Moving Object Trajectory Processing based on Multi-Laser Sensing

Yipu Zhao, Huijing Zhao, Jie Sha, and Hongbing Zha

***Abstract*— There have been many researches on moving object detection and tracking. There are also great needs in trajectory analysis and scene modeling so that to provide higher knowledge to surveillance and ITS application for decision making. However in crowded environment, trajectory data sets obtained through online processing contain many broken, group and fragment ones, which degrades trajectory quality, affect the performance in trajectory analysis and scene modeling. A trajectory processing algorithm is developed in this research on a multi-laser sensing system that was developed in our previous work. It contains a trajectory association algorithm, where an interaction graph is built to represent the relationships of trajectories; a graph-based trajectory labeling algorithm; and an EM-based trajectory parameter optimization algorithm. Experiments are conducted using the data collected at an intersection in Beijing with promising results demonstrated.**

I. INTRODUCTION

With the development of applications in ITS (Intelligent Transportation System) and surveillance, there is a great deal of needs to the method of moving object detection and tracking, e.g. in crowded traffic environments. Many researches have been devoted to video-based analysis, as cameras are getting cheaper with high image quality. Cameras can be installed on roadside to detect and track the pedestrians [1], vehicles [2] and general objects [3]. An extensive review to the current state-of-the-art in the development of visual surveillance systems is given in [4]. In addition to video-based approaches, laser scanners are getting more popular in robotics field, which are normally set on a robot or vehicle platform to detect obstacles and prevent from collision (e.g. [5]). Methods have also been developed using stationary laser scanners to detect and track pedestrians at an environment [6]. In order to take advantage of both camera and laser scanner, a framework architecture for multi-modal network sensing is proposed in [7], and fusion-based approaches using both laser and video have also been developed [8].

Based on the results of moving object detection and tracking, trajectory analysis and scene modeling are also active research topics. D. Makris [9] studies a semantic model of transportation scene. Infrastructure semantics of public area are analyzed using pedestrian trajectories in [10] and [11], Driving lanes on highway road are extracted with vehicle trajectories in [12] and [13].

Normally, trajectory analysis algorithms require that the results of moving object detection and tracking are of good quality. However, these are not always the case in many of the online systems at crowed scenes. For example, there might be many wrong trajectories which happened in cluttered situations: one moving object might hijack the trajectory of another when they crossed; the trajectory of a moving object might be broken due to occlusions, yielding a number of trajectory segments; the data of moving objects might cling together when they are close to each other, so that a trajectory of the grouped objects is tracked; the data of a single moving object might split into a number of clusters, yielding a number of fragment trajectories; etc. These wrong trajectories will influence the subsequent trajectory analysis dramatically. A procedure that refining the quality of online trajectories is required before forwarding them to the module of trajectory analysis.

There have been many researches dealing with the difficulties in moving object detection and tracking. However, most of them try to solve the problem in online procedures. A dynamic programming based data association is developed in [14]. Multiple hypothesis tracker can produce better tracking results [15]. Dynamic template based tracking works well in occlusion situations [16]. There have also been researches aiming at finding out and solving wrong trajectories. In order to connect trajectory segments, a trajectory linking method is developed in [17]. Measurement graph is used to find trajectories of groups and fragments in [18]. Based on the framework of Bayes network, fuzzy logic is used to find wrong trajectories in [19] and [20]. An inference graph is introduced to label the trajectories as object, fragment or group in [21].

A multi-laser based sensing system has been developed in our previous work, where a number of networked horizontal laser scanners were set on different locations, covering a horizontal plane of a large dynamic environment, such as an intersection; an algorithm of moving object detection and tracking on the multiple horizontal laser sensing data was developed; a number of experiments were conducted at the crowded intersection in central Beijing, as a result, the trajectories of pedestrians, bicycles, cars and buses that passed through the scenes were obtained during a long time span. However, the trajectory data quality are not as good due to the large amount of broken, group and fragment trajectories.

This paper focus on solving the problems of broken, group and fragment trajectories that frequently occur in the online processing of moving object detection and tracking, so as to provide a better data set for trajectory analysis and scene modeling. The trajectory set that obtained through multi-laser sensing are used in this research, however, we believe that some parts of the algorithm are also adaptable to visual-based

This work is partially supported by the Hi-Tech Research and Development Program of China

Y. Zhao, H. Zhao, J. Sha and H. Zha are with the State Key Lab of Machine Perception (MOE), Peking University. zhaoyp@cis.pku.edu.cn

processing. Major contributions of this research are:

1) A trajectory association algorithm is developed, where an interaction graph is built to represent the relationships of trajectories;

2) A graph-based trajectory labeling algorithm is developed that labels trajectories into objects, fragments and groups;

3) An EM-based trajectory parameter optimization algorithm is developed, which refine trajectory parameters with restrictions to both motion dynamics and observed data, i.e. multi-laser sensing data.

The rest of the paper is organized as follows. In section 2, a brief introduction to the multi-laser sensing system that was developed in our previous research is given first, as well as an outline to this research. Each processing module of this research will be addressed in details in section 3, 4 and 5, including trajectory association, labeling and parameter optimization. Experimental results and discussion are given in section 6, followed by a conclusion and future work.

II. SYSTEM OUTLINE

In this section, the multi-laser sensing system that provides the input data to this research is introduced first, followed by an outline to the algorithm developments of the research.

