2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

View Planning of Multiple Active Cameras for Wide Area Surveillance

Noriko Takemura Jun Miura
Department of Mechanical Engineering, Osaka University
Email: {takemura,jun}@cv.mech.eng.osaka-u.ac.jp

Abstract— **This paper describes a view planning of multiple cameras for tracking multiple persons for surveillance purposes. When only a few active cameras are used to cover a wide area, planning their views is an important issue in realizing a competent surveillance system. We develop a multi-start local search (MLS)-based planning method which iteratively selects fixation points of the cameras by which the expected number of tracked persons is maximized. Considering the fact that a person's motion can be estimated with its intermittent observations, we set a criterion which encourages frequent shifts of fixation points and develop a procedure for generating promising initial solutions for MLS. The method is shown to outperform the other approaches. We then modify the method such that it dynamically divides the cameras into mutually independent groups and determines fixation points within each group. The modified method is comparable to the original one with a much lower planning cost.**

I. Introduction

Visual surveillance is one of the active research areas in computer vision. Most previous works are concerned with development of image processing algorithms for detecting persons or vehicles reliably and/or for analyzing their activities [9], [13], [1]. This paper focuses another important problem in surveillance, namely, view planning of cameras.

One way to cover a wide area for surveillance is to use many fixed cameras whose fields of view collectively cover the area. This is, however, costly and sometimes difficult due to installation problems. We therefore take an approach of using a small number of active cameras; by appropriately controlling the fixation points of the cameras, the whole area, although it cannot be covered at a time, will be covered within a certain period of time. A key to effective surveillance in this approach is view planning of cameras.

There are several approaches to view planning of cameras/sensors. Each of them dealt with different planning problems/solutions.

Ukita and Matsuyama [14] developed a method of tracking multiple target by multiple active cameras. Multiple vision agents, each of which is responsible for controlling one camera, dynamically form several agencies (set of agents) according to the number of targets and their situations. Karuppiah et al. [6] proposed a method of dynamically configuring multiple cameras so that a target can be tracked reliably, using a utility function evaluating the measurement accuracy and the predictability of possible events. These works dealt with tracking of a few persons in a relatively small area.

Horling et al. [3] dealt with a cooperative vehicle monitoring by a distributed sensor network. They formulate the problem as a resource allocation problem in which what area to be sensed by each sensor and what information should be communicated are determined with consideration of sensor and communication uncertainties. Isler et al. [4] developed algorithms for assigning targets to multiple cameras so that the expected error in the target location estimation is minimized. These works treated the case where the number of cameras is relatively larger than that of targets.

Jung and Sukhatme [5] dealt with a coordination of multiple mobile robots to track multiple targets. They calculate the urgency over the field and use it to distribute the robots. The evaluation of urgency is based on the current distribution of targets not on a prediction of future states.

Miura and Shirai [11] dealt with a multi-camera multi-person tracking problem in the context of parallelization of planning and action. They used a heuristic planning algorithm which iteratively refines the assignment of persons to cameras, formulated as an anytime algorithm [2]. In determining fixation points, the algorithm uses one-step lookahead.

Krishna, Hexmoor, and Sogani [8] developed a view planning algorithm for a multi-sensor surveillance system. To avoid a combinatorial explosion, they dynamically prioritize the sensors based on their predicted coverage of targets. Coverage prediction is performed using statistical knowledge of the target distribution; however, they do not predict the respective motion of each person.

This paper deals with a view planning of multiple active cameras for tracking many persons. The rest of the paper is organized as follows. Sec. II defines the multi-camera multi-person tracking problem (called *MCMP problem*). Sec. III describes a model of person motion and a method of calculating the expected number of tracked persons for each camera view. Sec. IV proposes an evaluation criterion which encourages frequent changes of fixation points of cameras. Sec. V explains a multi-start local search-based planning method with an effective initial solution generation and shows that the method outperforms the others. Sec. VI modifies the method such that it dynamically divides the cameras into mutually independent groups and determines fixation points within each group. The modified method is shown to be comparable to the original one with a much lower planning cost. Sec. VII summarizes the paper and discuss future works.

