

Probabilistic Collision Prediction for Vision-Based Automated Road Safety Analysis

Nicolas Saunier, Tarek Sayed and Clark Lim

Abstract—This work aims at addressing the many problems that have hindered the development of vision-based systems for automated road safety analysis. The approach relies on traffic conflicts used as surrogates for collision data. Traffic conflicts are identified by computing the collision probability for any two road users in an interaction. A complete system is implemented to process traffic video data, detect and track road users, and analyze their interactions. Motion patterns are needed to predict road users' movements and determine their probability of being involved in a collision. An original incremental algorithm for the learning of prototype trajectories as motion patterns is presented. The system is tested on real world traffic data, including a few traffic conflict instances. Traffic patterns are successfully learnt on two datasets, and used for collision probability computation and traffic conflict detection.

I. INTRODUCTION

Traffic safety is one of the major world health problems. According to the World Health Organization, 1.2 million people were killed in road traffic crashes in 2002, and between 20 millions and 50 million were injured [1]. Traffic safety diagnosis has been traditionally undertaken using historical collision data. However, there are well-recognized problems of availability and quality associated with collision data. In many jurisdictions, the quantity and quality of collision data has been degrading for several years. Additionally, the use of collision records for safety analysis is a reactive approach: a significant number of collisions have to be recorded before action is taken. Because of these problems, the observation of traffic conflicts has been advocated as an alternative or complementary approach to analyze traffic safety from a broader perspective than collision statistics alone [2], [3], [4], [5], [6], [7]. Traffic Conflict Techniques (TCTs) involve observing and evaluating the frequency and severity of traffic conflicts at an intersection by a team of trained observers. A conflict is defined as “an observational situation in which two or more road users approach each other in space and time to such an extent that a collision is imminent if their movements remain unchanged” [8]. While the monitoring of the traffic conflicts occurring in a given location for a few hours is sufficient to assess its safety, the main drawbacks of TCTs are the data collection costs, and the subjectivity and reliability of observers.

Automated systems are needed to address these issues which hinder the wider use of TCTs. Some of the most

promising approaches rely on video sensors and intelligent techniques to interpret video data, including computer vision and machine learning. Vision-based systems for traffic monitoring would reduce the workload of human operators, help improve our understanding of traffic behaviour and further address the many problems that plague the road networks, such as congestion and collisions. Video sensors for traffic monitoring have a number of advantages, among which are the ease of installation, the possibility to get rich traffic description, and the scope of the areas covered by a camera.

This work aims at building a complete system for automated road safety analysis, by detecting traffic conflicts in video data. To detect traffic conflicts, the probability of collision for any two interacting road users can be computed using definitions and techniques adapted from [9]. This computation requires the learning of typical motion patterns to predict future positions and the occurrence of collision. This work presents an original incremental algorithm for the learning of motion patterns for motion prediction, with distinct advantages over the offline method of [9]. The next section of this paper discusses traffic conflicts and severity indicators, and provides computational definitions in a probabilistic framework that can be used in an automated system. The third section describes the implementation of a vision-based system that automatically detects traffic conflicts in intersections, using the definitions provided in the first section. The fourth section describes an original algorithm for motion pattern learning. Experimental results on real world video data are presented in the last section. Related work is introduced in each part.

II. A COMPUTATIONAL PERSPECTIVE ON TRAFFIC CONFLICTS

A. Traffic Conflicts and Severity Indicators

The concept of traffic conflicts was first proposed by Perkins and Harris in 1968 [10] as an alternative to collision data, which in many cases were scarce, unreliable, or unsatisfactory. Their objective was to define traffic events or incidences that occur frequently, can be clearly observed, and are related to collisions. The widely used definition of traffic conflict given in the introduction highlights the importance of the collision course. Users are defined to be on a collision course when, “unless the speed and/or the direction of the road users changes, they will collide” [6]. Deciding if two road users are on a collision course thus depends on extrapolation hypotheses. The definition of [6] uses the common hypothesis of extrapolation with constant velocity, i.e. speed and direction. Some definitions of traffic

Nicolas Saunier, Tarek Sayed and Clark Lim are with the Department of Civil Engineering, University of British Columbia, 6250 Applied Science Lane, Vancouver, BC, V6T1Z4, Canada {saunier,tsayed}@civil.ubc.ca, Clark.Lim@translink.bc.ca

conflicts also include that at least one of the road users involved takes an evasive action, often in emergency.