A. Multi-Laser Sensing System

An image of the multi-laser sensing system setting is given in Fig.1. Laser scanners are set on roadside profiling the intersection from different locations. Each is controlled by a client computer, which collects raw measurements from the sensor and performs preliminary processing on local scan data, such as background subtraction and clustering (i.e. extracting the data clusters of moving objects). All client computers are connected through a network to a server computer. The server computer collects laser scans as well as local processing results from all client computers, and conducts data integration—coordinate calibration and time synchronization. After data integration in server computer, an integrated frame that assembling the data from all laser scanners provides a view that describing a more complete horizontal contour of the moving objects at the moment. With the input of integrated data, moving object detection, tracking, classification, trajectory processing and trajectory analysis are executed subsequently.

Fig.2 displays the multi-laser sensing system's tracking results in an intersection at Beijing. Compared with video based tracking results, our tracking results provide more accurate state parameters such as direction and position. But there are more trajectory broken events in our tracking results due to the lack of color and texture information. Two trajectory broken events happened between frame 290 and frame 320 in Fig.2 (around by red circle).

A number of experiments of long time span have been conducted at the intersections in central Beijing, which are famous of their crowdedness. Large amount of trajectory data have been collected through the online-based procedure of moving object detection and tracking, which capture the motions of pedestrians, bicycles, cars and buses that passed the intersections. A set of trajectory data is visualized in Fig.3, where color denotes for different motion patterns. Some of the data sets can also be found at our website http://www.poss.pku.edu.cn. However, the data sets are not perfect, as there exist many incomplete and erroneous trajectories. This research aims at developing an algorithm to refine the trajectory qualities so as to be used in trajectory analysis and scene modeling.

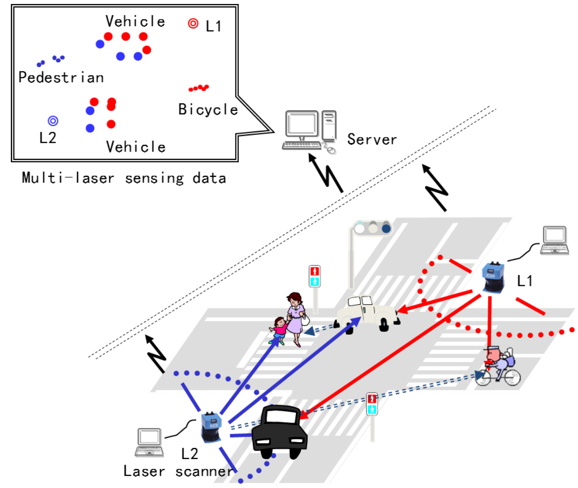


Fig. 1. Multi-laser sensing system

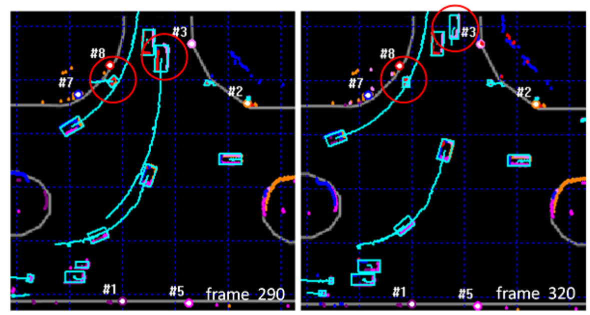
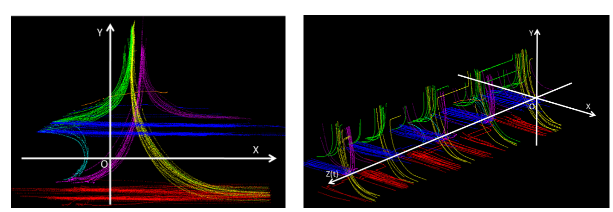


Fig. 2. Trajectories acquired by the multi-laser sensing system



(a) X-Y plane (b) X-Y-T coordinate space

Fig. 3. Trajectory clustering result

B. Outline of the Research

A three-step trajectory processing algorithm—trajectory association, labeling and parameter optimization—is proposed in the paper. In trajectory association, the events of trajectory split, merge and broken are detected, through which an interaction graph is generated. Each node of the graph denotes for a trajectory, a link connecting two nodes represents for their relationship. In trajectory labeling, an inference is conducted on the interaction graph, through which all trajectories are labeled as object—a perfect trajectory, fragment—a trajectory of a segment of moving object, or group—a merged trajectory of a number of moving objects. In trajectory parameter optimization, an EM-based method is developed to refine trajectory parameters by constricting on both motion dynamics and observation data, e.g. multi-laser sensing data. The algorithm details of each processing are introduced below.

III. TRAJECTORY ASSOCIATION

We borrow the idea from B. Bose [21] that the inclusion of tracking targets are defined as the relationships of corresponding trajectories. There are two steps in this procedure. First, the events of trajectory split, merge and broken are detected, from which the relationships between different trajectories are examined. Secondly, an interactive graph is generated on all trajectories, representing their inter-relationships. We explain each step below in details.

A. Split, Merge and Broken Events

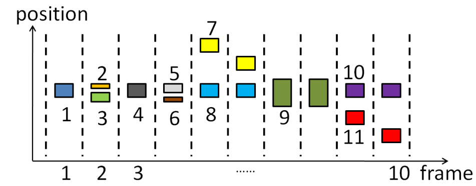
In order to examine the relationships between different trajectories, we have to find out all split, merge and broken events. A split event means that the trajectory of a single moving object is split into several ones, which start at the same time point, overlap temporally, while parallel spatially. Here, the word "same" means that within a certain threshold, the two points can be matched to one. A merge event means that the trajectories of a number of moving objects merge into a grouped one, where the number of trajectories ended at the same spatial and temporal time point. A broken event means that the trajectory of a single moving object is broken into a sequence of trajectory segments, which do not overlap temporally, while keep certain continuity spatially.

