Track Estimation of Multiple Targets from Bearing Sensor Data by Cooperative Processing

Norihiko Yoshida
Dept. of Computer and Information Sciences
Nagasaki University
Nagasaki 852-8521, JAPAN
yoshida@cis.nagasaki-u.ac.jp

Abstract

Track estimation of targets from sensor data is a crucial issue in *active dynamic scene understanding*. Multitarget motion analysis, where there are multiple moving targets and multiple fixed sensors which only measure bearings of the targets, is to associate targets and sensor data, and estimate target tracks based on that association. This is an NP-hard problem in general, and solved using stepwise relaxation. However, it is hard to obtain the optimal solution, as the method easily gets trapped in one of local optima.

We applied the decentralized cooperative search technique to this problem, and proved our method effective. The method uses more than one processors, each of which has its own partial search space, searching multiple possibilities in parallel. This approach leads to *cooperative distributed vision*. We are currently extending our method to address the case where targets move toward varying directions and in varying velocities. This report shows the current status of our research, and presents two prototypes of cooperative multiagent systems for "extended" multitarget motion analysis.

1 Introduction

This research is aiming at forming a basis for multitarget motion analysis by decentralized cooperation in the research project of 'Cooperative Distributed Vision for Dynamic Three Dimensional Scene Understanding' [1, 2] funded by The Research for the Future Program of the Japan Society for the Promotion of Science.

Localization and tracking of moving targets from data obtained by sensors is a crucial issue in *active dynamic scene understanding*. There is a class of problems called "multitarget motion analysis", where there are multiple moving targets and multiple fixed sensors which only measure bearings of the targets [3, 4, 5, 6].

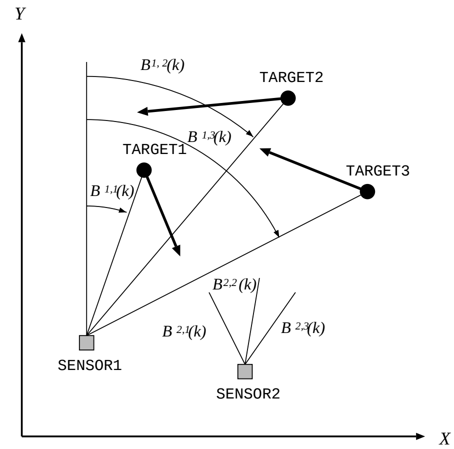


Figure 1: Multitarget Motion Analysis.

A typical target-sensor encounter is shown in Figure 1. Each sensor measures angles of the incoming signals. The positions and velocities of targets are estimated by finding the set of targets that generates bearing histories that match the bearing measurements best. The problem is divided into two: a data association problem and a bearings-only estimation problem.

Multitarget motion analysis in general is known to be an NP-hard problem, and therefore, several methods have been proposed, among which relaxation methods based on the maximum likelihood principle are the most commonly used. The conditional likelihood function of measurements is maximized over both data associations and target initial states in a stepwise manner. However, this likelihood is not a simple convex function; besides the global optimum, there are some local optima distributed randomly in the search space. The method easily gets trapped in one of these local optima. To obtain global convergence, a stochastic search method for association was proposed and proved to be effective [6]; however, it is highly computationally intensive.

There is another approach to obtain the global optimum. Decentralized cooperative search [7] is a method which uses more than one processors, each of which has its own partial search space, searching multiple possibilities in parallel. Data obtained from a sensor is sent only to near-by processors, each processor estimates target tracks from locally collected sensor data, and all the processors in the system exchange their intermediate estimation results with each other occasionally in order to ensure consistency of estimations and obtain the global optimum. This approach leads to *cooperative distributed vision*.

Most researches on multitarget motion analysis address the case where all the targets move toward fixed directions and in constant velocities. We already designed a multiagent system for such kind of multitarget motion analysis, and evaluated it by simulation [8, 9]. Our experiment proved that our cooperative method spent the same amount of time and gained better estimation qualities compared to conventional relaxation methods; in other words, our method was equivalent in estimation qualities and almost ten times faster compared to a stochastic relaxation based on simulated annealing [6].