Theories about traffic describe the relationship between traffic conflicts and collision, which must be established to use traffic conflicts as surrogates to collisions for safety analysis. Many researchers, especially in Scandinavian countries [2], [6], assume that all interactions can be ranked in a safety hierarchy, with collisions at the top. An interaction is defined as an observational situation in which two or more road users are close enough in space and time. The interactions located next to the collisions in the safety hierarchy are often called quasi-collisions. The interactions can thus be recursively ranked in the safety hierarchy (See Figure 1). For this concept to be operational, the safety hierarchy is transferred into measurable parameters based on certain assumptions. For each interaction in the hierarchy, a severity can be estimated, matching its location in the hierarchy, i.e. measuring the proximity to the potential occurrence of a collision, which is related to the probability of collision. Many severity indicators have been developed (See for example [3] for extensive reviews and discussions), e.g. the Time-To-Collision (TTC), defined for two road users on a collision course as the extrapolated time for the collision to occur, or the Post-Encroachment Time (PET), defined as the time measured from the moment the first road user leaves the potential collision point to the moment the other road user enters this point.

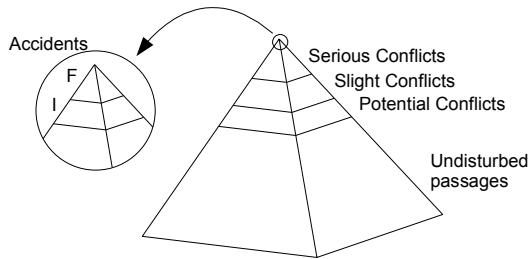


Fig. 1. The safety hierarchy, as presented in [2].

Traffic conflicts are interactions between road users on a collision course. When road users are in an interaction, various chains of events can lead to a collision, as opposed to one path for each road user with constant velocity [6]. The collision course is not a binary concept. It can be properly analyzed in a probabilistic framework by taking into account all possible movements of the road users to compute the collision probability. The collision probability can be considered as the normalized severity dimension of the safety hierarchy. When road users do not have the physical possibility to avoid a collision, the collision will occur, i.e. the probability of collision is 1.

B. How to Compute the Collision Probability

The formulas presented in this part are based on [9], and to a lesser extent on [11]. The collision probability for a given interaction between two road users can be computed at a given instant by summing the collision probability over all

possible motions that lead to a collision, given the road users' states. This requires the ability to generate for each road user at any instant a distribution over its possible future positions given its previous positions. A possible future motion, i.e. a temporal series of predicted positions, defines an extrapolation hypothesis. The collision probability computation is approximated by a discrete sum when taking into account a finite number of the most probable extrapolation hypotheses.

First the collision probability at time t_0 for two road users A_1 and A_2 with respective observed trajectories $Q_{1,t \leq t_0}$ and $Q_{2,t \leq t_0}$ (before t_0) is defined when considering only one extrapolation hypothesis for each, respectively H_i and H_j . The predicted positions according to the hypotheses H_i and H_j are computed for a number of time steps: the predicted time of the collision $t_{i,j}$ is the first instant at which the road users would be in contact. The larger $\Delta_{i,j} = t_{i,j} - t_0$, the more likely the road users can react and avoid the collision. This time takes into account speed and distance and is directly measurable against the road users' reaction times. The formula of the probability of collision given hypotheses H_i and H_j is taken from [9]

$$P(\text{Collision}(A_1, A_2)|H_i, H_j) = e^{-\frac{\Delta_{i,j}^2}{2\sigma^2}} \quad (1)$$

where σ is a normalizing constant. It is estimated in [9] that this probability should change when the elapsed time $\Delta_{i,j}$ is close to the road user reaction time. Therefore σ is chosen to be equal to an average user reaction time¹. Based on [9], the collision probability for two road users A_1 and A_2 at t_0 is

$$P(\text{Collision}(A_1, A_2)|Q_{1,t \leq t_0}, Q_{2,t \leq t_0}) = \sum_{i,j} P(H_i|Q_{1,t \leq t_0})P(H_j|Q_{2,t \leq t_0}) e^{-\frac{\Delta_{i,j}^2}{2\sigma^2}} \quad (2)$$