Normally, a perfect trajectory, which describes the motion of a moving object from its enter to exit of the scene, should has its start and end points on the marginal area of the scene. In other words, if a trajectory has its start point or end point located on the center area of the scene, it is considered as a participant in split, merge or broken event. Temporally, split event is one-to-many relationship, merge event is many-to-one, and broken event is one-to-one. Three rules are defined to find out these events.

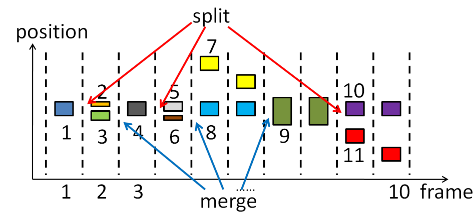
1. If a trajectory t ends in the center area, we try to find whether there are several trajectories, which start near t's end point shortly after t ends. t is considered splitting into $f_1, ..., f_k$ if trajectories $f_1, ..., f_k$ are found.

2. If a trajectory t starts in the center area, we try to find whether there are several trajectories, which end near t's start point shortly before t starts. $f_1, ..., f_k$ is considered merging into t if trajectories $f_1, ..., f_k$ are found.

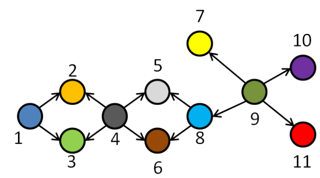
3. If a trajectory t starts in the center area, we try to find whether there is only one trajectory, which ends near t's start point shortly before t starts. It is considered that a broken event happens between t and f, if trajectory f is found.



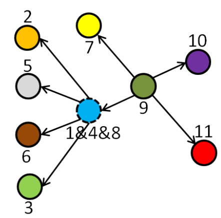
(a) 11 trajectories tracked in 10 frames



(b) Split, merge and broken events detected



(c) Initial interaction graph built with the 3 rules



(d) Interaction graph

Fig. 4. An example of interaction graph building algorithm

B. Interaction Graph Building Algorithm

An interaction graph is generated with each node denotes for a trajectory; edges between different nodes are drawn through the detection of split, merge and broken events; a direction is associated to each edge, e.g. $a \rightarrow b$, denoting that trajectory b corresponds (or partially) to the tracking target with trajectory a. An interaction graph is generated based on the following three rules.

1. For each trajectory, a node is generated at the interaction graph.

2. A directed edge $b \rightarrow a$ is added into the graph if trajectory a merges into trajectory b.

3. A directed edge $a \rightarrow b$ is added into the graph if trajectory a is split into trajectory b.

Fig.4(a) gives an example. 11 trajectories are tracked during 10 frames as shown in Fig.4(a), where different trajectories are denoted by colors. Split, merge and broken events are detected in Fig.4(b), and an interaction graph is generated based on the above three rules in Fig.4(c). For example, a split event is detected from trajectory 9 to 10 and 11, so that two directional edges $9 \rightarrow 10, 9 \rightarrow 11$ are added into the graph; a merge event is detected from trajectories 5 and 6 to 8, two directional edges $8 \rightarrow 5, 8 \rightarrow 6$ are added subsequently. We call Fig.4(c) an initial interaction graph as there exist many nodes (i.e. trajectories) that correspond to the same objects, which are merged subsequently to generate a final one as shown in Fig.4(d).

In detecting nodes of the same objects, the concept of

Target-Set Units introduced by B. Bose [21] is referred. A leaf node's tracking target cannot be further decomposed into subsets. So we call a leaf node's tracking target Target-Set Unit(TSU). Every node in the graph can be represent by corresponding TSUs. If two nodes have the same TSUs, they should be merged as they track the same target. Take Fig.4(c) as an example, we use $\{a\}$ to represent node a's corresponding tracking target. The set of target-set units is $\{2, 3, 5, 6, 7, 10, 11\}$. Then we can find that $\{1\} = \{2, 3\} = \{4\} = \{5, 6\} = \{8\}$, $\{7, 8\} = \{9\} = \{10, 11\}$. So we merge node $1, 4, 8$ and build the interaction graph in Fig.4(d).

Some directed edges are also added into interaction graph due to the inclusion of nodes' corresponding TSUs. If the TSUs of node a includes the TSUs of node b, and there is no directed edge $a \rightarrow b$ in the graph, a new edge $a \rightarrow b$ will be added into the graph.

After the node merging and edge adding step, the interaction graph which displays the relationships of trajectories is built (Fig.4(d)).

IV. TRAJECTORY LABELING

Let's give definitions to the labels for trajectories first. $Object$ denotes for a trajectory of a single moving object; $Group$ for that corresponds to a group of several moving objects; $Fragment$ for a trajectory which is tracked on a data segments of a moving object.

Recall the definitions to split/merge/broken events in previous section. There are four possible situations for a split event. $Group - Group$: a $Group$ trajectory is split into several several $Group$ trajectories. $Group - Object$: a $Group$ trajectory is split into several $Object$ trajectories. $Object - Fragment$: an $Object$ trajectory is split into several $Fragment$ trajectories. $Fragment - Fragment$: a $Fragment$ trajectory is split into several $Fragment$ trajectories. Similarly, there are four possible situations for a merge event, and four possible situations of relationships at the interaction graph accordingly.

In order to optimize trajectory parameters, it is important to infer the state of each trajectory node through its relationships with others: whether it is a group, an object or a fragment. With the interaction graph as the input, a trajectory labeling algorithm is developed by restricting on two conditions: **shape coherency** and **motion coherency**. In the followings, the two conditions are defined first, inference and labeling algorithm are described subsequently.

