Proc. ICAPS-2006 Workshop on Planning under Uncertainty and Execution Control for Autonomous Systems, pp. 7-15, 2006.

View Planning Algorithms for a Multi-Camera Surveillance System

Jun Miura   Noriko Takemura   Takuro Sakiyama
Department of Mechanical Engineering, Osaka University
Suita, Osaka 565-0871, Japan
{jun,takemura,saki}@cv.mech.eng.osaka-u.ac.jp

Abstract

This paper deals with a view planning of multiple active cameras for tracking multiple persons for surveillance purposes. We develop algorithms for dynamically planning viewing directions of cameras so that the expected number of tracked persons is maximized, based on a probabilistic model of person motion. Since a naive approach to this planning easily causes a combinatorial explosion, we adopt a meta-heuristic algorithm, namely, multi-start local search (MLS). We first develop an MLS-based algorithm that exhibits a comparable performance to an exhaustive search-based one but with a considerably low planning cost. We then modify the problem so that intermittent observations of a person are allowed for estimating the person's motion continuously. In this modified problem, cameras are encouraged to frequently change fixation points so that they can track as various persons as possible. For this problem, we develop another MLS-based planning method which searches the space of sequences of fixation points and uses an effective initial solution generation. Simulation results show the effectiveness of this planning method.

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 (Lee, Romano, & Stein 2000; Stauffer & Grimson 2000; Buxton 2003). 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 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.

Ukita and Matsuyama (2003) 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. (2005) 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 deal with tracking of a few persons in a relatively small area.

Horling et al. (2001) 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. (2005) 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 (2004) 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 (2002) 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 (Dean & Boddy 1988).

Krishna, Hexmoor, and Sogani (2005) 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 motion of each person independently.

This paper deals with a view planning of multiple active cameras for tracking many persons. We first define the multi-camera multi-person tracking problem (called *MCMP problem*). We then describe a model of person motion and a method of predicting the positional distributions of persons to be used for estimating the expected number of tracked persons. Concerning planning algorithms, we first compare

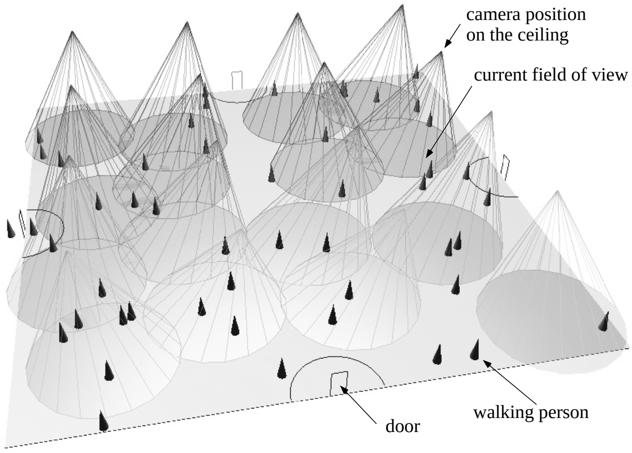


Figure 1: MCMP simulator.

two approaches, an exhaustive search-based one and a multi-start local search-based one, and show that the latter exhibits a comparable performance to the former with a considerably lower calculation cost. We then introduce a modified MCMP problem that allows a tracking with intermittent observations, and present an MLS-based planning method with an effective initial solution generation. We experimentally show that this method is better than other ones. We finally summarize the paper and discuss future works.

Multi-Camera Multi-Person Tracking Problem

This paper deals with the following MCMP problem. There are N_p persons arbitrarily waking 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 recognize any person and measure his/her position/velocity, as long as the person is inside the field of view of the camera.