where $P(H_i|Q_{1,t \leq t_0})$ is the probability of road user A_1 to move according to extrapolation hypothesis H_i (same for A_2 and H_j). The sum is done over a variety of extrapolation hypotheses, although this number must be limited to maintain reasonable computation times. This formula is illustrated in a simplified example in Figure 2. The expected values of traditional severity indicators such as the TTC can be introduced in this probabilistic framework. In a traditional TCT, one could choose a threshold on collision probability and other indicators to define traffic conflicts. In the new approach described in this paper, road safety can be automatically analyzed in detail by computing continuously the collision probability of all interactions. This allows detailed traffic and road safety analysis by taking into account interactions of all severity, similarly to the pioneering work of [6]. Detailed exposure measurements can thus be obtained for use in many traffic engineering applications. The next sections describe the implementation of a vision-based system that makes use of these formulas, including a method to learn the motion

¹A value of 1.5 seconds is chosen for the experiments described in this paper.

patterns of road users from traffic data in order to generate extrapolation hypotheses. The probabilities of extrapolation hypotheses are also automatically learnt.

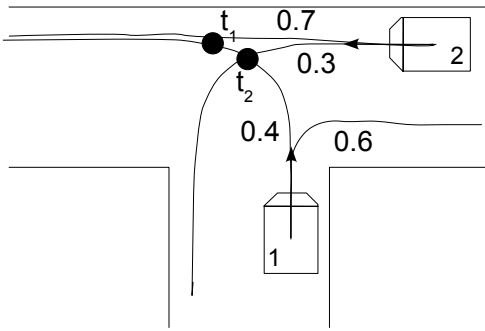


Fig. 2. In this simplified situation, two vehicles approach a T intersection at time t_0 . Only two extrapolation hypotheses are considered for each vehicle. Vehicle 1 is expected to turn left or right, with respective probabilities 0.4 and 0.6. Vehicle 2 is expected to go straight or turn left, with respective probabilities 0.7 and 0.3. There are two potential collision points, that can happen at times t_1 and t_2 . The collision probability at time t_0 is computed as $P(\text{Collision}) = 0.4 \times 0.7 \times e^{-\frac{(t_1-t_0)^2}{2\sigma^2}} + 0.4 \times 0.3 \times e^{-\frac{(t_2-t_0)^2}{2\sigma^2}}$.

III. OVERVIEW OF A VISION-BASED SYSTEM FOR AUTOMATED ROAD SAFETY ANALYSIS

Despite the potential benefits of automated traffic safety analysis based on video sensors, limited computer vision research has been directly applied to road safety [12], [9], [11], [13], [14], and even less so to the detection of traffic conflicts. Maurin et al. state in [15] that “despite significant advances in traffic sensors and algorithms, modern monitoring systems cannot effectively handle busy intersections”. Such a system requires a high level understanding of the scene and is traditionally composed of two levels of modules (see Figure 3):

- 1) a video processing module for road user detection and tracking,
- 2) interpretation modules for traffic conflict detection.

For road safety applications, our approach relies on the building of two databases: a trajectory database, where the results of the video processing module are stored, and an interaction database, where all interactions between road users within a given distance are considered, and for which various indicators, including collision probability and other severity indicators, are automatically computed. Identifying traffic conflicts and measuring other traffic parameters becomes the problem of mining these databases.

The road user detection and tracking module used in the system described in this paper relies on a feature-based tracking method that extends to intersections the method described in [16]. In this approach, distinguishable points or lines in the image are tracked: a moving object may have multiple features, which must be grouped for each object. A detailed description of the tracking algorithm is