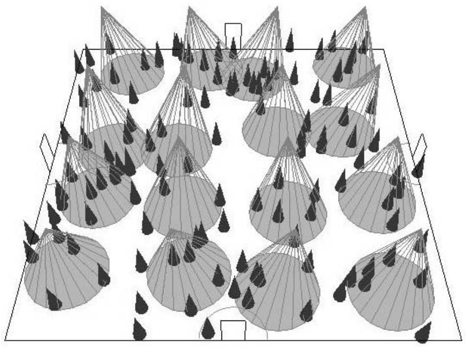


Fig. 1. MCMP simulator. Small black cones and circles on the floor indicate persons and field of views of cameras, respectively.

II. Multi-Camera Multi-Person Tracking Problem

This paper deals with the following MCMP problem. There are N_p persons arbitrarily walking in a room. There are N_c ($\ll N_p$) cameras fixed on the ceiling of the room so that no occlusions between persons occur. Each camera can change the viewing direction within a predetermined range. A single planning process controls the viewing directions of all cameras. The goal of the whole system is to track as many persons as possible during a certain period of time. Each camera is assumed to be able to identify any person and measure his/her position/velocity, as long as the person is inside the field of view (FOV) of the camera.

We made a simulator for the MCMP problem, as shown in Fig. 1. In addition to the general problem description above, we use the following detailed settings. Cameras are installed on the ceiling of $10[m]$ high. The FOV of a camera is assumed to be always a circle of $10[m]$ radius; view planning of a camera is thus equivalent to selecting its fixation point (the center of FOV) on the floor. A camera can move the fixation point within the circle of $10[m]$ radius centered at the home position right below the camera. The maximum speed of moving the fixation point is $2.5[m/s]$. The floor is discretized as a grid with $1[m]$ regular spacing and fixation points of cameras are limited to grid points. The cameras observe and change fixation points at the cycle of $1[s]$.

Concerning the number of cameras and the size of the room, we use the following two cases. In the *four-camera case*, four cameras ($N_c = 4$) are placed in a 2×2 array and the room is a $50[m] \times 50[m]$ square; fixation point candidates thus form a 100×100 grid. In the *sixteen-camera case*, sixteen cameras are place in a 4×4 array and the room is a $100[m] \times 100[m]$ square. The total area coverage at a time, which is the ratio of the sum of the areas of all FOVs to that of the room, is about $50\,\%$ for both cases.

The number of persons is 30 ($N_p = 30$) in the four-camera case and 120 ($N_p = 120$) in the sixteen-camera case. Each person basically performs a linear and constant motion but the velocity and the moving direction change every step according to a normal distribution with the variances $1.5[m^2/s^2]$ and $25[deg^2]$, respectively. When a person touches a wall, he/she changes the velocity in a regular reflection manner.

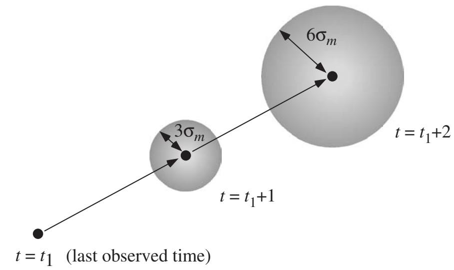


Fig. 2. Motion uncertainty model of person.

III. Prediction of Future States

Planning algorithms repeatedly determine the fixation points of all cameras at the *next* time step ($t = 1$) based on the prediction of states of tracked persons for future T time steps ($t = 1 \sim T$) (i.e., T-lookahead search).

A. Motion Modeling of Person

We use a linear motion model for predicting positions of persons. Concerning the uncertainty in prediction, we use a simple probabilistic model that the positional uncertainty of a person is isotropic and represented by the so-called 3σ portion of the normal distribution with variance $\sigma_m^2 t$, where t is the time step from the last time at which the person is observed (see Fig. 2). σ_m^2 is determined so that the predicted uncertainty covers the actual uncertainty. We assume that the position of a person can be predicted if the period of not observing the person is less than three steps; otherwise, that person's positional uncertainty is too large to be used for planning.

B. Predicting the Number of Tracked Persons for a Fixation Point

The objective of planning is to repeatedly determine fixation points that can maximize the expected number of tracked persons for a predetermined time duration. From the motion uncertainty model of person, we can calculate a set of positional distributions of the persons currently under consideration at a future time step. On the other hand, for each fixation point of a camera, its field of view (FOV) is calculated. The expected number of persons tracked by a camera directed to a specific fixation point at a time step is thus calculated as the summation of the probabilities of the persons being within the corresponding FOV.