We are currently extending our method to address the case where targets move toward varying directions and in varying velocities. This report shows the current status of our research, and presents two prototypes of cooperative multiagent systems for "extended" multitarget motion analysis.

1. Based on our previous results which realized linear track estimation, the first method repeatedly estimates short linear portions of target tracks, and overlaps them to estimate entire curve tracks.

2. The second method is a new one. It estimates the locations of targets every time, and by sequencing them, estimates arbitrary shapes of entire tracks.

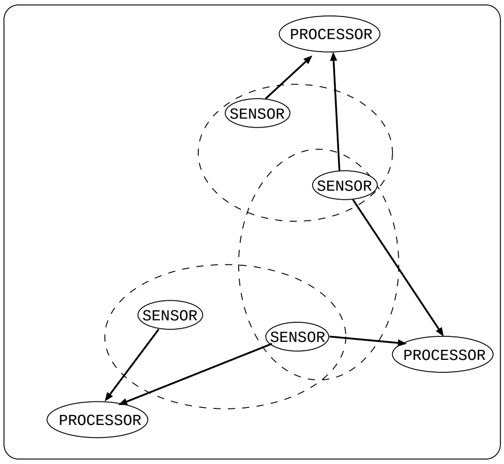


Figure 2: Decentralized Processing.

The second one is simpler and less computationally intensive, however more prone to errors and mutual occlusions than the first.

Section 2 overviews the decentralized cooperative framework for multiagent systems for multitarget motion analysis. Section 3 summarizes linear track estimation in multitarget motion analysis. Section 4 describes its extension for curve track estimation, and Section 5 describes the new sequenced location-wise estimation. Section 6 contains some concluding remarks.

2 Cooperative Framework for Track Estimation

In contrast to most previous systems for multitarget motion analysis which have centralized processors to collect all the sensor data, our system has several processors in a decentralized fashion, i.e. there is no centralized manager nor supervisor. Sensors are grouped into some, and each sensor group has its own processor. Figure 2 shows an outline of this scheme. Notice that a sensor may belong to more than one group. Sensors are fixed at some locations and measure bearings only, therefore every group must have at least two sensors.

Each processor does estimation using data from the sensors in its group. However, if a processor would do this in an entirely isolated fashion, i.e. without any interaction with other processors, the system could only get a set of some mutually-independent estimations, each based on incomplete data, and there would be no integrity nor consistency in overall estimation results. Therefore, it is important to exchange some information occasionally among processors. This is a common situation in distributed artificial intelligence systems and autonomous decentralized systems.

Exchanging raw (unprocessed) sensor data would make this decentralized system equivalent to a centralized system where all the sensor data were collected by all the processors. In our system, each processor exchanges intermediate estimations with all other processors once in an estimation iteration. Acquiring all the intermediate estimations (including its own), a processor chooses the best one in the light of its local sensor data, and then proceeds to the next iteration.

3 Linear Track Estimation

3.1 Problem Formulation

As shown in Figure 1, the system consists of n targets moving to fixed directions with constant velocities and s fixed sensors in a two-dimensional space. The state of a target t $(t = 1, \ldots, n)$ is defined by its position (r_x^t, r_y^t) and velocity (v_x^t, v_y^t) as

$$\begin{aligned} r_x^t(k) &= r_x^t(0) + k\Delta v_x^t(0) \\ r_y^t(k) &= r_y^t(0) + k\Delta v_y^t(0) \\ v_x^t(k) &= v_x^t(0) \\ v_y^t(k) &= v_y^t(0). \end{aligned}$$

where Δ is the sampling period, and k is the time index. The state of a sensor is described by its position (r_{xs}^i, r_{ys}^i). The relative state vector of the target t to the sensor i is thus

$$X^{t,i}(k) = \left(r_x^{t,i}(k), r_y^{t,i}(k), v_x^t(k), v_y^t(k)\right)',$$

and bearing data is

$$\beta^{t,i}(k) = \tan^{-1}\left[\frac{r_x^{t,i}(k)}{r_y^{t,i}(k)}\right]$$

An $n \times n$ assignment matrix $C^i(k)$, which contains only $0-1$ elements and just one 1 element in every row and column, is introduced for each measurement vector $\beta^i(k)$, as an example for the case of $n = 3$ is shown below:

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \beta^{1,i}(k) \\ \beta^{2,i}(k) \\ \beta^{3,i}(k) \end{pmatrix}$$
$$\Rightarrow \begin{pmatrix} \beta^{2,i}(k) \\ \beta^{3,i}(k) \\ \beta^{1,i}(k) \end{pmatrix} \begin{array}{l} \leftarrow \text{target } 1 \\ \leftarrow \text{target } 2 \\ \leftarrow \text{target } 3. \end{array}$$

There are $n!$ possibilities for $C^i(k)$. The entry $\left[C^i(k)\right]_{jt} = 1$ denotes that the jth element of the measurement vector $\beta^i(k)$ is associated with the target t. We seek a joint maximum likelihood solution for C^k and

$$X^t(0) = \left(r_x^t(0), r_y^t(0), v_x^t(0), v_y^t(0)\right)'$$

3.2 Track Estimation

An outline of the estimation procedure is

Procedure (centralized)
Set a suitable target initial state vector;
repeat
(1) Given a target initial state vector,
find the best assignment matrix;
(2) Given an assignment matrix,
find the best target initial state vector;
until converged.

Given an estimated target initial state of the target t

$$\hat{X}_0^t = \left(\hat{r}_x^t(0), \hat{r}_y^t(0), \hat{v}_x^t(0), \hat{v}_y^t(0)\right)',$$

each scalar bearing estimate is related to the state vector by

$$\hat{\beta}^{t,i}\left(j, \hat{X}_0\right) = \tan^{-1}\left[\frac{\hat{r}_x^{t,i}(j)}{\hat{r}_y^{t,i}(j)}\right],$$

the predicted bearing vector for the sensor i at time j is

$$\hat{\beta}^i\left(j, \hat{X}_0\right) = \left(\hat{\beta}^{1,i}\left(j, \hat{X}_0\right), \ldots, \hat{\beta}^{n,i}\left(j, \hat{X}_0\right)\right)',$$

To maximize the conditional likelihood of β^k given C^k and $\hat{X}_0$, we can minimize the corresponding average square error (ASE), which is

$$E = \frac{1}{skn} \sum_{t=1}^{n} \sum_{i=1}^{s} \sum_{j=1}^{k} \left(\frac{\beta^{c(i,j,t),i} - \hat{\beta}^{t,i}}{\sigma_i} \right)^2$$

where $m = c(i, j, t)$ if $[C^i(j)]_{mt} = 1$.

With C^k fixed, these n independent E_t's can be minimized with respect to $\hat{X}_0^t$ by the Gauss-Newton iteration. This is the step (2) in Procedure (centralized).

For a given $\hat{X}_0$, minimizing E with respect to C^k is equivalent to minimizing each individual term independently with respect to $C^i(j)$. This minimization is a linear assignment problem, in which n bearing trajectory estimates $\hat{\beta}^{1,i}(j, \hat{X}_0)$, $\cdots$, $\hat{\beta}^{n,i}(j, \hat{X}_0)$ are to be assigned n raw measurements $\beta^{1,i}(j)$, $\cdots$, $\beta^{n,i}(j)$. This is the step (1) in Procedure (centralized).

