8-21

MVA2009 IAPR Conference on Machine Vision Applications, May 20-22, 2009, Yokohama, JAPAN

Predicting Attainable Region of Vehicle Using Trajectory Clustering

Noritaka Sekiyama, Jien Kato, and Toyohide Watanabe
Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan
{sekiyama,jien,watanabe}@watanabe.ss.is.nagoya-u.ac.jp

Abstract

Some traffic monitoring systems have been proposed for predicting behaviors of vehicles. However, a few attempts have been made at estimating how vehicles influence each other at some time point in the near future. In order to estimate such interactions, we introduce a concept of Attainable Region to predict future behaviors of vehicles. Attainable Region is efficient to estimate interactions between vehicles in the near future. Through evaluation experiments, we show the feasibility of prediction with Attainable Region.

1 Introduction

Visual surveillance in the real world is an active research topic in machine vision community [1]. Many researchers have made great efforts to improve techniques in this area and many applications have been developed. Especially, traffic monitoring systems are well-known applications of visual surveillance. For example, anomaly detection [2] and collision prediction [3, 4] are generally cited as the applications. Such systems provide valuable information using knowledge about vehicle behaviors by modeling and analyzing traffic scenes. Future behaviors of vehicles are an important facet of the knowledge for the system to assist drivers. Furthermore, interactions between vehicles at some time point in the near future cannot be neglected when vehicles influence each other in traffic scenes. For example, in case two vehicles are about to collide, the interaction between them should be estimated and the drivers are alerted as soon as possible. In order to consider how vehicles influence each other in the near future, a vehicle which has an influence on a behavior of a particular vehicle should be estimated.

Atev, et al. [3] have presented a vision-based system that issues warnings about imminent collisions on the assumption that velocities of vehicles stay constant. Although this assumption is reasonable in case of short time interval for prediction, it is unreasonable in case of some measure of the time interval because velocity and acceleration of vehicle are changeable elements at each frame. Prior knowledge in the scene is essential to predict vehicle behaviors for such time interval. Saleemi, et al. [5] have proposed a method for modeling and learning the scene activity to obtain prior knowledge in the scene. Future positions of vehicles are estimated based on their velocities and prior knowledge. This approach, however, ignores size of vehicles and does not address estimating interactions between moving objects.

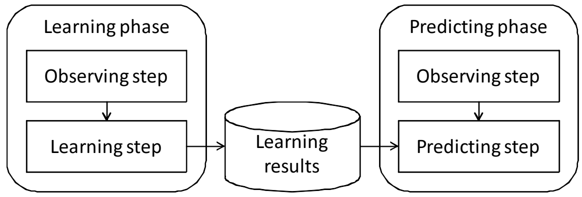

Figure 1: Our framework. The learning phase is an off-line step and the predicting phase is an on-line step.

In this paper, we focus on such interactions and propose a concept of *Attainable Region* for predicting future behaviors of vehicles. Attainable Region is defined as a region where a vehicle can attain in a few seconds. It includes multiple possibilities of vehicle behaviors which depend on prediction, and describes them as one region at one frame and one vehicle. Since possibilities of each vehicle's behavior after some frames are predicted, interactions between vehicles can be efficiently represented. With this concept, we can easily estimate the interactions by checking whether the regions overlap each other. In order to estimate Attainable Region, prior knowledge is obtained by learning the scene model. Our learning algorithm is based on clustering of vehicles' trajectories. We also employ observed information and combine it with the prior knowledge to predict Attainable Region.

A driver support system which warns about the danger of approaching vehicles can be developed with Attainable Region. This approach can be combined with a system that monitors driver's attention [6]. The work presented here is a step toward this direction.

Our framework is shown in Figure 1. In the observing step, vehicles in the scene are tracked at each frame. In the learning step, trajectories of vehicles are clustered with threshold and a representative trajectory of each cluster is calculated. In the predicting step, future behaviors of vehicles are estimated by integrating learning results with observed information.