This probability is calculated by integrating the person's positional distribution within the FOV. Since the FOVs and the distributions are both circular, we can prepare a look-up table indexed by the variance of the distribution (which is equivalently the number of steps during which a person is not in any FOVs) and the distance between the mean position and the fixation point.

When FOVs of two or more cameras overlap with each other, the calculation of the expected number becomes a little more complex. The probability that a person is within any of FOVs is calculated as follows:

- If the positional distribution of the person is completely within the FOV of at least one camera, the probability is one.
- If the distribution of the person is completely out of all FOVs, the probability is zero.
- If only a part of the distribution is within some FOVs, we classify this case into the following three subcases:
  - If that part is included only in one FOV, the probability is calculated by the table look-up.
  - If that part is included in multiple FOVs but not in any intersection of them, the probability is the sum of the probabilities of being included these FOVs (i.e., the sum of the results of the table look-up).
  - If that part is included in the intersection of some of the FOVs, we need to integrate the probabilities inside the union of such FOVs; but this is costly because the simple table look-up cannot be used.

Although the last subcase should be, in principle, treated differently from the others, we approximate the probability for the subcase with the one calculated in the same way as the other subcases because we examined many data and found that the frequency that this subcase happens is about 1%.

IV. Tracking with Frequently Changing Fixation Points

When we visually track many arbitrarily walking persons, we usually take a strategy of changing the fixation point frequently from person to person at various positions. Even if we do not look at a person for a short period of time, we can estimate (or interpolate) his/her movement from the intermittent observation data[1]. This strategy may thus achieve a high number of tracked persons while keeping a sufficient accuracy in motion estimation.

A. Evaluation Criterion for Tracking with Intermittent Observations

We assume that a low-level tracking system is working beneath the view planner. Such a system is often developed based on statistical data integration methods such as Kalman filter [7] or particle filters [10]. These methods use a probabilistic model of state evolution. Such a model usually indicates that the positional uncertainty of a target increases as time elapses if no observations are available, and that the target will eventually be lost if it is not observed for a long time.

This implies that as long as the time period during which a target is not observed is *sufficiently* short, the target's movement can reliably be estimated. In this paper, for simplicity, we set a threshold and if the non-observation time period for a target is less than or equal to the threshold, the target is considered being tracked even for that time period. Currently, we use two as the threshold. That is, when a person is observed at time t_1 and t_2 ($t_1 < t_2$) and not observed at times $\{t \,|\, t_1 < t < t_2\}$, the total number of tracking for the person given at time t_2 is $t_2 - t_1$ if $t_2 \leq t_1 + 3$ and one otherwise. We use this way of counting for calculating the expected number of tracked persons (see Sec. III-B).

[1]Note that not observations themselves but those for a person are intermittent; that is, cameras obtain observations at every time step but targets of observation may be different from time to time.

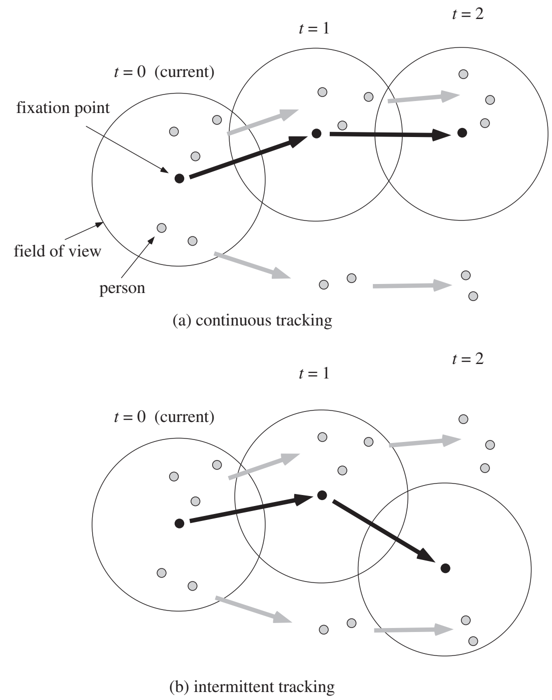
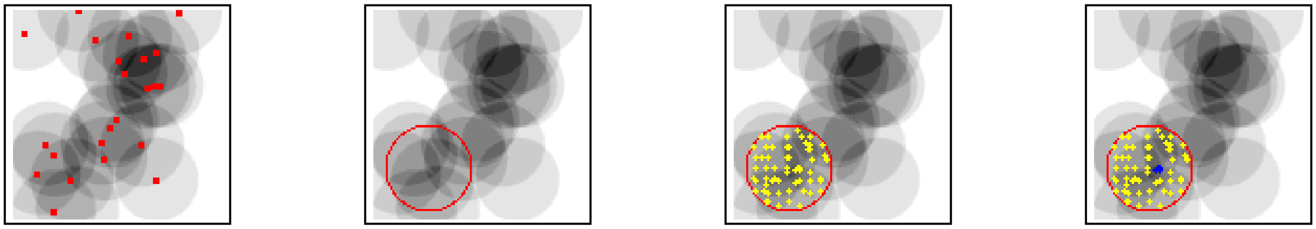