We made a simulator for the MCMP problem, as shown in Fig. 1. In addition to the general problem description mentioned above, we use the following detailed settings. The room is a $50[m] \times 50[m]$ square and four cameras are placed ($N_c = 4$) in a 2×2 array on the ceiling of $10[m]$ high. The field of view (FOV) of each 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. Each 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 whole 2D space is discretized as a grid with $0.5[m]$ regular spacing and fixation points of cameras are limited to grid points; fixation point candidates thus form a 100×100 grid. The cameras observe and change

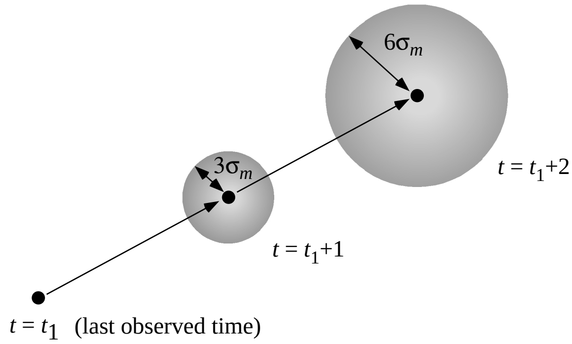


Figure 2: Motion uncertainty model of person.

fixation points at the cycle of $1[s]$.

The number of persons is 30 ($N_p = 30$) in the four camera case. Each person basically performs a linear and constant motion but the velocity and the moving direction change every step according to the 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.

We additionally use another setting in which the room is a $100[m] \times 100[m]$ square with 120 persons ($N_p = 120$) and sixteen cameras are placed in a 4×4 array ($N_c = 16$).

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 ($t = 1 \sim T$).

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.

Predicting the Number of Tracked Persons for a Fixation Point

The objective of planning is to repeatedly determine fixation points that can maximize the expectation of the 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 the

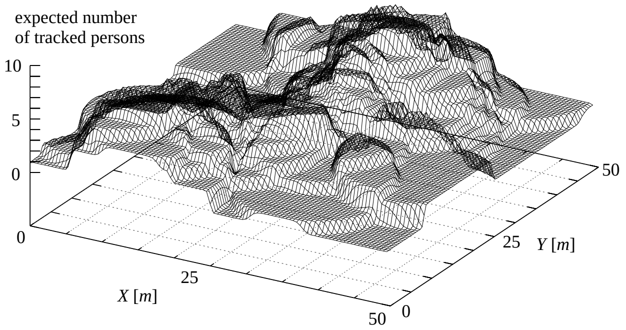


Figure 3: An example map of the expected number of tracked persons.

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. Since the fixation points are on grid points in the room and all cameras have the same characteristics, we make a 2D grid map of the expected number of tracked persons and use it for every camera. This map is generated for each future time step to be considered. Fig. 3 shows an example map for $N_p = 30$.

The probability that a person is within an FOV 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 the FOVs, 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 calculation 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 cases because we examined many data and found that the frequency that this subcase happens is about 1%.

Exhaustive Search vs. Multi-Start Local Search

This section compares two planning methods based on an exhaustive search with pruning and a multi-start local search (MLS). The performance of the former will be a benchmark for evaluating the latter.

Criteria for Evaluating Fixation Points

The primary criterion for selecting fixation points is the expected number of tracked persons for a certain period of time. Since several fixation points may have the same expected number of tracked persons, 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 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.

Exhaustive Search with Pruning

A planning method based on exhaustive search is used for obtaining optimal solutions. There are two parameters for controlling the search. T is the depth of look-ahead and V is the number of fixation point candidates to be kept at a depth. The order of the computation is thus $\mathcal{O}(V^T)$. The maximum value of V is given by C^{N_c}, where C is the number of all possible fixation point candidates for a camera and N_c is that of cameras. When this maximum value is used, the search is completely exhaustive. Due to a high computation time, we only tested the following two combinations of parameters: $(T, V) = (1, C^{N_c}),\ (2, 20)$.

We adopt two techniques for speeding up the planning. One is the pruning using an upper bound of the expected number of tracked persons, which is the one calculated under the assumption that any FOV does not overlap with the others. In examining a combination of fixation points, every time the fixation point of a camera is chosen, the upper bound of the combination is updated (using the upper bounds for the unchosen cameras) and compared with the current-best solution obtained so far. If the current combination is found to be unpromising, the computation for the combination is terminated. The upper bound for each camera is easily calculated by referring to the map described above.