2 Learning the Scene Model

In this section, we discuss prior knowledge of traffic scenes and propose a model which learns behavior patterns of vehicles. The prior knowledge describes a general behavior of vehicles in traffic scenes. This information can be obtained by learning time-series location trails, called trajectories, of vehicles. Hu, et al. [7] have presented an algorithm for learning trajectories using fuzzy K-means clustering. Each cluster has a centroid which means a representative trajectory. The system can detect anomaly in the scenes and predict behaviors of vehicles by using these cluster centroids. Thus, clustering trajectories is effective for modeling and analyzing traffic scenes. We also adopt a clustering method of trajectories to model scenes and obtain prior knowledge.

In our framework, input data in the learning step is previously obtained in the observing step. This observed information consists of time stamp, vehicle location, velocity and instantaneous acceleration, direction, and size. The information of vehicle i at frame n is defined as 8-dimensional vector $\boldsymbol{O_i} = (t_n, x_n, y_n, v_n, a_n, \theta_n, width, height)$. t_n is the time stamp at frame n, x_n and y_n are the local coordinates of vehicle i, v_n is the velocity, a_n is the acceleration, θ_n is the direction, and $width$ and $height$ are the size of vehicle i. Trajectory vector $\boldsymbol{T_i}$ is, then, obtained from observed vector $\boldsymbol{O_i}$ in the following definition:

$$\boldsymbol{T_i} = \{(x_b, y_b), (x_{b+1}, y_{b+1}), \cdots, (x_e, y_e)\} \tag{1}$$

Here, b is the frame index at which the vehicle i entered the scene, and e is the frame index at which the vehicle i exited the scene.

For clustering trajectories, similarities between $\boldsymbol{T_i}$ and $\boldsymbol{T_j}$ must be calculated. It is important to choose distance function between two trajectories. The Euclidean distance is widely used for measuring similarity between two time-series data. However, it cannot be applied to our problem because it can be used only if two time-series are of equal length. More generalized similarity measurements include Dynamic Time Warping (DTW), the Longest Common Subsequence (LCSS) [8], Edit Distance on Real sequences (EDR) [9], and the Sequence Weighted Alignment (Swale) [10]. Swale can achieve greater accuracy than DTW, LCSS, and EDR. Moreover, Morse, et al. [10] have presented the Fast Time Series Evaluation (FTSE) method which can be used for evaluating LCSS, EDR, and Swale quickly. Therefore, we employ Swale as distance function for clustering trajectories and FTSE as a speed-up algorithm. Swale is defined as follows:

$$Swale(R, S) = \begin{cases} n * gap & \text{if } m = 0 \\ m * gap & \text{if } n = 0 \\ rew + Swale(Rest(\boldsymbol{R}), Rest(\boldsymbol{S})) & \text{if } |r_1 - s_1| \le \epsilon \\ max\{gap + Swale(Rest(\boldsymbol{R}), \boldsymbol{S}), \\ \quad gap + Swale(\boldsymbol{R}, Rest(\boldsymbol{S}))\} & \text{otherwise} \end{cases} \tag{2}$$

Here, $\boldsymbol{R}$ is time series $\{r_1, \cdots, r_m\}$ and $\boldsymbol{S}$ is time series $\{s_1, \cdots, s_n\}$, r_i is the i-th element of $\boldsymbol{R}$. $Rest(\boldsymbol{R})$ is $\boldsymbol{R}$ with the first element removed. rew is a match reward and gap is a gap cost.

We, then, employ the group-average clustering [11]. This method enables effective clustering when there is only similarity between vehicles and cluster centroids cannot be calculated. All trajectories are clustered with Swale hierarchically based on this method. A medoid obtained in the learning step indicates a representative trajectory in each cluster and is used in the predicting step. The medoid of cluster c is defined as follows:

$$m_c = \arg\max_{i \in V_c} \sum_{j \in (V_c - \{i\})} Swale(\boldsymbol{T_i}, \boldsymbol{T_j}) \tag{3}$$

We express the vehicle index which represents medoid in cluster c as m_c. V_c is a set of vehicles included in cluster c. These medoids are the result of a learning phase and used in the predicting phase.