An outline of the decentralized estimation procedure is

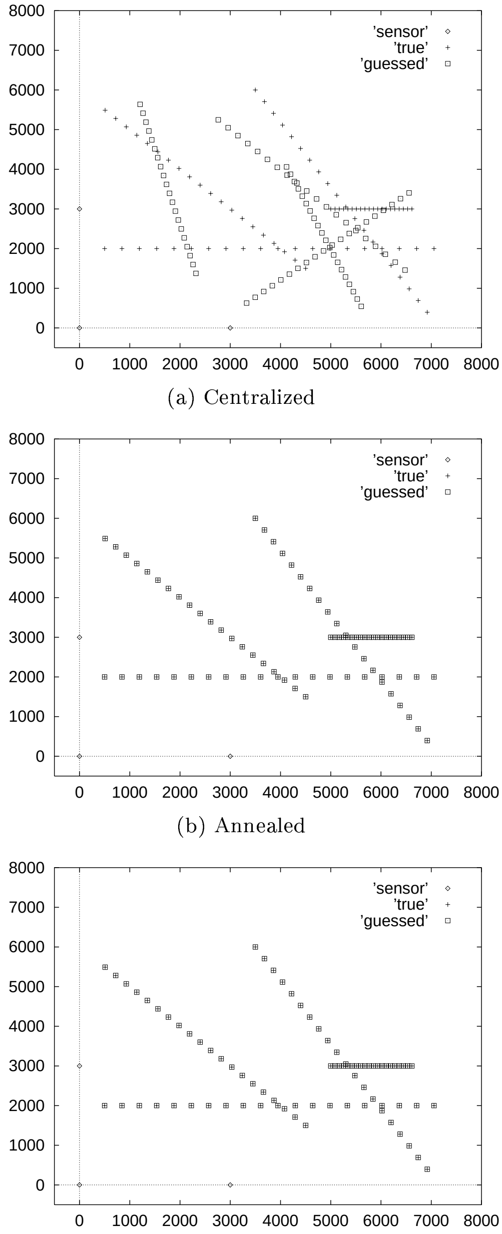
Procedure (cooperative) in each processor

Set a suitable target initial state vector;
repeat
(1) Given a target initial state vector,
find the best assignment matrix;
(2) Given an assignment matrix,
find the best target initial state vector;
(3) Send the target initial state vector
to all other processors;
(4) Receive target initial state vectors
sent from all other processors;
(5) Choose a target initial state vector
with the least ASE;
until converged.

The steps (1) and (2) in Procedure (cooperative) decrease the non-negative value E monotonically (at least do not increase), and the step (5) chooses a vector with the least ASE. Therefore, in each processor, the procedure is guaranteed to converge. When the procedure in every processor converges, the whole system terminates. Each processor has its own estimation then, and the best one is the final solution.

3.3 Simulation Experiments

We did some experiments by simulation to evaluate our system. We used randomly generated 16 cases, and compared some methods (This report shows three out of four which we tried; more details is in [8, 9]):



(a) Centralized

(b) Annealed

(c) Cooperative

Figure 3: Estimated Tracks.

Table 1: Summary of Experiment Results.

	Cent.	Ann.	Coop.
Final ASE			
Ave.	39.47	10.62	14.02
Min.	0.50	0.00	0.00
Iteration			
Ave.	9.5	121.4	9.7
Min.	3	58	3
Time (sec)			
Ave.	24.5	288.4	17.6
Min.	10	100	7

Centralized: Procedure (centralized);
Annealed: "Centralized" improved by simulated annealing;
Cooperative: Procedure (cooperative).

All the cases share some settings: there are four targets, three sensors and 100 data samples. In the cooperative method, sensors are grouped into three: each group has a processor and two sensors, and each sensor belongs to two groups.

Experiment results are summarized in Table 1. Final (global) ASE's, the number of iteration and execution time are measured for the three methods. Final ASE's show that the cooperative method gains almost the same quality of estimation as the annealed one, and more than ten times faster. Compared to the centralized one, the cooperative method is faster by 1.3, which must be the result of smaller amount of data per processor. Figure 3 shows estimation results in one example case.

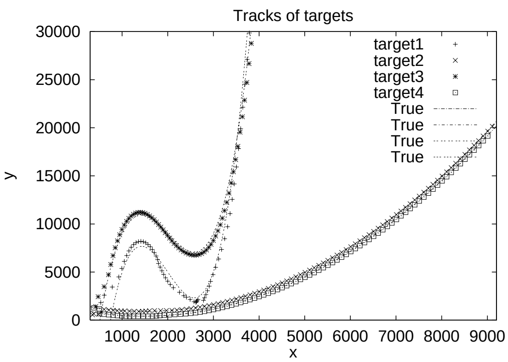


Figure 4: Curve Track Estimation.

4 Extended Linear Track Estimation

Most researches on multitarget motion analysis including our previous one address the case where all the targets move toward fixed directions and in constant velocities. We are currently extending our method to address the case where targets move toward varying directions and in varying velocities.