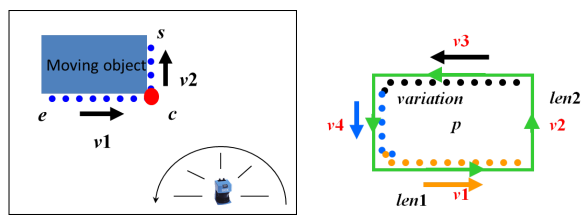
A. Condition 1 - Shape Coherency

1) Definition: Every moment, the shapes of all fragment trajectories that consist of an object are consistent with the object's contour. In order to give the mathematical description of shape coherency, we should introduce the shape model we use here.

2) Shape model: We borrow the definition to shape model from H. Zhao [22], which acts as a fundamental concept in moving object detection and tracking using multi-laser sensing data. The shape model is briefly described below.

Suppose a laser scanner does counterclockwise scanning, the horizontal contour of a car is measured by a sequence of laser points from s to e. Simplifying the shape of a car using a rectangle, edges that representing two vertical sides of the car could be detected through a corner detector and line fitting on the laser points. According to the scanning order in laser points, e.g. from a later measured point to an earlier one, a directional vector v_i is defined associating with each edge (see in Fig.5(a)). No matter where a laser scanner is placed, the directional vectors v_i are equal if they are the observations to the same side of the object.

The definition of shape model is given below (see in Fig.5(b)). Here we simplify objects using a rectangular model. Four directional vectors v_1, v_2, v_3, v_4 are the four sides of the moving object. The length of the object is $len1$, while the width is $len2$ ($len1 \geq len2$). p is the center of the object. var is the variation of distances from laser points on the object to corresponding sides, which describes the match degree of observed data and shape model. dir is the direction of the object. $speed$ is the instantaneous speed of the object, while acc is the instantaneous acceleration.



(a) The directional vectors (b) The shape model and parameters

Fig. 5. Definition of shape model

3) Evaluation: Now we give the mathematical description of shape coherency. Assume there are several trajectories $t_1, t_2, ..., t_n$. In a frame f they all exist, the observed data set s consists of these trajectories' corresponding observed data $ob_1, ob_2, ..., ob_n$. The shape model m is extracted from data set s, while every laser point in s corresponds to a directional vector in m. As already stated, the shape model parameter var describes the match degree of observed data set s and shape model m. Here we assume that var follows zero-mean Gaussian distribution(positive axis). If the tracking targets of $t_1, t_2, ..., t_n$ are fragments of the same object, the variance σ_v^2 of the distribution should be small. Otherwise the variance τ_v^2 of the distribution should be large. After all, the possibility that trajectories $t_1, t_2, ..., t_n$ meet the condition of shape coherency (denoted by s) is

$$\begin{aligned} & p(s(t_1, ..., t_n) = true|var) \\ \propto\ & p(s(t_1, ..., t_n) = true) * p(var|s(t_1, ..., t_n) = true) \\ =\ & 0.5 * 1/\sigma_v * \Phi(var/\sigma_v) \end{aligned} \tag{1}$$

Meanwhile, the possibility that they don't meet the condition of shape coherency is

$$
\begin{aligned}
& p(s(t_1, ..., t_n) = false | var) \\
\propto\ & p(s(t_1, ..., t_n) = false) * p(var | s(t_1, ..., t_n) = false) \\
=\ & 0.5 * 1/\tau_v * \Phi(var/\tau_v) \qquad (2)
\end{aligned}
$$

After normalization, the possibility of shape coherency $p(s(t_1, t_2, ..., t_n) = true | var)$ is calculated. If it is lager than a threshold (e.g. 0.7), we consider that trajectories $t_1, t_2, ..., t_n$ meet the condition of shape coherency.

B. Condition 2 - Motion Coherency

1) Definition: Every moment, the motion parameters of all fragment trajectories that consist of an object, such as instantaneous speed $speed$ and direction dir, are quite similar.

2) Evaluation: The mathematical description of motion coherency is given below. Considering two trajectories t_1, t_2, we assume the difference of $speed$—$\Delta speed$ and the angle between their dir—Δdir both follow zero-mean Gaussian distribution. If the tracking targets of t_1, t_2 are fragments of the same object, the variance σ_s^2 of $\Delta speed$'s distribution should be small, so does the variance σ_d^2 of Δdir. Otherwise the variance τ_s^2, τ_d^2 are both large. Then the possibility that t_1, t_2 meet the condition of motion coherency (denoted by m) is

$$
\begin{aligned}
& p(m(t_1, t_2) = true | \Delta speed, \Delta dir) \\
\propto\ & p(m(t_1, t_2) = true) * p(\Delta speed, \Delta dir | m(t_1, t_2) = true) \\
=\ & 0.5 * 1/\sigma_s * \Phi(\Delta speed/\sigma_s) * 1/\sigma_d * \Phi(\Delta dir/\sigma_d) \qquad (3)
\end{aligned}
$$

Meanwhile, the possibility that they don't meet the condition of motion coherency is

$$
\begin{aligned}
& p(m(t_1, t_2) = false | \Delta speed, \Delta dir) \\
\propto\ & p(m(t_1, t_2) = false) * p(\Delta speed, \Delta dir | m(t_1, t_2) = false) \\
=\ & 0.5 * 1/\tau_s * \Phi(\Delta speed/\tau_s) * 1/\tau_d * \Phi(\Delta dir/\tau_d) \qquad (4)
\end{aligned}
$$

After normalization, the possibility of motion coherency $p(m(t_1, t_2) = true | \Delta speed, \Delta dir)$ is calculated. Considering several trajectories $t_1, t_2, ..., t_n$, we find two trajectories t_i, t_j with the minimum possibility of motion coherency. This minimum possibility is set as the motion coherency possibility of trajectories $t_1, t_2, ..., t_n$. If it is lager than a threshold (e.g. 0.7), we consider that trajectories $t_1, t_2, ..., t_n$ meet the condition of motion coherency.

$$
\begin{aligned}
& p(m(t_1, t_2, ..., t_n) = true | \Delta speed, \Delta dir) \\
=\ & \min_{i,j}\{p(m(t_i, t_j) = true | \Delta speed, \Delta dir)\} \qquad (5)
\end{aligned}
$$

C. Inference

Considering a parent node p and child nodes $c_1, c_2, ..., c_k$, the exact situation of the relationship can be classified preliminarily based on the two conditions. Two rules used to classify the exact situation are given below.