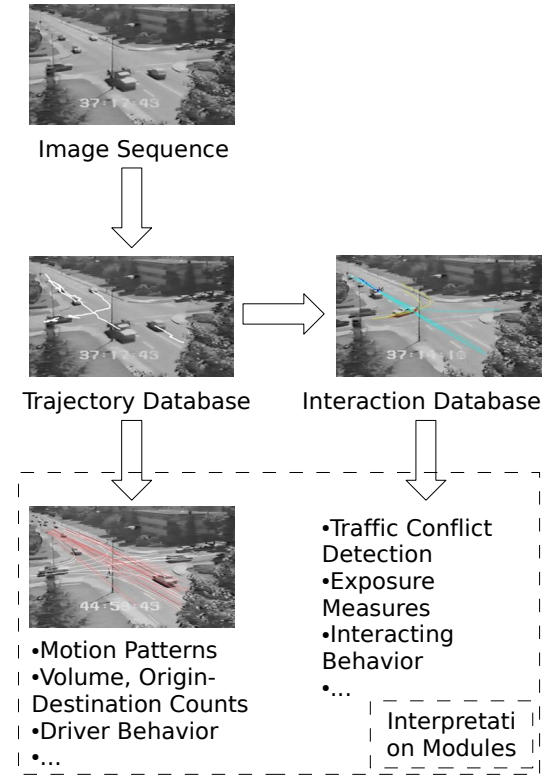


Fig. 3. Overview of a modular system for vision-based automated road safety analysis.

presented in [17]. The tracking accuracy for motor vehicles has been measured between 84.7% and 94.4% on three different sets of sequences (pedestrians and two-wheels may also be tracked, but less reliably). This means that most trajectories are detected by the system, although overgrouping and oversegmentation still happens and creates some problems. The most important limitation for traffic conflict detection is the inaccuracy in the estimation of road users’ sizes. Because of this inaccuracy, the center of each group of features is currently used for each road user, and a threshold on distances $D_{collision}$ is used to determine a potential future collision (to determine the predicted collision time used in Equation (1)).

IV. MOTION PATTERN LEARNING FOR MOTION PREDICTION

As stated in the previous section, our approach requires the ability to generate for each road user at any instant a distribution over its possible future positions given its previous positions. A large number of outcomes are possible. However, road users do not move randomly. Instead of using default extrapolation hypotheses, knowledge about the typical road user motions can be used, e.g. the possible turns in an intersection. Regular typical movements, called motion patterns, in a given location can be learnt from a given set of observed trajectories, in order to propose more realistic and accurate motion prediction.

A. Related Work

Similarly to trajectory clustering algorithms [18], a method to learn motion patterns must address three problems:

- choose a suitable data representation of motion patterns,
- define a distance or similarity measure between trajectories or between trajectories and motion patterns,
- define a method to update the motion patterns.

Although this is a fairly new research area, significant work has already been done. A good overview can be found in [19]. In [20], the probability density functions of object trajectories generated from image sequences are learnt using self organizing neural networks. Movement is described as a sequence of flow vectors, i.e. four-dimensional vectors consisting of the object position and velocity. Such methods require long learning processes and were considered to be ill-suited for motion prediction. An improved self organizing map is used in [9]. Without justification, the authors later abandoned this approach in favor of the fuzzy K-means algorithm in [19]. Collision probability estimation is presented only in [9].

The unsupervised approach presented in [21] relies on an online quantification of the vector representations of tracked moving objects, considered in a set without temporal information. A hierarchical classification is done on the accumulated co-occurrence in the trajectories, which yields interpretable clusters of activities. It is not clear how motion prediction can be achieved. Similarly, a semantic scene model is learnt by trajectory analysis and clustering in [22], which allows to detect abnormal activities.

In [23], laser range data collected indoor is clustered using sequences of Gaussian distributions with a fixed standard deviation. A hidden Markov model (HMM) is derived for movement prediction. A similar cluster model is used in [18], where clusters are organized in a tree-like structure that, when augmented with probability information, can be used for behaviour analysis, e.g. anomalous events. This is one of the few works focusing on incremental learning for online use. Path models are learnt in [24] to identify and analyze entry/exit/junction zones and routes. Many trajectory clustering algorithms rely on HMM models [25], [26], [27]. Various similarities or distance measures between trajectories have been used, from the Euclidean distance [28], to dynamic time warping (DTW), the longest common sub-sequence similarity (LCSS) [29], [30], and distances derived from the Hausdorff distance [31]. Others advocate indirect sequence clustering, using an intermediate space to represent the trajectories, such as the Fourier coefficients [32].

B. An Incremental Algorithm to Learn Trajectory Prototypes

The choices of all three elements, a suitable data representation for motion patterns, a distance, and a method to update motion patterns, depend on each other. To accommodate many learning algorithms, the trajectories must often be pre-processed, e.g. re-sampled (by linear interpolation) or padded with default values (repeating the last position or extrapolating the last position with constant direction and

speed) [9], [23]. Such pre-processing is detrimental for our application as it discards velocity information or distorts the data. Methods that require such pre-processing are therefore avoided. Non-destructive pre-processing such as smoothing can be employed if needed.