An extension of our linear estimation method described in the previous section is very simple. The extended method estimates short linear portions of target tracks, then overlaps them for estimating entire curve tracks. When estimation at the time t is done with a data sequence of

$$(B(t-k), \ldots, B(t))$$

then the next estimation is done at the time $t + \Delta t$ with a data sequence of

$$(B(t+\Delta t-k), \ldots, B(t+\Delta t))$$

Figure 4 shows an example of curve track estimation. Two targets move along quadratic curves, and the other two targets move along cubic curves, where $k = 100$ and $\Delta t = 10$.

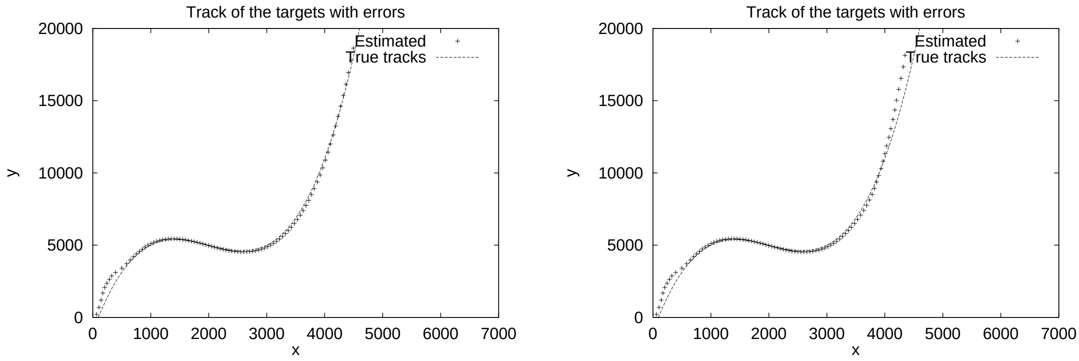
We also evaluated robustness against sensing errors. Errors of $1 - 8\%$ on bearing data was artificially added, and estimation results were observed, out of which the cases of 5% error and 8% error are shown in Figure 5 for a single target.

5 Sequenced Location Estimation

We have designed and are investigating another new method. It is to estimate the locations of targets every time, and by sequencing them, estimates arbitrary shapes of entire tracks. Therefore, this is not track estimation but location estimation.

The estimation procedure can be outlined as below.

1. Each processor collects bearing data from three sensors: sensors of itself, left and right neighbors. Let them be tagged with m, l and r for explanation.



(a) Error: 5%

(b) Error: 8%

Figure 5: Robustness against Errors.

2. A processor must determine the association between bearings and targets. It is equivalent to determine the association between $data^m$, $data^l$ and $data^r$. Data is a set of bearings $(b_1, \ldots, b_n)$ where n is the number of targets. The processor tries all the combination of b_i^m and b_j^l, b_j^l and b_k^r, and b_k^r and b_i^m.

3. Each processor computes possible locations of a target as:

$$\begin{aligned} L12 = & \quad \text{location estimated from } b_i^m \text{ and } b_j^l \ ; \\ L23 = & \quad \text{location estimated from } b_j^l \text{ and } b_k^r \ ; \\ L31 = & \quad \text{location estimated from } b_k^r \text{ and } b_i^m. \end{aligned}$$

and computes:

$$\begin{aligned} D12 = & \quad \text{distance between } L12 \text{ and } L23 \ ; \\ D23 = & \quad \text{distance between } L23 \text{ and } L31 \ ; \\ D31 = & \quad \text{distance between } L31 \text{ and } L12. \end{aligned}$$

Then the processor determines the combination of i, j and k which minimizes deviation $(D12 + D23 + D31)/3$, and estimates the location of the target as $(L12 + L23 + L31)/3$.

4. Each processor exchanges its estimation result, target locations and deviations, with each other. Then it chooses the best estimation in the light of its sensor data. The system repeats this loop until converged.

Figure 6 shows an example of location estimation with the sensing error of 0% and 5%. As easily seen, this method is very prone to sensing errors, although it is very computationally efficient.