3 Predicting Behavior of Vehicle

3.1 Determining Attainable Region

We propose a method for combining prior knowledge with observed information to determine Attainable Region. In the predicting step, Attainable Region is determined based on cluster medoids which are obtained in the learning step and observed vectors described in the previous section. Once a target vehicle is determined, a partial trajectory $\boldsymbol{P}$ of the vehicle from frame b to frame $curr$ is obtained in Equation (4).

$$\boldsymbol{P} = \{(x_b, y_b), (x_{b+1}, y_{b+1}), \cdots, (x_{curr}, y_{curr})\} \tag{4}$$

We define a current frame index at which prediction begins as $curr$ and a frame index which is $curr$ minus the minimum number of frames required for prediction as b.

The partial trajectory $\boldsymbol{P}$ is, then, compared with all cluster medoids. Only medoids whose similarity to $\boldsymbol{P}$ exceeds a threshold are selected as candidates of future trajectory. In the predicting step, the point is not dissimilarity but similar segments between two sequences. Figure 2 shows the reason why similar segments are important. When two sequences are compared, one is an incomplete trajectory and the other is a complete trajectory. If we employ $Swale$, dissimilar segments between two sequences gets gap cost and a value of $Swale$ becomes larger improperly. The number of similar segments between two sequences should be counted up. LCSS is, therefore, a better choice than DTW, EDR, and Swale. LCSS is defined in Equation (2).

$$LCSS(\boldsymbol{R}, \boldsymbol{S}) = \begin{cases} 0 & \text{if } m = 0 \text{ } or \text{ } n = 0 \\ LCSS(Rest(\boldsymbol{R}), Rest(\boldsymbol{S})) + 1 & \text{if } |r_1 - s_1| \le \epsilon \\ max\{LCSS(Rest(\boldsymbol{R}), \boldsymbol{S}), \\ \quad LCSS(\boldsymbol{R}, Rest(\boldsymbol{S}))\} & \text{otherwise} \end{cases} \tag{5}$$

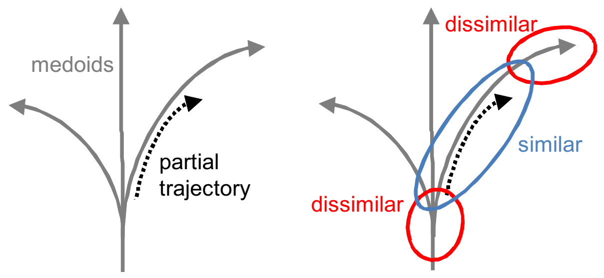

Figure 2: Comparing a partial trajectory and medoids.

Algorithm 1 Build Rectangle List

Ensure: *Rectangle List L*

$i \leftarrow curr$;
$j \leftarrow \arg\min\ dist(p_i, c_j)$;
$move \leftarrow 0$;
$\epsilon \leftarrow v_{curr}t + \frac{1}{2}a_{curr}t^2$;
while $proceed < \epsilon$ **do**
  $move \leftarrow c_{j+1} - c_j$;
  $p_{i+1} \leftarrow p_i + move$;
  $proceed \leftarrow proceed + move$;
  Obtain $rect_{i+1}$ from p_{i+1}, $width$, $height$, and θ_n;
  Insert $rect_{i+1}$ into L;
  $i \leftarrow i + 1$;
  $j \leftarrow j + 1$;
end while

An index set C_P of the candidate trajectories is given in Equation (6).

$$C_P = \{m_c \mid LCSS(\boldsymbol{P}, \boldsymbol{T_{m_c}}) > threshold\} \quad (6)$$

Note that multiple candidate trajectories can be obtained. They represent possibilities of vehicle behaviors which depends on prediction. The candidate trajectories are used for determining future moving direction of vehicles. Rectangle lists, which represent occupied regions of vehicles at future frames, can be calculated based on such trajectories. The technique used to obtain a rectangle list of a vehicle is shown in Algorithm 1. $\boldsymbol{P}$ is the partial trajectory of the target vehicle, $\boldsymbol{C}$ is the candidate trajectory, p_i is the i-th element included in $\boldsymbol{P}$, c_j is the j-th element included in $\boldsymbol{C}$, $curr$ is the frame index at which the current element of the target vehicle is, ε is a distance threshold, t is the time interval for prediction, θ_n is the direction, and $width$ and $height$ are the size of vehicle. In this algorithm, $dist(a,b) = \sqrt{(b_x - a_x)^2 + (b_y - a_y)^2}$. Attainable Region is defined as a set of rectangle lists along the candidate trajectories.