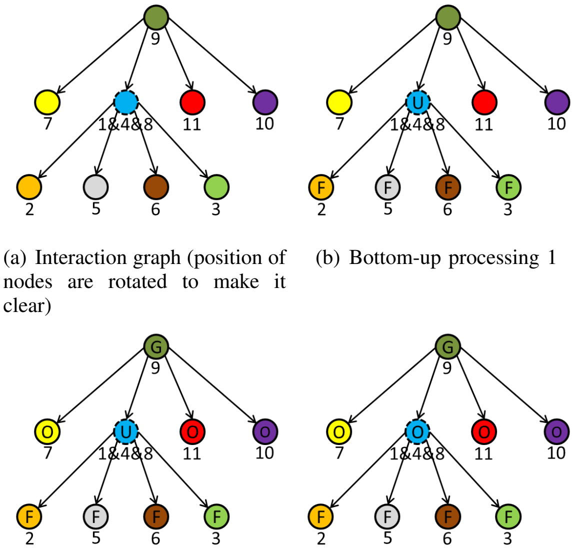
a) If trajectories stored in $c_1, c_2, ..., c_k$ meet shape coherency and motion coherency, we consider that the child nodes' tracking targets are fragments, while the parent node is object or fragment.

b) If trajectories stored in $c_1, c_2, ..., c_k$ don't meet shape coherency or motion coherency, we consider that the child nodes' tracking targets are objects, while the parent node is group.

Some prior knowledge are used to restrict possible situations.

c) If c_i is labeled as fragment, its parent p could be fragment, object or group.

d) If c_i is labeled as object or group, its parent p must be group.



(a) Interaction graph (position of nodes are rotated to make it clear) (b) Bottom-up processing 1

(c) Bottom-up processing 2 (d) Checking uncertain nodes

Fig. 6. An example of trajectory labeling algorithm

D. Trajectory Labeling Algorithm

The trajectory labeling algorithm has two steps. The first step is a bottom-up processing that labels most nodes based on the two rules. The second step is a top-down processing that checks all uncertain nodes based on the prior knowledge.

1) Bottom-up processing: Consider a node p. If p is a leaf node, we mark it as *checked*. If nodes $c_1, c_2, ..., c_k$ are child nodes of p, we start to check p when $c_1, c_2, ..., c_k$ are all marked as *checked*. A node can be labeled as uncertain if the exact label can't be inferred temporarily. Based on rules a) and b), all node are labeled as uncertain, fragment, object or group.

2) Checking uncertain nodes: Firstly, any root node labeled as fragment is updated to object. Then a top-down processing is proposed based on prior knowledge c) and d). Finally, every uncertain node is labeled as fragment, object or group. Fig.6 displays an example of the trajectory labeling algorithm with the interaction graph generated in Fig.5.

V. TRAJECTORY PARAMETER OPTIMIZATION

With the trajectories correspond to a single moving object have been associated, their multi-laser sensing data are used

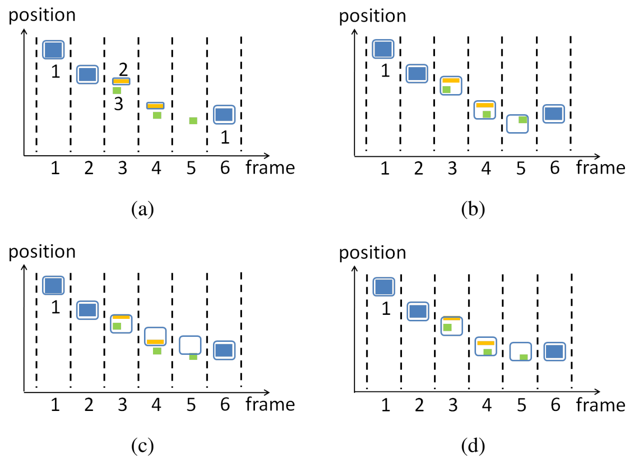


Fig. 7. An example of trajectory parameter optimization algorithm dealing with fragments. The box represents the state of trajectory. The area represents the observed data. (a) The initial states and observed data. (b) A state sequence with low expectation. The prior probability is too low. (c) A state sequence with low expectation. The likelihood measure is too low. (d) A state sequence with high expectation.

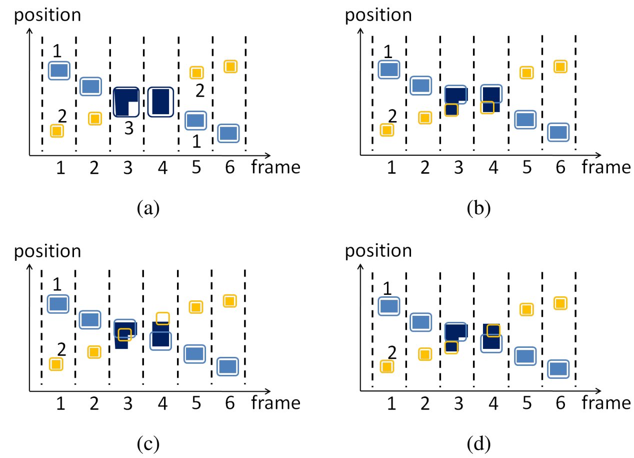


Fig. 8. An example of trajectory parameter optimization algorithm dealing with groups. (a) The initial states and observed data. (b) A state sequence with low expectation. The prior probability is too low. (c) A state sequence with low expectation. The likelihood measure is too low. (d) A state sequence with high expectation.

to find an optimized estimation to trajectory parameters, where an EM (Expectation Maximization) based algorithm is developed. We introduce EM algorithm based trajectory optimization first, then briefly describe the processing flow.