Table 1: Comparison of exhaustive search- and sequential MLS-based methods.

method	look-ahead	tracking ratio (%)	std. dev. (%)	calculation time per step (sec.)	std. dev. (sec.)
exhaustive	$T = 1$	66.6	3.76	0.23	0.28
exhaustive	$T = 2$	67.5	2.47	143.62	94.0
sequential MLS	$T = 1$	65.3	3.78	0.019	0.0029
sequential MLS	$T = 2$	65.8	3.67	0.028	0.0039
sequential MLS	$T = 5$	67.3	3.53	0.073	0.0044

Another technique is to decompose the problem into a set of independent subproblems. A group of cameras can be planned independently with the other cameras as long as the FOVs of the cameras in the group do not overlap with those of the other cameras for the period of time under consideration. So we first segment cameras into such independent groups, then make a subplan for each group, and finally merge the subplans into the plan of all cameras.

Multi-Start Local Search

Multi-start local search (MLS) is a commonly-used algorithm for solving large-scale combinatorial problems (Yagiura & Ibaraki 2001). In MLS, local search (LS) is repeated from a number of initial solutions and the best solution found during the entire search is output.

In our tracking problem, the expected number of persons roughly continuously changes over the entire space and the number of local minima is expected to be relatively small (see Fig. 3). MLS is thus suitable for our problem.

Search Space, Neighborhood and Initial Solution We consider the space of all combinations of fixation points of the cameras. The search space at each time step is a subspace determined by the movable ranges of the cameras by that time step. We define the neighborhood of a solution (a point in the space) as the set of solutions in which the fixation point of only one camera is different from the solution by one step in the grid representation of 2D position (so-called 8 neighbors). The number of neighboring solutions is thus $8N_c$. We randomly generate N_{init} initial solutions within the search space.

Sequential MLS The algorithm for MLS-based planning (called *sequential MLS*) is as follows.

1. Choose N_{init} initial solution (i.e., set of fixation points for all cameras) for the next step ($t = 1$) randomly.
2. For each initial solution, repeat the following for $t = 1 \sim T$:
   (a) Perform local search for the locally-best solution. We use the *best admissible move strategy*, in which the best solution in the neighbor of the current solution is chosen as the next one.
   (b) Randomly generate one initial solution for the next time step from the locally-best solution, if $t < T$.
3. Select the first step of the best among N_{init} solutions, which maximizes the expected number of tracked persons for the duration $[1, T]$), as the movement for the next step.

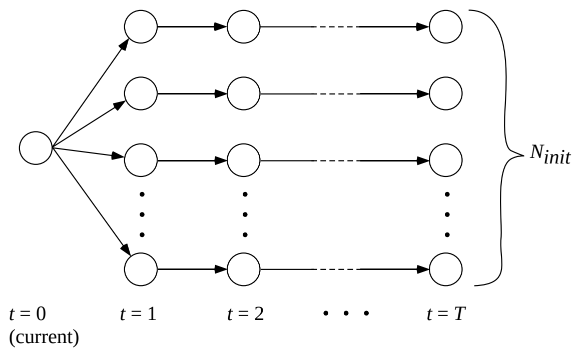


Figure 4: Search tree of *sequential MLS*. Each link indicates a pair of initial solution generation and local search.

Fig. 4 shows the search tree for *sequential MLS*. We currently use $N_{init} = 15$, which is empirically determined.

Results

We made 10 sets of simulation data, each of which is composed of 100 step movements of 30 persons. Using these sets, we compared the exhaustive search-based method with $T = 1, 2$ and the sequential MLS-based method with $T = 1, 2, 5$ for the four-camera setting. 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 the sequential MLS is a randomized method, for each data set, we ran the method 10 times and calculated the average of the resulting tracking ratios.

Table 1 summarizes the comparison results. The table shows the average tracking ratio of all the data sets. The computation time for the exhaustive search-based method becomes very large even for $T = 2$ to be used in practical systems. In addition, the variance of computation time is larger. Concerning the sequential MLS, as the look-ahead becomes longer, the performance increases while the computation time increases only approximately linearly. The sequential MLS with $T = 5$ exhibits a comparable performance to the exhaustive with $T = 2$, and spends a very short computation time, which is short enough to be used for on-line planning.