3.2 Data structure for Attainable Region

We introduce a three-layered data structure based on R*-tree [12] to represent Attainable Region. R*-tree is an efficient data structure for spatial indexing. Figure 3 represents this data structure. In the first layer, ROI is defined as a bounding box of Attainable Region of each vehicle. Every ROI of vehicle is stored as R*-tree. By checking overlaps between these ROIs, two vehicles which may influence each other in the near future can be estimated. In the second layer, ROI is defined as a bounding box of each occupied region of which Attainable Region is composed. A bounding box of an occupied region of Attainable Regions in which overlaps are detected is stored as ROI. By checking these ROI, segments which may overlap each other between two Attainable Regions can be found. In the third layer, two occupied regions whose bounding boxes overlap each other in the second layer are stored. The system checks overlaps between them.

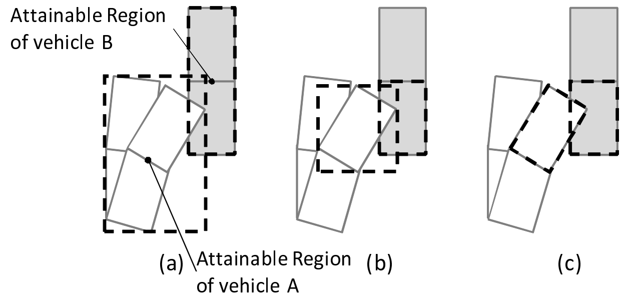

Figure 3: Three-layered data structure. Attainable Regions of vehicle A and vehicle B are described. Dashed line represents ROI. (a) is the first layer, (b) is the second layer, and (c) is the third layer.

This data structure enables to check overlaps between Attainable Regions quickly because unnecessary comparisons are omitted.

4 Experimental Results

We present experimental results regarding the feasibility of prediction based on Attainable Region. In this experiment, we use the Next Generation Simulation (NGSIM) [13] trajectory data sets. The data represents traffic flows on a segment of Lankershim Boulevard in the Universal City neighborhood of Los Angeles, California in 2005.

In the learning phase, we used the data set between 8:30 am and 8:45 am as observed information and trajectories of 1210 vehicles were learned. In the predicting phase, we used the data set between 8:45 am and 9:00 am as observed information. 100 vehicles were selected arbitrarily as the target vehicles. In our experiments, we express the time interval for prediction as Δt. The prediction result was compared with the situation after a few seconds from the current frame. This time stamp is expressed as t_f.

An area ratio of the actual occupied region in predicted Attainable Region is an important factor. If whole of the actual occupied region of the vehicle is included in the computed Attainable Region, the area ratio is 1. Furthermore, in order to focus on the possibility that an accurate Attainable Region is estimated by prediction, we define recall as a fraction of the num-

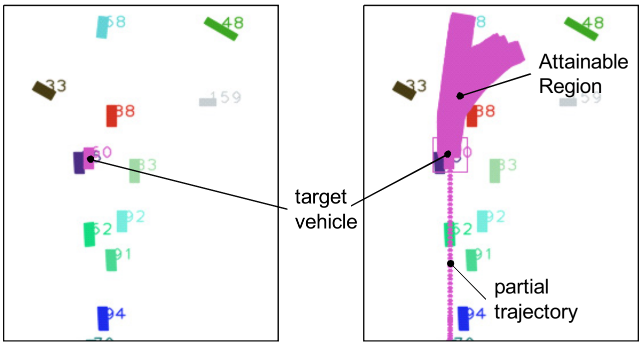

Figure 4: Bird's-eye view in the scene. Rectangles indicate vehicles and the number described around each vehicle indicates a vehicle index. The left figure represents a traffic situation at frame 696. The right figure means an image overlapped with Attainable Region.

Table 1: The average area ratio and recall.

$\Delta t(sec)$	$t_f(sec)$	Average area ratio	Recall
2	1	0.795	0.79
3	2	0.683	0.68
4	3	0.613	0.59
5	4	0.551	0.54

ber of accurate Attainable Region N_R in the number of actual occupied regions after time interval for prediction N_C. We estimated that the prediction is accurate if the area ratio was greater than 0.7.