Fig. 3. Different behaviors for different evaluation criteria.

The expected number is used as the primary criterion. Fig. 3 shows an illustrative example of the behavior of cameras based on this criterion. There are two groups of persons on the upper and the lower side of the space, respectively, and the camera cannot capture both groups at times $t = 1, 2$. When we maximize the number of persons within the FOV of the camera (in the case where the persons should be continuously tracked for motion estimation), the camera moves like Fig. 3(a) and the total number of the tracked person is eleven. On the other hand, if we use the evaluation criterion explained above, the camera will move like Fig. 3(b) and the total number of the tracked persons now becomes twelve because we count two persons out of FOV at time $t = 1$; the camera tends to move to the persons that have been out of FOVs for a while.

Since several fixation points may have the same expected number, we use two more criteria for evaluation.

- The amount of movements of camera. Smaller values are better. This is for evaluating the smoothness of camera movements.
- The distance of the fixation point of a camera from its home position. Smaller values are better. This is for evaluating the distribution of camera fixation points. If persons are distributed widely in the room, then this criterion will be more important. In addition, more highly



(a) persons (red squares) and the map of expected number of tracked.

(b) a range of fixation points at a time step.

(c) representative points for uniformly-divided regions.

(d) selected promising candidate points (indicated by blue).

Fig. 4. Generating a map of promising fixation points at a time step.

distributed fixation points are better for (fortunately) capturing currently-untracked persons.

These criteria used in the following order: the expected number of tracked persons, the amount of camera movements, and the distance from the home position. If two or more solutions are equivalent in terms of a preceding criterion, the next one is used for ordering the solutions. Ties under all criteria are broken randomly.

V. Multi-Start Local Search-Based Planning

A. Multi-Start Local Search

The planning problem treated in this paper has a very large search space due to a combination of various fixation points of cameras and multi-step lookahead. Multi-start local search (MLS) is a commonly-used algorithm for solving such large-scale combinatorial problems [15]. In MLS, local search (LS) is repeated from a number of initial solutions and the best solution found during the entire search is output.

We previously compared an MLS-based method with an exhaustive search-based one and learned that the the former exhibits a comparable performance to the latter with a much less computation time [12]. MLS is thus suitable for the basic strategy for our problem.

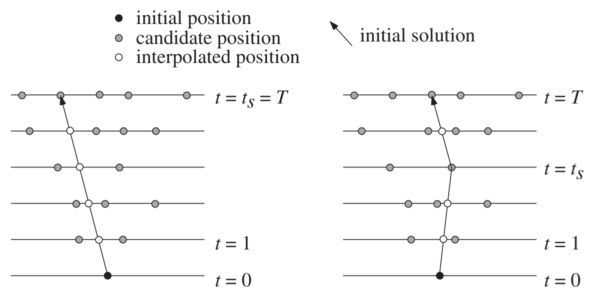
An MLS algorithm is characterized by the following: search space, neighborhood, local search strategy, and initial solution generation. We explain these in the following subsections.

B. Search Space, Neighborhood, and Local Search Strategy

In tracking with intermittent observations, fixation positions cannot be evaluated at one time but should be evaluated as a sequence of them. We therefore define the search space as all combinations of reachable fixation points of the cameras during the whole time period under consideration.

We define the neighborhood of a solution (a point in the search space) as the set of solutions in which the fixation point of only one camera at only one time is different from the solution by one step in the grid representation of 2D position (so-called 8 neighbors). Letting T be the depth of lookahead, the number of neighboring solutions is thus $8N_cT$.