Since no pre-processing is done, one needs methods that can naturally handle variable length sequential data. Trajectories obtained from video data are also noisy, they don't start in the same areas and can be truncated or cut into multiple sub-trajectories because of tracking errors and occlusion. Indirect sequence clustering is inherently unsuited for the representation of highly complex trajectories. Various similarity measures for sequential data have been proposed in the literature. Distances based on the Euclidean distance are reviewed in [33] and found too simple to accommodate noisy and partial trajectories. The edit distance has such advantages. Primarily used for nominal sequences, it has been extended for numerical sequences in various ways, such as DTW, LCSS and the Edit Distance on Real sequences [34]. It is argued in [30] that LCSS is less sensitive to noise than other sequence similarity methods, as some sequence elements can be unmatched (which is very useful for outliers). The intuitive idea of the LCSS is to match two sequences by allowing them to stretch, without rearranging the sequence of the elements, but allowing some elements to be unmatched.

A trajectory is noted $Q_i = \{q_{i,1}, \dots, q_{i,n}\}$ where $q_{i,k} = (x_{i,k}, y_{i,k})$ are the object position coordinates². Let $Head(Q_i)$ be the sequence $\{q_{i,1}, \dots, q_{i,n-1}\}$. Given a real positive number ϵ , the LCSS similarity of two trajectories Q_i and Q_j of respective lengths m and n , $LCSS_\epsilon(Q_i, Q_j)$ is defined as

- 0 if $m = 0$ or $n = 0$,
- $1 + LCSS_\epsilon(Head(Q_i), Head(Q_j))$ if the points $q_{i,n}$ and $q_{j,m}$ match,
- $\max(LCSS_\epsilon(Head(Q_i), Q_j), LCSS_\epsilon(Q_i, Head(Q_j)))$ otherwise.

Two points q_{i,k_1} and q_{j,k_2} match if $|x_{i,k_1} - x_{j,k_2}| < \epsilon$ and $|y_{i,k_1} - y_{j,k_2}| < \epsilon$ (ϵ is the matching threshold). Other conditions can be added to enforce similarity between the trajectories, e.g. on the velocity or object size. A parameter β is added in [30] to controls how far in time it can go in order to match a given point from one trajectory to a point in another trajectory. This is not used in this work because a trajectory and its truncated sub-trajectory (e.g. truncated at the beginning by more than β points) will not always be similar with such a definition, which is crucial to accommodate trajectories reconstituted from video data. To be independent of the trajectory lengths, $LCSS_\epsilon(Q_i, Q_j)$ is normalized by the minimum length of Q_i and Q_j , and a distance is defined as $DLCSS_\epsilon(Q_i, Q_j) = 1 - \left(\frac{LCSS_\epsilon(Q_i, Q_j)}{\min(n, m)}\right)$. The LCSS can be computed by a dynamic programming algorithm in $O(nm)$. Using trajectory bounds avoids unnecessary computations if trajectories are

²Coordinates are in either the image or the world two-dimensional space (if homography information is available). Other features such as size can be added.

too far away.

Distances between trajectories are computed using the LCSS distance. There are many similarity-based clustering algorithm, such as spectral clustering methods. However, clustering is not a practical approach for motion prediction. The clusters need to be aggregated, summarized in some way that can be readily used for motion prediction. There is no easy way of “mixing”, or averaging, a set of trajectories, even if they are similar. The idea presented here is to use trajectories of the learning set, without modifying them, and to update the prototypes by keeping the longest trajectories, as they will be the most useful for accurate motion prediction (See Algorithm 1).

Input: A set of trajectories $Q = \{Q_i\}$, the allowed matching distance ϵ in the LCSS similarity definition, and the maximum LCSS distance δ for two trajectories to match ($0 \leq \delta \leq 1$).

Output: A set of prototype trajectories $P = \{P_j\}$.

for all Trajectory Q_i **do**

for all Prototype P_j in P **do**

Compute $DLCSS_\epsilon(Q_i, P_j)$.

if $DLCSS_\epsilon(Q_i, P_j) < \delta$ AND P_j is shorter than Q_i **then**

P_j is removed from P .

if Q_i didn't match any prototype OR Q_i matched at least one shorter prototype **then**

Q_i is added to P .