These results show that MLS-based methods are suitable for the MCMP problem.

Tracking with Frequently Changing Fixation Points

When we visually track many arbitrarily walking persons, we do not continuously track the same group of persons but 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 can thus increase the number of tracked persons while keeping a sufficient accuracy in motion estimation. This section applies this strategy to the MCMP problem using MLS.

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 (Koyasu, Miura, & Shirai 2001) or particle filters (Maskell *et al.* 2003). 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 the evaluation criterion based on this calculation of the number of tracked persons.

This change of evaluation criterion will alter the behavior of cameras. Fig.5 shows an illustrative example. 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 (by the previous evaluation criterion), the camera moves like Fig. 5(a) and the total number of the tracked person is eleven. On the other hand, if we use the new evaluation criterion, the camera will move like Fig. 5(b) and the total number of the tracked persons now becomes twelve; the camera tends to move to the persons that have been out of FOVs for a while.

Search Space and Neighborhood

In the previous MLS-based method (sequential MLS), the search space is composed of all combination of fixation points of the cameras *at one time step* and a set of fixation points is sequentially determined from the next step to the final step. In the new method, however, fixation positions cannot be evaluated at one time step but should be evaluated as a sequence of them. We therefore define the search space as all combinations of fixation points of the cameras during the whole time period under consideration.

[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.

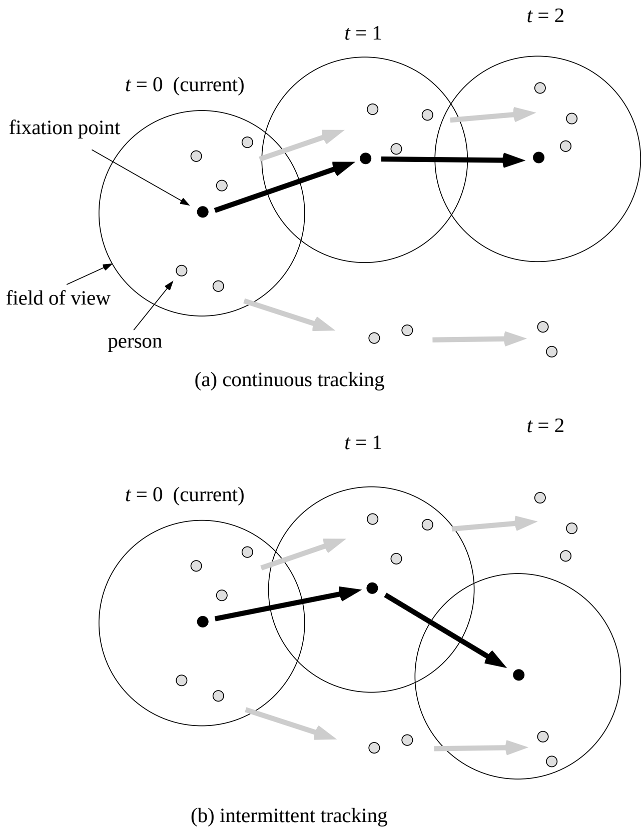


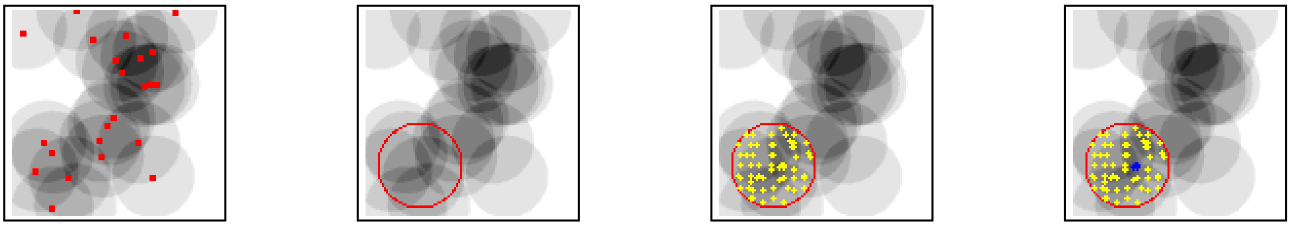
Figure 5: Different behaviors for different evaluation criteria.