We use the best admissible move strategy as the local search strategy.



(a) candidate position at time T is selected.

(b) candidate position at time $t_s < T$ is selected.

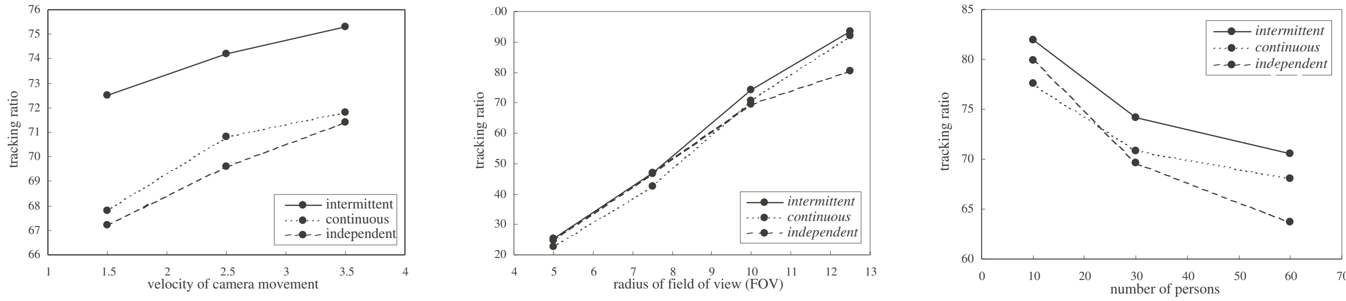
Fig. 5. Generate initial solutions.

C. Generating Initial Solutions

The search space of our MLS-based algorithm is considerably large and usually requires a large number of initial solutions to get satisfactory results, thus increasing the computation time. To keep the necessary number of initial solutions low, we take an approach of explicitly enumerating promising fixation points in *space-time* and using them for generating *good* initial solutions.

The steps for generating initial solutions are as follows. These steps are performed for each camera independently (i.e., we do not consider the overlap of FOVs at this stage).

1) Generate maps of the expected number of tracked persons for the time steps under consideration ($t = 1 \sim T$). Since the fixation points are on grid points, and since all cameras have the same characteristics, we can make a 2D grid map of the expected number of tracked persons at each time step applicable to every camera (see Fig. 4(a) for the map for a time step).
2) Divide the maps into a set of uniform-sized regions (composed of 5×5 grid points) within the movable range of each camera (see Fig. 4(b)) and select one representative point within each region which has the maximum expected number (see Fig.4(c)). The expected number becomes the score of the region.
3) Determine the maximum score and set a threshold for *promising* fixation points as the $\alpha\%$ of the maximum (currently, $\alpha = 90$). The representative points of the



(a) velocity of camera movement vs. tracking ratio. (b) radius of field of view of cameras vs. tracking ratio. (c) number of persons vs. tracking ratio.

Fig. 6. Comparison in various problem settings.

regions whose scores are higher than the threshold become a set of fixation point candidates (see Fig.4(d)).

4) Repeat the following for each camera to select N_{init} initial solutions. Select one fixation point among the candidates randomly. Let t_s be the time step at which the fixation point is. If $t_s = T$ then the fixation points at $t = 1 \sim T-1$ are determined by the interpolation. Fig. 5(a) shows such a case. The horizontal lines in the figure represent a side view of 2D maps. If $t_s < T$, then the fixation points at $t = 1 \sim t_s - 1$ are determined by the interpolation, and those at $t > t_s$ are determined recursively (select one candidate point at $t > t_s$ randomly and so on) (see Fig. 5(b)).
5) Merge N_{init} sets of initial solutions for all cameras.

D. Planning Algorithm

The planning algorithm performs MLS using the initial solutions mentioned above. We examined the performance of planning for several N_{init}'s and decided to use $N_{init} = 15$. Once the set of fixation point candidates is generated (steps 1 to 3 in the above), the rest of the initial solution generation and the local search are completely parallelizable. We thus use a PC cluster system with 15 CPU's to speed up the planning. The average computation time for one time step is about $0.3\,[sec]$ in the four-camera case.