Algorithm 1: Algorithm for the learning of trajectory prototypes.

In this algorithm, the number of motion patterns is not required, and the parameters are limited to the allowed matching distance ϵ in the LCSS similarity definition and the maximum LCSS distance δ for two trajectories to match ($0 \leq \delta \leq 1$). ϵ controls the granularity and the number of learnt motion patterns, and must be tuned depending on the intersection and the application. More motion patterns will entail a higher computational cost for motion prediction, but also offer higher resolution and accuracy. δ has a low value, typically between 0.05 and 0.1, to allow for very limited mismatch between trajectories.

In an online situation, the trajectories are processed as they become available, and will thus adapt to changing traffic patterns. In an offline situation, the set of training trajectories Q can be randomly accessed. There is no need to process all trajectories systematically to learn the motion patterns. In such an unsupervised task, large amounts of data are available, and one can assume that in the long run, trajectories representing all motion patterns will be considered in the learning process. Therefore, one can afford to be cautious and select the trajectories to use for motion learning, as it is done in [31]. Regularity conditions can be used to avoid trajectories resulting from tracking errors. In the system presented here, feature tracks with unrealistic abrupt turns and large accelerations are discarded in the road user detection and tracking module, and only a minimal

length is subsequently enforced. Furthermore the reconstituted road user trajectories are the result of averaging various feature tracks disrupted at different times. Therefore, they are noisier and less numerous than feature tracks, which are used instead as input to the algorithm. During the learning process, which can run online continuously, the number of matched trajectories is stored for each prototype, which allows the computation of probabilities of extrapolation hypotheses, required to compute collision probabilities.

C. Using the Learnt Motion Patterns for Collision Probability Computation

Given a set $P = \{P_j\}$ of prototype trajectories representing motion patterns, including matching counts for each motion pattern, the collision probability is computed for each interaction at time t_0 . The extrapolation hypotheses are obtained by matching the trajectory Q_i of each object A_i with the prototypes (using a maximum matching LCSS distance δ). The matched prototypes are translated to a point on the object (e.g. the object center) and re-sampled using the speed of the object: this provides a set of predicted positions. An extrapolation hypothesis H_j is determined by a matched prototype, a road user point and a speed. A variety of hypotheses for each prototype can be obtained by using varied points and speeds. Equation (2) is computed by summing over all extrapolation hypotheses. Currently, only the road user position and speed measured by the tracking module are used as this requires already significant computation.

V. EXPERIMENTAL RESULTS

The core architecture of the system has been implemented, using the Intel OpenCV library³. The road user detection and tracking module presented in [17] processes the video data and extracts the feature and road user trajectories. On the contrary to [9] which uses toy cars, the present work is tested on real traffic video data, and a few traffic conflict instances identified by trained traffic conflict observers⁴. Two sets of data are used. The first is a set of traffic sequences on the same location initially used for the training of traffic conflict observers in the 1980s. Their length ranges from 10 seconds to 60 seconds. Despite the videotape age, the approximate alignment of the field of view between sequences, and occasional camera jitter, it could be digitized and used to test our method. This “Conflict” set contains 2941 feature trajectories of a minimum length of 40 frames, and 327 road user trajectories. The second dataset is a long sequence, close to one hour, recorded at an intersection in the Twin Cities, in Minnesota⁵. This “Minnesota” set contains 47084 feature tracks of a minimum length of 40 frames, and 6242 road user trajectories.

³<http://sourceforge.net/projects/opencvlibrary/>

⁴Additional experimental results are available at the address <http://www.confins.net/saunier/data/saunier07itsc.html>.

⁵The authors gratefully acknowledge Stefan Atev from the University of Minnesota in the Twin Cities who provided us with the video sequences of the Minnesota dataset.