A. EM Algorithm based Trajectory Optimization

Let $os^t = \{os_1, os_2, ..., os_t\}$ denotes for the initial state sequence of a trajectory T, and $d^t = \{d_1, d_2, ..., d_t\}$ be its observation sequence, i.e. multi-laser sensing data. An EM algorithm based iteration is conducted to refine the state estimations. Suppose a state sequence $s^t[i] = \{s[i]_1, s[i]_2, ..., s[i]_t\}$ is obtained at the ith iteration. For the $i+1$th iteration, the state parameters are refined on the objective function as follows.

$$s^t[i+1] = \arg\max_s \{E_s[\ln P(d^t|s)]\} \tag{6}$$

Where s is constrained on $s^t[i]$ with a threshold ε.

$$\text{for each } s_j \epsilon s, \text{ dist}(s_j, s[i]_j) < \varepsilon$$

Let ss be a sampling sequence of s where

$$\text{for each } ss_j \epsilon ss, \text{ dist}(ss_j, s_j) < \varepsilon$$

The expectation $E_s[\ln P(d^t|s)]$ is calculated with all possible sample sequence ss.

$$E_s[\ln P(d^t|s)] = \int_{ss} \ln P(d^t|s) * p(ss) dss \tag{7}$$

Where $P(d^t|s)$ the likelihood measure, $p(ss)$ the prior probability. The estimation to them are defined below.

1) The likelihood measure: $P(d^t|s)$ is the likelihood measure denoting for that given state sequence s, the probability an observation d^t be measured. It is estimated as follows.

$$\begin{aligned} P(d^t|s) &= P\ (d_1, ..., d_t|s_1, ..., s_t) \\ &= \Pi_i\ \{P(d_i|s_i)\} \\ &= \Pi_i\ \{1/\nu * \Phi(var_i/\nu)\} \end{aligned} \tag{8}$$

var_i is the shape model parameter var of state d_i and observation s_i. It is used again to estimate the state likelihood at time i, and it is assumed that var_i is a zero-mean Gaussian distribution with a variance ν^2.

2) The prior probability: $p(ss)$ is the prior probability denoting the coherency of state sequence ss. Normally, the trajectory of a moving object changes its state smoothly. ΔX is the difference of X at time $i, i+1$. $\omega_1, \omega_2, \omega_3, \omega_4, \omega_5$ are weights of corresponding parameter changes, which are set manually. Then the prior probability is estimated as follows.

$$P(ss) = 1/\mu\ * \Phi(\overline{\Delta ss_i}/\mu) \tag{9}$$

$$\begin{aligned} \Delta ss_{i+1} = &\ \omega_1\ * \Delta len1 + \omega_2 * \Delta len2 + \omega_3 * \Delta speed \\ + &\ \omega_4 * \Delta dir + \omega_5 * \Delta acc \end{aligned} \tag{10}$$

Where the changes of state parameters like $len1$, $len2$, $speed$, dir and acc are used to describe the change of states between time $i, i+1$. The arithmetic mean value of change of states, which is a zero-mean Gaussian distribution with a variance μ^2, is used to estimate the prior probability of state sequence.

An example of parameter optimization on common tracking error $Object-Fragment-Object$ is displayed in Fig.7; while $Object - Group - Object$ in Fig.8.

B. Processing Flow

After trajectory association and labeling, each node in the interaction graph represents a tracking target labeled as fragment, object or group. Consider a moving object obj. Every object trajectory that tracks obj correctly is stored in the corresponding object node n. Every group trajectory that tracks a group of obj and other objects is stored in a node n's parent node. Every fragment trajectory that tracks a part of obj is stored in a node n's child node. Therefore, we can acquire the complete state and observation sequences of obj based on the trajectories stored in object node n, corresponding group nodes and fragment nodes. The EM algorithm based iteration is conducted then, which ends when the amount of state updating is less than a given threshold or iteration times exceed a given number.

VI. EXPERIMENTAL RESULTS

A large amount of trajectory data have been collected at the central intersections in Beijing using a multi-laser sensing system [22]. As the trajectories were tracked through an online-based processing of moving object detection and tracking, there exist many broken, group and fragment ones, which greatly degrade the quality of trajectory data sets, affect the performance in trajectory analysis and scene modeling.

An experiment is conducted in this research using a data set that was collected through a 20 minutes acquisition (44785 frames of multi-laser sensing data), and contains 6281 trajectories (see in Table 1).

A number of typical results are picked up and shown in Fig.9 and Fig.10. The figures on the left column show the trajectories tracked through the online processing of detection and tracking, while the right ones are the results after trajectory processing. The shape model of each detected object is represented using a rectangle in light blue, a line as well as a number associated to the rectangle denotes for the trajectory and ID of the moving object.

Fig.9 demonstrates the processing results for fragment trajectories, which are marked using red circles. For example at frame 530, a split event happened to the data of a car, yielding two trajectories 141 and 23, while after processing, the two fragment trajectories are merged successfully, and trajectory parameters get refined, e.g. the shape model is more accurate, and the trajectory line is more smooth.