E. Experimental Results

This section describes experimental results using 10 sets of simulation data in the four-camera case, each of which is composed of 100 step movements of 30 persons. We evaluate the methods in terms of *tracking ratio*, which is the averaged number of tracked persons per time step divided by the total number of persons. Since MLS is a randomized method, for each data set, we ran the method 10 times and calculated the average of the resulting tracking ratios.

1) Comparison with Other Methods: We here compare the following five methods:

- Proposed method (called *intermittent*).
- Another MLS-based method which does not consider intermittent tracking (called *continuous*).
- Select fixation point of each camera independently (called *independent*).
- Select fixation points randomly (called *random*).
- Fixed cameras (called *fixed*).

TABLE I
COMPARISON OF FIVE METHODS.

intermittent	continuous	independent	random	fixed
74.2%	70.8%	69.6%	48.8%	46.9%

Table I compiles the results. Note that the evaluation criterion which allows intermittent tracking (see Sec. IV-A) is used for evaluating all methods. The table shows that *random* and *fixed* produce much worse results. Among the other three, *intermittent* exhibits the best performance.

2) Comparison in Various Problem Settings: We then compare the three methods (*intermittent*, *continuous*, *independent*) in various problem settings. In general, the difference in performance between planning methods is smaller in easier problems. As the problem becomes harder, however, only *good* methods are expected to exhibit a satisfactory performance. We therefore change several parameters determining the *hardness* of the problem to examine if there exists such a tendency.

Fig. 6(a)-(c) show the comparison results for changing the maximum velocity of the camera fixation point, the radius of the field of view, and the number of persons, respectively. In all cases, the *intermittent* method outperforms the others and its performance degradation according to the problem being harder is smaller. These results show the effectiveness of the proposed *intermittent* method.

VI. GROUPING OF CAMERAS FOR REDUCING THE PLANNING COST

The planning method that determines fixation points simultaneously has a larger search space than the one that determines a fixation point for each camera independently, and is thus more costly. To reduce the planning cost, we divide the cameras into mutually independent groups and determine fixation points within each group.

A. Dividing the Cameras into Mutually Independent Groups

We divide the cameras into mutually independent groups based on the relations between their fixation points and on the distribution of tracked persons. A group of cameras can be

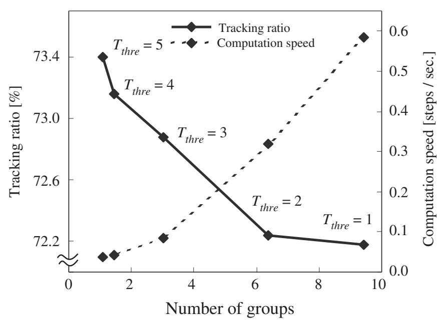
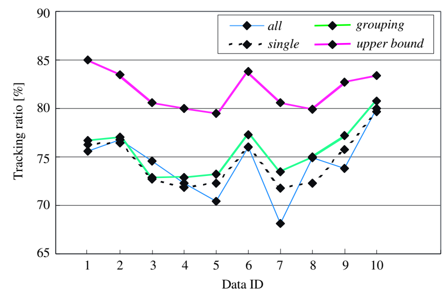
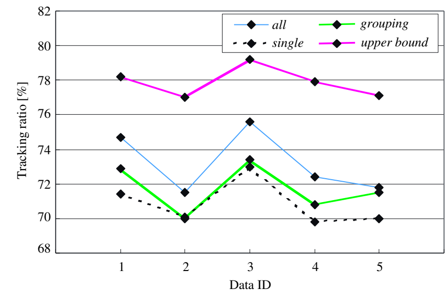


Fig. 8. Tracking ratio for changing T_{thre}.

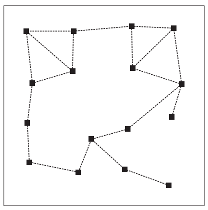


(a) four-camera case.

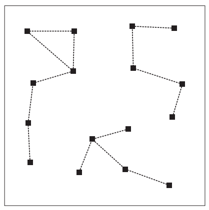


(b) sixteen-camera case.

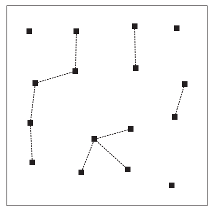
Fig. 9. Comparison results.



(a) $T_{thre} = 4$



(b) $T_{thre} = 3$



(c) $T_{thre} = 2$

Fig. 7. Example groupings of cameras.

planned independently with the others if the cameras in the group do not track the persons that are tracked by the others for the period of time under consideration.