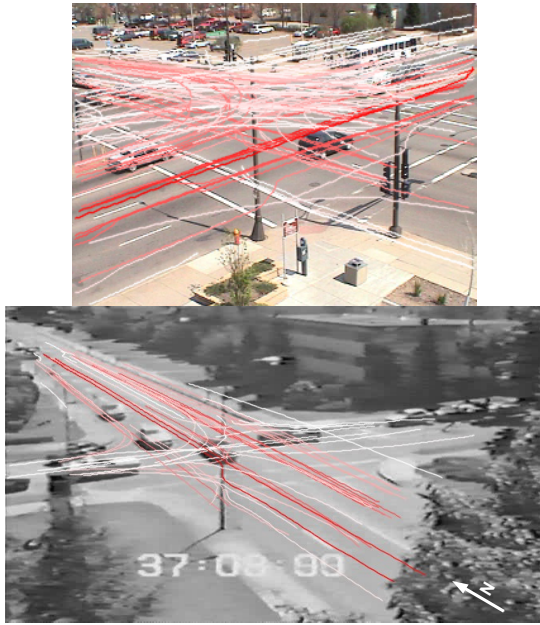


Fig. 4. Prototype trajectories learnt respectively on the Minnesota sequence (top) and the ten sequences for the training of traffic conflict observers (bottom), resulting in respectively 128 and 58 prototype trajectories. The tracks are displayed in color, from white to red indicating the number of matched trajectories in the sequence for each pattern, i.e. the traffic volume along these patterns.

First the motion patterns are learnt from the feature trajectories using Algorithm 1, which are smoothed using a Kalman filter beforehand. It is difficult to evaluate such an unsupervised task. The learnt prototypes for the two datasets are presented in Figure 4. The visual examination of the motion patterns suggests a plausible division of the trajectory space. Traffic patterns are well identified, and the traffic volumes are consistent with observation.

The results should be analyzed with respect to the application: motion prediction for traffic safety analysis. Since only a few traffic conflict instances are available in the Conflict dataset, only preliminary results obtained for the three detectable traffic conflict instances are reported in this paper. Traffic conflicts involving two wheels cannot be studied as their trajectories are not reliably detected because of the video data quality. It appears that the prototype trajectories are well suited for the computation of the collision probability. An example of movement prediction is presented for one conflict in Figure 5. The curves of the computed collision probability as a function of time, for the three traffic conflicts, are displayed in Figure 6. For each of these instances, one vehicle is over-segmented, resulting in two trajectories, and thus two traffic events (and two curves). It appears that the collision probability shows an expected evolution over time, starting with low values, increasing until the probability of collision reaches a maximum, to decrease afterward, often truncated due to tracking errors and disrupted trajectories.

Over-segmentation of tracked road users can cause major problems. The same road user detected twice can entail the

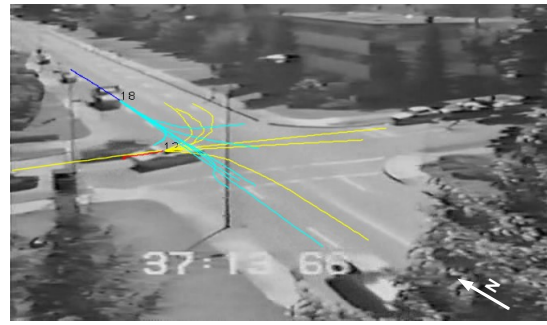


Fig. 5. An example of movement prediction in a real traffic conflict situation (Sequence 1, See left plot in Figure 6). The vehicle trajectories are red and blue, with a dot marking their position, and the future positions are respectively cyan and yellow.

detection of an interaction between two very close “imaginary” road users, often with very high computed collision probability. Fortunately, these interactions are mostly filtered out by testing for the similarity between the trajectories of interacting road users using the LCSS distance. In two of the three sequences containing traffic conflicts, querying interactions for which the collision probability reaches values above 0.1 returns only the traffic conflicts. For the third sequence, it returns the traffic conflict and some interactions between road users in traffic moving in opposite directions. Querying the other sequences that contain no detectable traffic conflicts also return these “normal” interactions that can be easily identified. This shows that traffic conflict detection can be achieved by computing the collision probability. Adding other severity indicators will further improve the detection results.

VI. CONCLUSION

This paper has presented the development of a complete vision-based system for automated road analysis. After discussing traffic conflicts and defining a computable collision probability, a new incremental learning algorithm of motion patterns for motion prediction was introduced. The approach relies on the use of actual trajectories as motion patterns, or prototypes, and the LCSS to compare trajectories. This method has distinct advantages: it does not require any special pre-processing of trajectories, it is incremental and therefore suitable for online use, and requires limited tuning to produce useful results. The system was tested on extensive real traffic video data, and a few traffic conflict instances. It demonstrated that automated traffic conflict detection can be achieved by computing the collision probability. Future work will focus on the estimation of road users’ sizes and detection of over-segmented road users. We are in the midst of commencing in a comprehensive validation of the traffic conflict detection algorithms developed to-date. This work is expected to be reported in future publications.