We define the neighborhood of a solution 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 (again, 8 neighbors). Letting T be the depth of look-ahead, the number of neighboring solutions is thus $8N_cT$. We randomly generate N_{init} initial solutions within the search space.

We also use the best admissible move strategy in this method.

Generating Initial Solutions

The new MLS-based algorithm searches the space of a sequence of sets of fixation points up to the depth limit. This means that the search space is considerably larger than that of the previous MLS-based method (sequential MLS) which determines a set of fixation points for one time step to another. A larger search space usually requires more initial solutions to get satisfactory results in MLS, thus increasing the computation time. It is possible to use the result of sequential MLS as an initial solution. Sequential MLS, however,



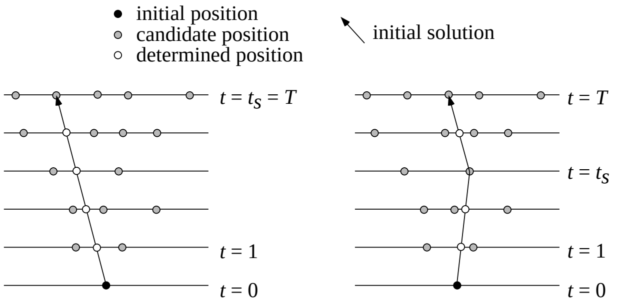
(a) persons (red squares) and the map of expected number of tracked.

(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).

Figure 6: Generating a map of promising fixation points at a time step.



(a) candidate position at time T is selected.

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

Figure 7: Generate initial solutions.

tries to generate solutions in which the same group of persons tends to be tracked *continuously* with relatively smooth camera movements (as shown in Fig. 5(a)), and may not be appropriate for generating solutions with frequent changes of fixation points (as shown in Fig. 5(b)). We thus take another approach in which promising fixation points in *space-time* are explicitly enumerated and used for generating 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, as described above, for the time steps under consideration ($t = 1 \sim T$) (see Fig. 6(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. 6(b)) and select one representative point within each region which has the maximum expected number (see Fig.6(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 regions whose scores are higher than the threshold become a set of fixation point candidates (see Fig.6(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. 7(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. 7(b)).

5. Merge N_{init} sets of initial solutions for all cameras.

Planning Algorithm

The new 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 $0.8\,[sec]$.

Experimental Results

This section describes experimental results using the same data sets as the one used in the previous comparison (i.e., 10 sets of simulation data, each of which is composed of 100 step movements of 30 persons).

Comparison of Methods for Generating Initial Solutions

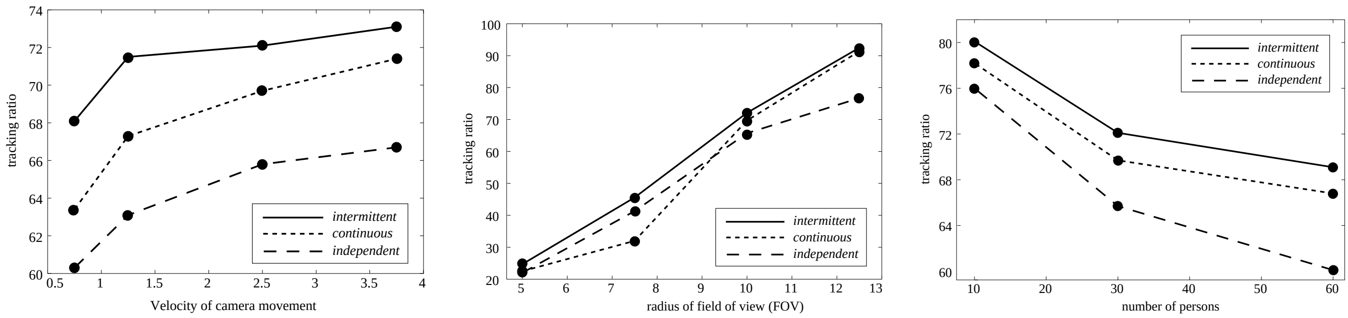
We compare the following three methods:

- Explicitly enumerates promising fixation points for generating initial solutions (proposed method).
- Use the results of the sequential MLS method as initial solutions.
- Randomly generate initial solutions.