When a person can be tracked at one time by a time horizon T_{thre} ($\leq T$) by a pair of cameras, namely, when there is a possibility of a person being in the intersection of *possible* FOVs of a pair of cameras at a time step up to T_{thre}, the pair of cameras are considered *dependent*. The possible FOV of a camera at a time is the union of the FOVs calculated from the set of fixation points that can be reached by that time. We then make groups of cameras based on the dependent relationship; a set of cameras which are connected by the dependent relationship forms one group.

Fig. 7 shows example groupings of the cameras for several time horizons. Small squares and lines indicate fixation points and dependent relationships, respectively. A longer time horizon produces a smaller number of independent groups. Fig. 8 shows the tracking ratio and the computation speed (the number of planned steps per second) for various T_{thre}'s. From the figure, $T_{thre} = 2$ or 3 seems to provide a reasonable performance with an allowable cost.

B. Planning Algorithm

The new planning algorithm with the camera grouping performs the MLS using the initial solution generation described in Sec. V. We sort the groups in the descending order of their numbers of cameras, and prioritize them in this order. In planning of the group with a priority, we consider the overlaps with the FOVs of all cameras within the groups with higher priorities.

TABLE II

TRACKING RATIO

	(a)	(b)	(c)
Four-camera case	74.2 %	75.7 %	74.5 %
Sixteen-camera case	73.2 %	71.7 %	70.9 %

TABLE III

ACHIEVEMENT RATIO

	(a) / (d)	(b) / (d)	(c) / (d)
Four-camera case	0.906	0.924	0.910
Sixteen-camera case	0.940	0.920	0.910

TABLE IV

COMPUTATION TIME

	(a)	(b)	(c)
Four-camera case	0.3 s	0.2 s	0.1 s
Sixteen-camera case	48.0 s	10.6 s	1.1 s

C. Experimental Results

We made five sets of simulation data in the sixteen-camera case, each of which is composed of 100 step movements of 120 persons. Using these data sets and the same data sets as the one used in the previous comparison (i.e., ten sets of simulation data in the four-camera case, each of which is composed of 100 step movements of 30 persons), we compare the following four methods:

(a) Determine fixation points of all cameras simultaneously (*intermittent* in Sec. V-E).
(b) Determine fixation points with grouping.
(c) Determine one fixation point after another. This is equivalent to the case where each group has a single camera in (b).
(d) Determine fixation points of all cameras simultaneously assuming all persons' movements are known. This is used as an *upper bound*.

Figs. 9(a) and 9(b) show the tracking ratio in the four-camera and the sixteen-camera case, respectively. Table II shows the averaged tracking ratios.

We analyze the results considering the following two factors that affect the tracking performance:

- *Interference between cameras*. The degree of interference between cameras goes up as the number of cameras increases. This interference can be managed to some extent by simultaneously planning multiple cameras.

- *Size of the search space*. This also increases as the number of cameras planned simultaneously increases. The larger the search space, the more difficult it would be to find a good solution.

In the four-camera case, the order of the effect of the interference between cameras is $(a) < (b) < (c)$, because, by planning multi-cameras simultaneously, the effect is reduced. On the other hand, the order of the size of the search space is $(a) > (b) > (c)$. This seems to be the reason why method (b) is the best.

In the sixteen-camera case, the orders of the two factors are the same as in the four-camera case. Although the both effects become larger as the number of cameras increases from four to sixteen, that of the size of the search space seems smaller thanks to our effective initial solution generation; the effect of the interference between cameras is thus dominant. This is considered to be the reason why method (a) is the best.

The area coverage at a time is, as mentioned above, about 50%. Our planning algorithm is, therefore, considered to achieve about a 25% increase of the performance. To see how good this is, we compared the actual results with the upper bound (method (d)). Table III gives the proportion of the tracking ratio to the upper bound. This shows the performances of the methods are quite good.

Table IV gives the computation time; it shows that the planning cost of method (b) has been considerably reduced compared to method (a).

As shown in the tables, the performance and the planning cost are in a trade-off relation. The important point is that we can manage the trade-off by adjusting the parameter T_{thre} depending on the problem settings such as the computational power and the persons' walking speed.

