by our method. Red arrows and pink regions indicate the estimated routes and the FOVs, respectively.

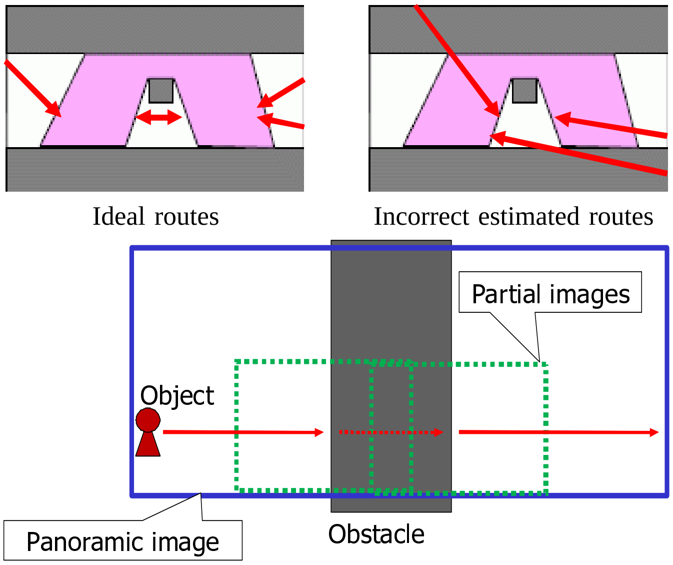


Fig. 10. Incorrect estimated routes and the cause of these errors.

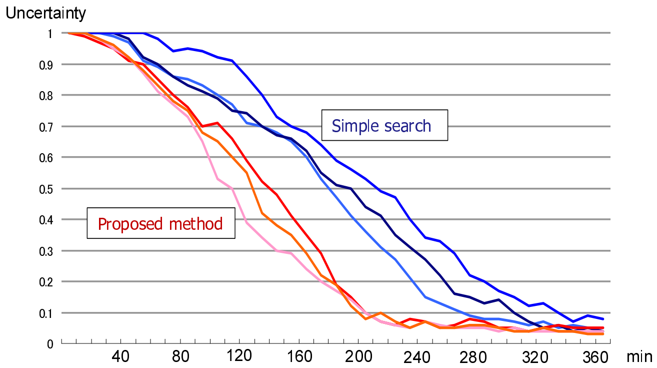


Fig. 11. Temporal histories of the uncertainty scores in the huge environment.

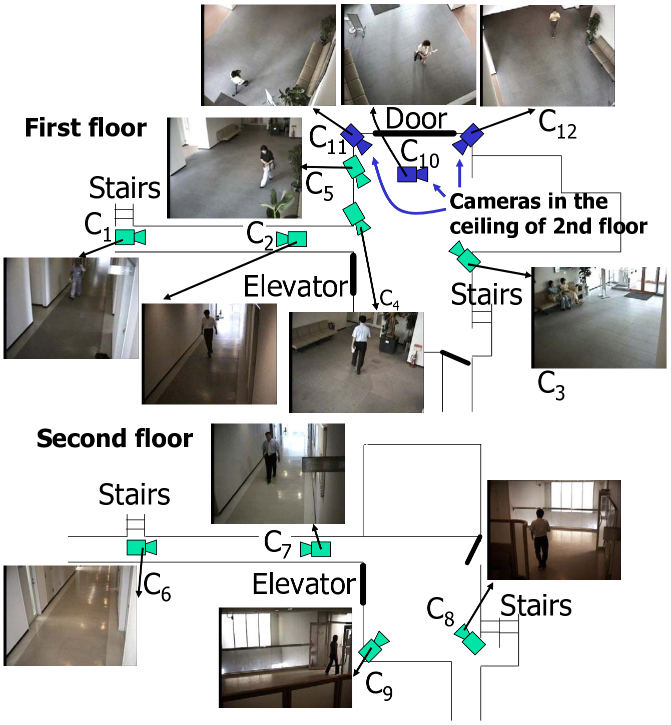


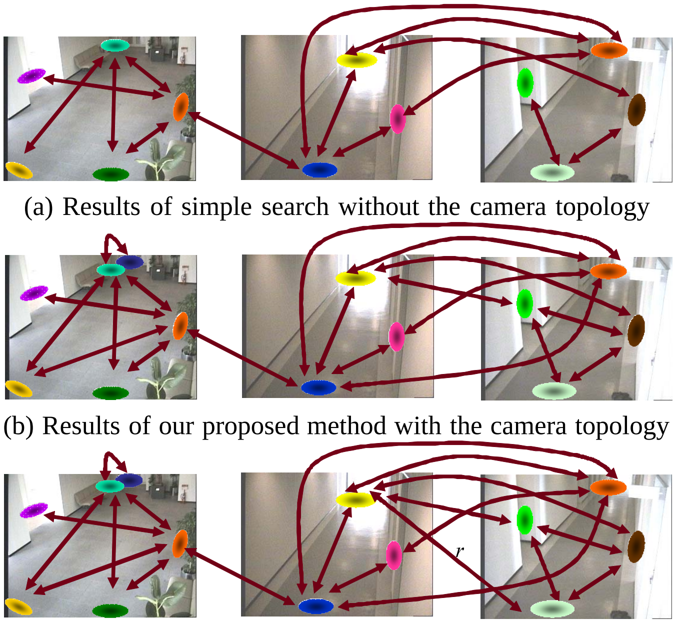
Fig. 12. View from above of the scene and image examples.

speed up camera topology calibration.