Fig.10 demonstrates the processing results for group trajectories. As marked in red circles, two pedestrians with trajectories 442 and 496 approached to each other at frame 1550, crossed at frame 1600 then left. At the online tracking results, trajectories 442 and 496 got merged at frame 1550, yielding a new trajectory 576. However, when they left from each other, a split event happened yielding new trajectories again, i.e. 576 and 591. At the results after processing, it can be found that the pedestrians keep their ID during and after crossing, and their trajectories correctly represent their path and motion parameters.

However some limitations of the algorithm are also found through the experiment as marked in yellow circle in Fig.9. Due to the lack of observation (laser points), shape model parameters of the object can not be correctly estimated. The trajectory line is not smooth, which reflects that there are large vibrations in the trajectory parameters. Such kind of problems are to be solved through the future work.

Table 1 gives a count to the experiment. After trajectory processing, the number of trajectories are reduced from 6281 to 4195 through the processing of 310 group, 1033 fragment, and 1826 broken trajectories in 20 minutes at a 2.53GHz PC.

TABLE I

A COUNT TO THE EXPERIMENT

Period	07:00-07:20, a.m.
Frames	44785
Number of trajectories before processing	6281
Number of trajectories after processing	4195
Number of group trajectories	310
Number of fragment trajectories	1033
Number of broken trajectories	1826

VII. CONCLUSIONS AND FUTURE WORKS

There have been many researches on moving object detection and tracking. There are also great needs in trajectory analysis and scene modeling so that to provide higher knowledge to surveillance and ITS application for decision making. However in crowded environment, trajectory data sets obtained through online processing contain many broken, group and fragment ones, which degrades the performance in trajectory analysis and scene modeling. A trajectory processing algorithm is developed in this research on a multi-laser sensing system that was developed in our previous work. It contains a trajectory association algorithm, where an interaction graph is built to represent the relationships of trajectories; a graph-based trajectory labeling algorithm; and an EM-based trajectory parameter optimization algorithm. Experiments are conducted using the data collected at an intersection in Beijing with promising results demonstrated.

REFERENCES

[1] C. Pai, H. Tyan, Y. Liang, H. Liao, and S. Chen, "Pedestrian Detection and Tracking at Crossroads", Pattern Recognition, vol.37, no.5, pp.1025-1034, 2004.

[2] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol", IEEE Trans. Pattern Analysis and Machine Intelligence, vol.31, no.2, pp.319-336, 2009.

[3] Z. Kim, "Real Time Object Tracking based on Dynamic Feature Grouping with Background Subtraction", Proc. Computer Vision and Pattern Recognition, 2008.

[4] M. Valera1, S. A. Velastin, "Intelligent Distributed Surveillance Systems: A Review", IEEE Proc. Vision, Image and Signal Processing, vol.152, pp.192-204, 2005.

[5] T. Vu, O. Aycard, and N. Appenrodt, "Online Localization and Mapping with Moving Object Tracking in Dynamic Outdoor Environments", Proc. Intelligent Vehicle Symposium, pp.190-195, 2007.

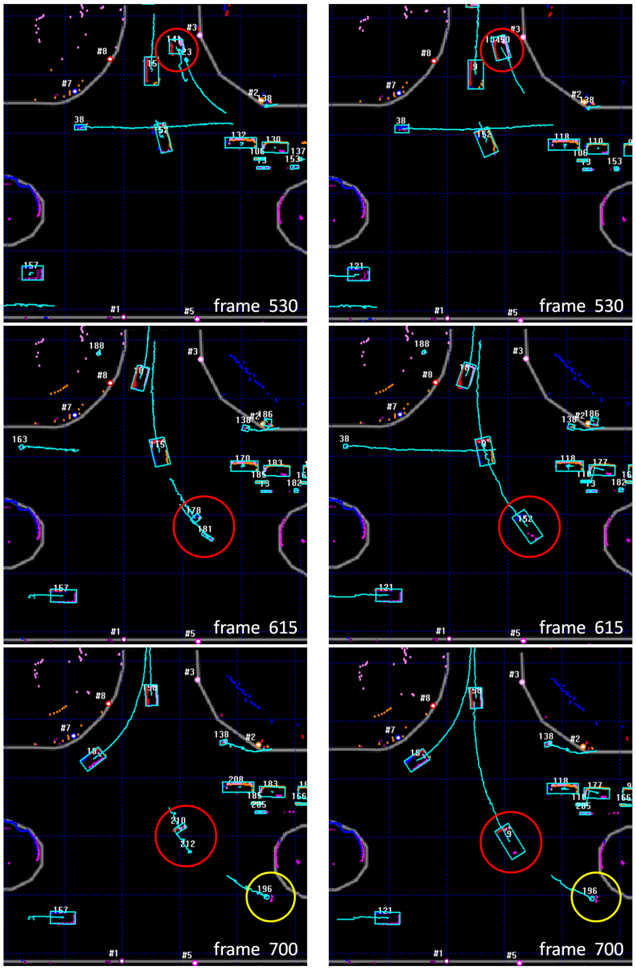
[6] J. Cui, H. Zha, H. Zhao, and R. Shibasaki, "Laser-Based Detection and Tracking of Multiple People in Crowds", Computer Vision and Image Understanding, vol.106, pp.300-312, 2007.

[7] Oh Songhwai, L. Schenato, P. Chen, and S. Sastry, "Tracking and Coordination of Multiple Agents Using Sensor Networks: System Design, Algorithms and Experiments", Proceedings of the IEEE, vol.95, pp.234-254, 2007.

[8] J. Cui, H. Zha, H. Zhao, and R. Shibasaki, "Tracking Multiple People using Laser and Vision", IEEE Int. Conf. Intelligent Robots and Systems, pp.2116-2121, 2005.