VII. Conclusions and Discussion

This paper has presented methods of view planning for multi-camera surveillance applications. We have defined a multi-camera multi-person tracking problem (MCMP problem), in which the objective of planning is to maximize the number of tracked persons. We introduced an evaluation criterion that allows tracking with intermittent observations thus encouraging frequent changes of fixation points. For this criterion, we have developed MLS-based method that searches the space of combinations of fixation points of all cameras during a lookahead. We also developed a method of generating initial solutions from a set of promising fixation points in space-time. This MLS-based method outperforms other methods, especially when the problem is *hard*. In order to reduce the cost while keeping the performance, we then modified the method such that it divides the cameras into mutually independent groups and determines fixation points within each group. We have shown that the modified method exhibits a comparable performance to the original method with much less computation time.

Currently, we make several assumptions: no occlusion, negligible target recognition time, perfect recognition ability. A future work is to remove these assumptions in order to consider more realistic situations such as occasional occlusion and recognition failure. Especially, when we remove the assumption of perfect recognition ability, we need to model the performance of recognition, which will decrease as the time for not observing a target increases. We then need to consider the tradeoff between increasing recognition performance by observing each target frequently and increasing the number of tracked persons by frequently changing fixation points.

Another future work is to apply the current method to the cases where the above assumptions almost hold. An example case is the one where cameras are set at high positions and persons with distinctive colors walk in a simple background. The proposed method can also be applied to the case where we analyze very large images from stationary cameras and need to select a portion of the images to analyze at each frame due to computation limitation.

References

[1] H. Buxton. Learning and understanding dynamic scene activity: a review. *Image Vision Comput.*, 21(1):125–136, 2003.

[2] T. Dean and M. Boddy. An analysis of timedependent planning. In *Proceedings of AAAI-88*, pages 49–54, 1988.

[3] B. Horling, R. Vincent, R. Mailler, J. Shen, R. Becker, K. Rawlins, and V. Lesser. Distributed sensor network for real time tracking. In *Proceedings of the 5th Int. Conf. on Autonomous Agents*, pages 417–424, 2001.

[4] V. Isler, S. Khanna, J. Spletzer, and C. Taylor. Target tracking with distributed sensors: The focus of attention problem. *Computer Vision and Image Understanding.*, 100(1-2):225–247, 2005.

[5] B. Jung and G. Sukhatme. A generalized region-based approach for multi-target tracking in outdoor environments. In *Proceedings of the 2004 IEEE Int. Conf. on Robotics and Automation.*, pages 2189–2195, 2004.

[6] D. Karuppiah, R. Grupen, A. Hanson, and E. Riseman. Smart resource reconfiguration by exploiting dynamics in perceptual tasks. In *Proceedings of the 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.*, pages 1854–1860, 2005.

[7] H. Koyasu, J. Miura, and Y. Shirai. Realtime omnidirectional stereo for obstacle detection and tracking in dynamic environments. In *Proceedings of the 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.*, pages 31–36, 2001.

[8] K. Krishna, H. Hexmoor, and S. Sogani. A t-step ahead constrained optimal target direction algorithm for a multi sensor surveillance system. In *Proceedings of the 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1840–1845, 2005.

[9] L. Lee, Raquel Romano, and Gideon Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):758–767, 2000.

[10] S. Maskell, Malcolm Rollason, Neil Gordon, and David Salmond. Efficient particle filtering for multiple target tracking with application to tracking in structured images. *Image Vision Comput.*, 21(10):931–939, 2003.

[11] J. Miura and Y. Shirai. Parallel scheduling of plannning and action for realizing an efficient and reactive robotic system. In *Proceedings of the 7th Int. Conf. on Control, Automation, Robotics and Vision*, pages 246–251, 2002.

[12] J. Miura, N. Takemura, and T. Sakiyama. View planning algorithms for a multi-camera surveillance system. In *Proc. ICAPS-2006 workshop on planning under Uncertainty and Execution Control for Autonomous Systems*, pages 7–15, 2006.

[13] C. Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):747–757, 2000.

[14] N. Ukita and T. Matuyama. Real-time cooperative multi target tracking by communicating active vision agents. In *Proceedings of 6th Int. Conf. on Information Fusion*, pages 439–446, 2003.

[15] M. Yagiura and T. Ibaraki. On metaheuristic algorithms for combinatorial optimization problems. *Systems and Computers in Japan*, 32(3):33–55, 2001.