A successful example of experimental results is shown in Figure 4. In this case, Δt was 3 seconds, and t_f was 2 seconds. The area ratio was 0.94. Our approach achieved sufficient accuracy and the computed Attainable Region enabled estimating the future behavior of the vehicle accurately.

Next, we show the prediction performance of 100 arbitrary vehicles in some tuples of Δt and t_f. The average area ratio and recall are shown in Table 1. Our approach had the best performance when Δt was 2 and t_f was 1. Most of the Attainable Regions of vehicles were successfully estimated in this case. In case that Δt was 5 and t_f was 4, both the area ratio and recall was worse than those in case that Δt was 2 and t_f was 1. The reason is that it is difficult to predict correct behavior when the time interval for prediction is long. The shorter the value of time interval for prediction was set to be, the greater the area ratio and recall became. Especially, there were some cases that the area ratio was estimated to be 0. The number of these cases was higher than the number of the cases of 0.1, 0.2, and so on. Since the case of 0 means complete failure, we must consider what caused it. One of the reasons for prediction failure is that our approach does not consider the behavior of "stop". Another reason is that similarity between partial trajectory and candidate trajectories was not appropriately calculated. We must address these problems and improve our approach.

5 Conclusion

In this paper, we have proposed the concept of Attainable Region to predict future behaviors of vehicles. Furthermore, we have proposed a three-layered data structure based on R*-tree to represent Attainable Region. Our approach is efficient to estimate interactions between vehicles. Experimental results have shown the feasibility of prediction with Attainable Region.

In our future work, we apply Attainable Region to various moving objects such as pedestrians, bicycles, and motorbikes. Although reduction of a computational cost is not addressed in this paper, we must address it and keep the cost to a minimum. In addition, we should improve accuracy and effectiveness for prediction and estimating interactions between objects.

Acknowledgments

This research is supported in part by International Communications Foundation (ICF).

References

[1] Weiming Hu, et al.: "A Survey on Visual Surveillance of Object Motion and Behaviors", *IEEE Trans. SMC*, vol.34, no.3, pp.334-352, 2004.

[2] Arslan Basharat, et al.: "Learning Object Motion Patterns for Anomaly Detection and Improved Object Detection", *IEEE Conf. CVPR*, pp.1-8, 2008.

[3] Stefan Atev, et al.: "A Vision-Based Approach to Collision Prediction at Traffic Intersections", *IEEE Trans. ITS*, vol.6, no.4, pp.416-423, 2005.

[4] Nicolas Saunier, et al.: "Probabilistic Collision Prediction for Vision-Based Automated Road Safety Analysis", *IEEE Conf. ITS*, pp.872-878, 2007.

[5] Imran Saleemi, et al.: "Probabilistic Modeling of Scene Dynamics for Applications in Visual Surveillance", *IEEE Trans. PAMI*, Accepted for future publication.

[6] Luis M. Bergasa, et al.: "Real-Time System for Monitoring Driver Vigilance", *IEEE Trans. ITS*, vol.7, no.1, pp.63-77, 2006.

[7] Weiming Hu, et al.: "A System for Learning Statistical Motion Patterns", *IEEE Trans. PAMI*, vol.28, no.9, pp.1450-1464, 2006.

[8] Michail Vlachos, et al.: "Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures", *ACM SIGKDD*, pp.216-225, 2003.

[9] Lei Chen, et al.: "Robust and Fast Similarity Search for Moving Object Trajectories", *ACM SIGMOD*, pp.491-502, 2005.

[10] Michael D. Morse, et al.: "An Efficient and Accurate Method for Evaluating Time Series Similarity", *ACM SIGMOD*, pp.569-580, 2007.

[11] Sepandar D. Kamvar, et al.: "Interpreting and Extending Classical Agglomerative Clustering Algorithms using a Model-Based Approach", *International Conference on Machine Learning*, pp.283-290, 2002.

[12] Norbert Beckmann, et al.: "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", *ACM SIGMOD*, pp.322-331, 1990.

[13] Next Generation Simulation (NGSIM): http://ngsim.fhwa.dot.gov/