6 Concluding Remarks

This report presented a new approach to the well-known problem of multitarget motion analysis, and showed decentralized cooperative estimation effective. More rigorous investigation for this approach should be of further study, for example, regarding the effect of occlusions and clutters.

Acknowledgments

We are grateful to Mr. Akio Mitani who implemented the original version of our multitarget motion analysis system.

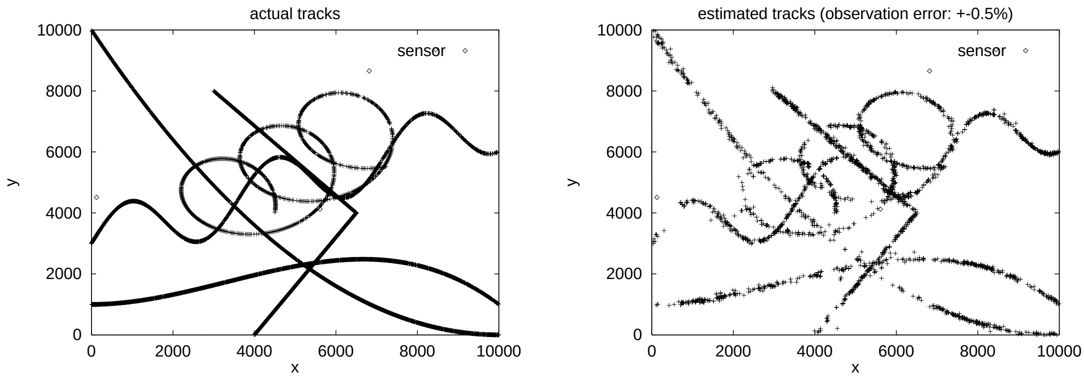
This work was supported by the Research for the Future Program of the Japan Society for the Promotion of Science, 'Cooperative Distributed Vision for Dynamic Three Dimensional Scene Understanding' Project (JSPS-RFTF96P00501).

References

[1] T. Matsuyama, "Cooperative Distributed Vision", *Proc. 1st Int'l Workshop on Cooperative Distributed Vision*, 1–28 (Oct., 1997)

[2] T. Matsuyama, "Cooperative Distributed Vision – Integration of Visual Perception, Action, and Communication –", *Proc. Image Understanding Workshop* (Nov., 1998)

[3] K. R. Pattipati, S. Deb, Y. Bar-Shalom and R. B. Washburn, "A New Relaxation Algorithm



(a) Error: 0% (b) Error: 5%

Figure 6: Location Estimation.

and Passive Sensor Data Association", *IEEE Trans. Auto. Contr.*, AC-37:2, 198–213 (1992).

[4] K. W. Lo and C. K. Li, "An Improved Multiple Target Angle Tracking Algorithm", *IEEE Trans. Aerosp. Electron. Syst.*, AES-28:3, 797–805 (1992).

[5] C. R. Rao, C. R. Sastry and B. Zhou, "Tracking the Direction of Arrival of Multiple Moving Targets", *IEEE Trans. Signal Process.*, SP-42:5, 1133–1144 (1994).

[6] P. Y. Ting and R. A. Iltis, "Multitarget Motion Analysis in a DSN", *IEEE Trans. Syst. Man Cyber.*, SMC-21:5, 1125–1139 (1991).

[7] S. Narazaki, H. Yamamura and N. Yoshida, "Strategies for Selecting Communication Structures in Cooperative Search", *Int'l J. of Cooperative Information Systems*, 4:4, 405–422 (1995).

[8] N. Yoshida and A. Mitani, "Decentralized Processing for Multitarget Motion Analysis", *Proc. IEEE Int'l Conf. on Multisensor Fusion and Integration for Intelligent Systems*, 297–303 (1996)

[9] N. Yoshida and A. Mitani, "Multitarget Motion Analysis by Decentralized Cooperation" (in Japanese), *J. Information Processing Soc. Japan*, 38:2, 206–214 (1997)

[10] N. Yoshida, T. Sayo and K. Yamamoto, "Multitarget Motion Analysis by Cooperative Multiagent System", *Proc. 2nd Int'l Workshop on Cooperative Distributed Vision*, 203–212 (1998)