Finally, in the tracking period, the percentage of detected IN/OUT data was evaluated. Total IN/OUT data was known from the simulation data. In our proposed method with the camera topology model, the mean percentages of three trials were 97 % and 85 % in camera configurations (a) and (b), respectively. Without the probabilistic camera topology, on the other hand, the percentages were 58 % and 51 %.

The same comparative experiments were conducted with real cameras. To evaluate the results of different methods in the same object trajectories, first of all, a long image sequence was captured by each real camera. Then each real image was regarded as a panoramic image and its partial images were regarded as ones captured by a virtual pan tilt camera. In the experiments, six partial images (3 pan divisions × 2 tilt divisions) were prepared. Note that only one partial image could be observed at each moment.

Figure 12 shows the experimental environment and examples of the images captured by fixed cameras. Twelve 640 × 480 cameras were used. Object detection was implemented with a simplified version of [22]. All detected pixels were then grouped into each object region based on connectivity of the detected pixels. For tracking, each detected object region is identified with object regions in the previous frame based on proximity. Figure 13 (a) and (b) show the results. For comparison, the result obtained by [18] from the original size images is also shown in Figure 13 (c) in which all routes were verified by hand. Note that the routes that are close to



(a) Results of simple search without the camera topology

(b) Results of our proposed method with the camera topology

(c) Results of fixed camera calibration with original images

Fig. 13. Examples of detected routes. Arrows and ellipses indicate the detected routes and their beginning and end points, respectively. Only route r could not be detected in the proposed method.

each other are merged in the figures for viewability. These topology models were the results obtained when U_r was less than 0.05 in every route. With/without the camera topology model, 387/603 minutes were needed until the calibration was finished. In the tracking period, the percentage of the duration of object observation was evaluated. 100 % objects were detected from the sequences of the original size observed images. Object tracking was achieved for 300 minutes. In our proposed method with the camera topology model, the mean percentage of three trials was 91 %. Without the probabilistic camera topology, on the other hand, the percentages was 67 %. From these results, it can be confirmed that our proposed method could improve both accuracy and efficiency compared with the simple search and tracking methods without the camera topology model.

VI. Conclusion

We proposed a method for estimating the topology of distributed pan tilt cameras and its probabilistic model. To observe objects as long time as possible, pan tilt control is an important issue not only in a tracking period but also in a topology estimation period for efficient modeling. Our method controls the pan tilt cameras so that every route has reliable probabilistic data and objects are detected as many as possible.

Future work includes the following aspects:

- Appearance similarity for object identification between FOVs[24].
- Experiments with much more cameras, 1000 cameras or more. In such an environment, local camera grouping should be required for correct IN/OUT pair classification.
- Experiments with real pan tilt cameras.

References

[1] 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.

[2] K. Heath, L. Guibas, “Multi-Person Tracking from Sparse 3D Trajectories in a Camera Sensor Network,” *ICDSC*, 2008.

[3] H. Iwaki, G, Srivastava, A. Kosaka, J. Park, and A. Kak, “A novel evidence accumulation framework for robust multi-camera person detection,” *ICDSC*, 2008.

[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,” *IVC*, Vol.22, No.7, pp.551–561, 2004.

[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] 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.

[7] B. Song, C. Soto, A. K. Roy-Chowdhury, and J. A. Farrell, “Decentralized camera network control using game theory,” *ICDSC*, 2008.

[8] 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.

[9] 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.

[10] 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.

[11] H. Pasula, S. Russell, M. Ostland, and Y. Ritov, “Tracking Many Objects with Many Sensors,” in *Proc. of IJCAI*, pp.1160–1171, 1999.

[12] V. Kettnaker and R. Zabih, “Bayesian Multi-camera Surveillance,” in *Proc. of CVPR*, pp.253–259, 1999.

[13] J. E. Boyd, J. Meloche, and Y. Vardi, “Statistical Tracking in Video Traffic Surveillance,” in *Proc. of ICCV*, pp.163–168, 1999.

[14] 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.

[15] J. Davis and X. Chen, “Calibrating Pan-Tilt Cameras in Wide-area Surveillance Networks,” in *Proc. of ICCV2003*, pp.144–149, 2003.

[16] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, “Tracking Across Multiple Cameras With Disjoint Views,” in *Proc. of ICCV*, pp.952–957, 2003.

[17] D. Makris, T. Ellis, and J. Black, “Bridging the Gaps between Cameras,” in *Proc. of CVPR*, Vol.2, pp.205–210, 2004.

[18] N. Ukita, “Probabilistic-Topological Calibration of Widely Distributed Cameras,” *MVA*, Vol.18, No.3–4, pp.249–260, 2007.

[19] Y. M. Li and B. Bhanu, “Utility-based dynamic camera assignment and hand-off in a video network,” *ICDSC*, 2008.

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

[21] 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.

[22] K. Toyama, J. Krumm, B. Brumitt and B. Meyers: “WallFlower: Principles and Practice of Background Maintenance”, in *Proc. of ICCV*, pp.255–261, 1999.

[23] J. Vermaak, A. Doucet, and P. Perez: “Maintaining Multi-Modality through Mixture Tracking,” in *Proc. of ICCV*, pp.1110–1116, 2003.

[24] C. Madden, E. D. Cheng, and M. Piccardi, “Tracking people across disjoint camera views by an illumination-tolerant appearance representation,” *MVA*, Vol.18, No.3, pp.233–247, 2007.