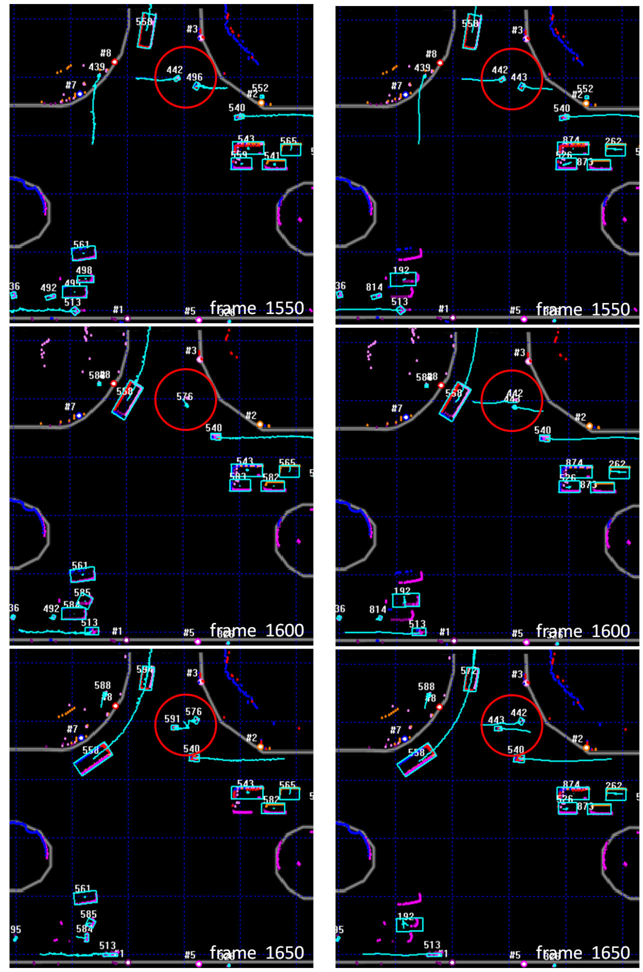
[9] D. Makris, T. Ellis, "Learning Semantic Scene Models from Observing Activity in Visual Surveillance", IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics, vol.35, pp.397-408, 2005.

[10] N. Brandle, D. Bauer, and S. Seer, "Track-Based Finding of Stopping Pedestrians C A Practical Approach for Analyzing a Public Infrastructure", Proc. Intelligent Transportation Systems, pp.115-120, 2006.



(a) Online tracking results (b) Parameter optimization results

Fig. 9. Trajectory processing results (fragment)



(a) Online tracking results (b) Parameter optimization results

Fig. 10. Trajectory processing results (group)

[11] D. Biliotti, G. Antonini, and J. P. Thiran, "Multi-Layer Hierarchical Clustering of Pedestrian Trajectories for Automatic Counting of People in Video Sequences", IEEE Workshop Motion and Video Computing, vol.2, pp.50-57, 2005.

[12] X. Wang, K. Tieu, and E. Grimson, "Learning Semantic Scene Models by Trajectory Analysis", Eur. Conf. Computer Vision, 2006.

[13] J. Melo, A. Naftel, A. Bernardino, and J. Santos-Victor, "Detection and Classification of Highway Lanes Using Vehicle Motion Trajectories", IEEE Trans. Intelligent Transportation Systems, vol.7, pp.188-200, 2006.

[14] P. Kumar, S. Ranganath, K. Sengupta, and W. Huang, "Cooperative Multitarget Tracking With Efficient Split and Merge Handling", IEEE Trans. Circuits and Systems for Video Technology, vol.16, pp.1477 - 1490, 2006.

[15] Z. Khan, T. Balch, and F. Dellaert, "Multitarget Tracking with Split and Merged Measurements", Proc. Computer Vision and Pattern Recognition, vol.1, pp.605-610, 2005.

[16] L. J. Latecki, R. Miezianko, "Object Tracking with Dynamic Template Update and Occlusion Detection", Int. Conf. Pattern Recognition, pp.556-560, 2006.

[17] A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu, "Multi-Object Tracking Through Simultaneous Long Occlusions and Split-Merge Conditions", Proc. Computer Vision and Pattern Recognition, pp.666-673, 2006.

[18] Y. Ma, Q. Yu, and I. Cohen, "Multiple Hypothesis Target Tracking Using Merge and Split of Graph's Nodes", Int. Sym. Visual Computing, vol.4292, pp.783-792, 2006.

[19] P. Nillius, J. Sullivan, and S. Carlsson, "Multi-Target Tracking-Linking Identities using Bayesian Network Inference", Proc. Computer Vision and Pattern Recognition, vol.2, pp.2187-2194, 2006.

[20] J. Sullivan, S. Carlsson, "Tracking and Labelling of Interacting Multiple Targets", Eur. Conf. on Computer Vision, pp.619–632, 2006.

[21] B. Bose, X. Wang, and E. Grimson, "Multi-Class Object Tracking Algorithm that Handles Fragmentation and Grouping", Proc. Computer Vision and Pattern Recognition, 2007.

[22] H. Zhao, J. Cui, H. Zha, K. Katabira, X. Shao, and R. Shibasaki, "Sensing an Intersection Using a Network of Laser Scanners and Video Cameras", *IEEE Intelligent Transportation Systems Magazine*, vol.1, no.2, pp.31-37, 2009.

[23] T. K. Moon, "The Expectation-Maximization Algorithm", IEEE Signal Processing Magazine, vol.13, pp.47-60, 1996.

[24] J. A. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models", *International Computer Science Institute*, vol.4, 1998.