Table 2 summarizes the comparison result. The proposed method outperforms the others.

Table 3: Comparison of five methods.

	Data ID										
	1	2	3	4	5	6	7	8	9	10	Average
intermittent	73.2	74.1	71.9	71.3	64.7	74.3	67.2	73.8	73.0	77.7	72.1
continuous	72.9	72.8	70.3	67.1	65.0	73.7	66.1	64.8	68.3	75.8	69.7
independent	67.7	69.6	65.2	66.5	56.8	67.5	60.7	63.1	64.0	60.2	64.1
random	46.1	52.0	49.9	47.4	48.5	47.8	48.1	48.2	47.0	46.8	48.2
no planning	45.6	53.2	50.2	41.3	41.9	51.5	47.2	46.8	44.7	46.1	46.9



(a) maximum velocity of camera movement vs. tracking ratio.

(b) radius of field of view of cameras vs. tracking ratio.

(c) maximum number of persons to be tracked vs. tracking ratio.

Figure 8: Comparison in various problem settings.

Table 2: Comparison of methods of generating initial solutions in terms of tracking ratio.

Method	Proposed	MLS solution	Random
Tracking ratio	72.1%	70.1%	69.1%

Comparison with Other Methods

We here compare the following five methods:

- New MLS-based method for tracking with intermittent observations (called *intermittent*).
- Sequential MLS method (called *continuous*).
- Select fixation point of each camera independently for tracking with intermittent observations (called *independent*).
- Select fixation points randomly at every time step (called *random*).
- Fixed cameras (called *no planning*).

Table 3 compiles the results. Note that the new evaluation criterion that allows tracking with intermittent observations is used for evaluating all methods. The table shows that *random* and *no planning* produce much worse results. Among the other three, the *intermittent* method exhibits the best performance.

We have also compared *intermittent* and *continuous* for another problem setting in which the room is a $100[m] \times 100[m]$ square with 120 persons ($N_p = 120$) and sixteen cameras are placed in a 4×4 array ($N_c = 16$). The averaged tracking ratios of *intermittent* and *continuous* for 3 data sets are 72.7% and 68.5%, respectively. The proposed *intermittent* method again has exhibited the best performance.

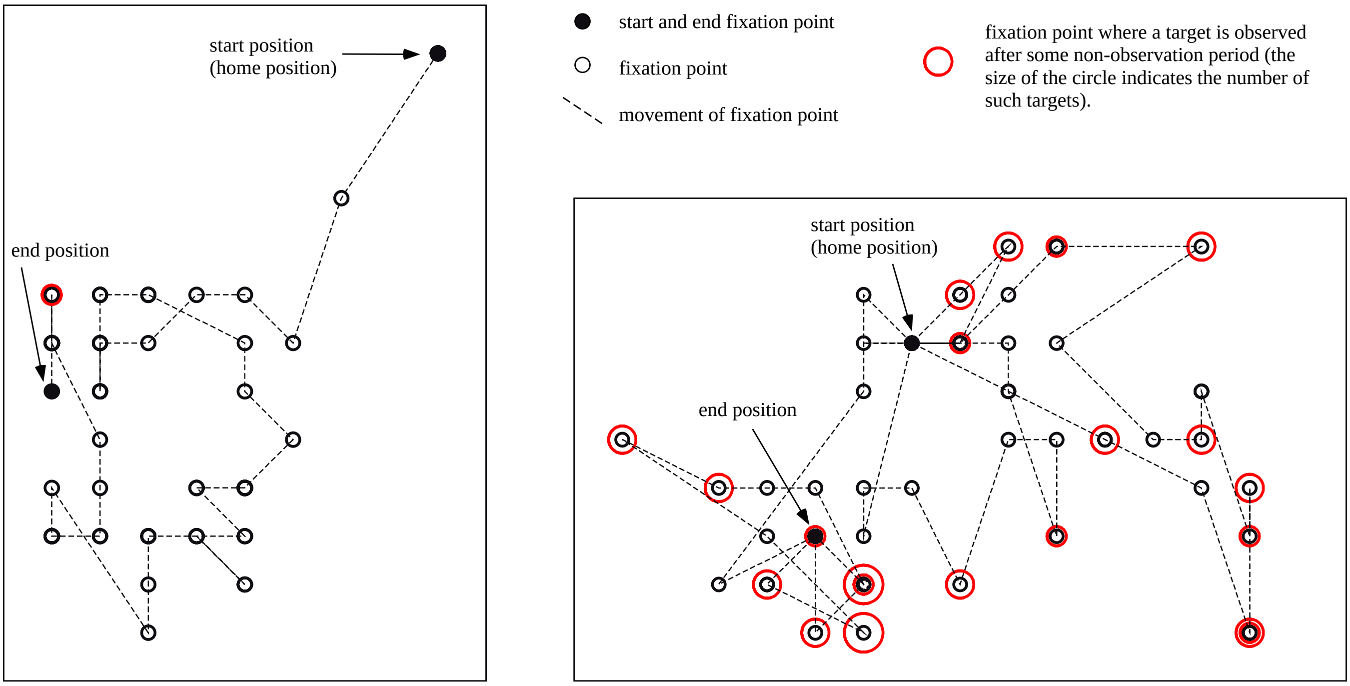
Comparison in Various Problem Settings

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. 8(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.

Comparison of Camera Behaviors for Continuous and Intermittent

Fig. 9 shows a comparison of the movements of a camera for a set of data. Fig. 9(a) shows the movement generated by the *continuous* planning method, while Fig. 9(b) shows the one generated by the *intermittent* planning method. Red circles indicate the fixation points where one or more targets



(a) *continuous* tracking

(b) *intermittent* tracking

Figure 9: Camera movements for *continuous* and *intermittent*. The start positions are the same actually.

are observed after some non-observation period. The figure shows that by *intermittent* planning method, the camera changes fixation points more frequently and widely in order to capture as various persons as possible. The total numbers of tracked persons for the whole time period (i.e., 100 time steps) for *continuous* and *intermittent* planning are 432 and 506, respectively. This also indicates the effectiveness of the proposed *intermittent* planning method.

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 first compared an exhaustive search-based method and a multi-start local search (MLS)-based method and have shown that the latter method exhibits a comparable performance to the former with much less computation time. We then introduced a new evaluation criterion that allows tracking with intermittent observations thus encouraging frequent changes of fixation points. For this criterion, we have developed another MLS-based method that searches the space of combinations of fixation points of all cameras during the look-ahead. 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*.

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

Buxton, H. 2003. Learning and understanding dynamic scene activity: A review. *Image and Vision Computing* 21(1):125–136.

Dean, T., and Boddy, M. 1988. An analysis of time-dependent planning. In *Proceedings of AAAI-88*, 49–54.

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

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

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

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

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

Krishna, K.; Hexmoor, H.; and Sogani, S. 2005. 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*, 1840–1845.

Lee, L.; Romano, R.; and Stein, G. 2000. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Trans. on Pattern Analaysis and Machine Intelligence* 22(8):758–767.

Maskell, S.; Rollason, M.; Gordon, N.; and Salmond, D. 2003. Efficient particle filtering for multiple target tracking with application to tracking in structured images. *Image and Vision Computing* 21(10):931–939.

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

Stauffer, C., and Grimson, W. 2000. Learning patterns of activity using real-time tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22(8):747–757.

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

Yagiura, M., and Ibaraki, T. 2001. On metaheuristic algorithms for combinatorial optimization problems. *Systems and Computers in Japan* 32(3):33–54.