REFERENCES

- [1] World Health Organization, “World report on road traffic injury prevention: Summary,” http://www.who.int/violence_injury_prevention/road_traffic/en/, 2004.

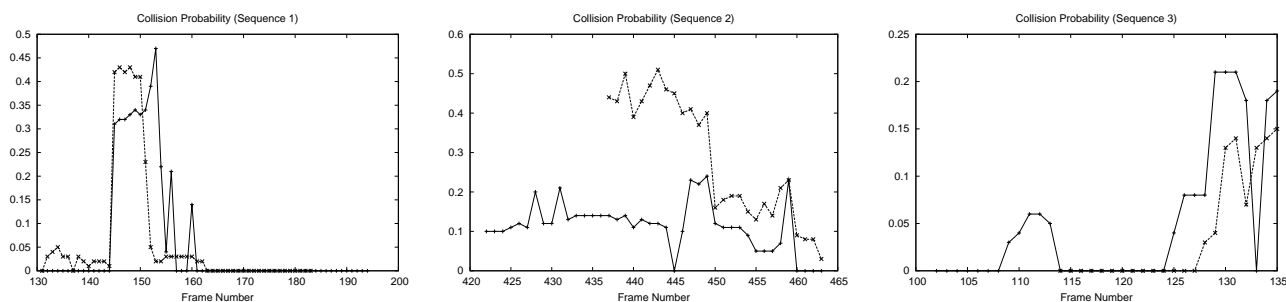


Fig. 6. The collision probability for the three traffic conflicts, as a function of time (counted in frame numbers). In all sequences, vehicle 1 travels south-bound through the intersection and vehicle 2 comes from an opposing approach. Vehicle 2 turns left in sequence 1 (left) (See Figure 5), right in sequence 2 (middle) and stops in sequence 3 (right).

- [2] C. Hydén, "The development of a method for traffic safety evaluation: The Swedish traffic conflicts technique," Ph.D. dissertation, Lund University of Technology, Lund, Sweden, 1987, bulletin 70.
- [3] R. van der Horst, "A time-based analysis of road user behavior in normal and critical encounter," Ph.D. dissertation, Delft University of Technology, 1990.
- [4] G. R. Brown, "Traffic conflict for road user safety studies," *Canadian Journal of Civil Engineering*, vol. 21, pp. 1–15, 1994.
- [5] T. Sayed, G. R. Brown, and F. Navin, "Simulation of Traffic Conflicts at Unsignalised Intersections with TSC-Sim," *Accident Analysis & Prevention*, vol. 26, no. 5, pp. 593–607, 1994.
- [6] A. Svensson, "A method for analyzing the traffic process in a safety perspective," Ph.D. dissertation, University of Lund, 1998, bulletin 166.
- [7] T. Sayed and S. Zein, "Traffic conflict standards for intersections," *Transportation Planning and Technology*, vol. 22, pp. 309–323, 1999.
- [8] F. Amundsen and C. Hydén, Eds., *Proceedings of the first workshop on traffic conflicts*. Oslo, Norway: Institute of Transport Economics, 1977.
- [9] W. Hu, X. Xiao, D. Xie, T. Tan, and S. Maybank, "Traffic accident prediction using 3d model based vehicle tracking," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 3, pp. 677–694, May 2004.
- [10] S. R. Perkins and J. I. Harris, "Traffic conflicts characteristics: Accident potential at intersections," *Highway Research Record*, vol. 225, pp. 35–43, 1968, highway Research Board, Washington D.C.
- [11] S. Messelodi and C. M. Modena, "A computer vision system for traffic accident risk measurement: A case study," ITC, Tech. Rep. ITC-irst T05-06-07, 2005.
- [12] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 108–118, June 2000.
- [13] S. Atev, H. Arumugam, O. Masoud, R. Janardan, and N. P. Papanikolopoulos, "A vision-based approach to collision prediction at traffic intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 4, pp. 416–423, Dec. 2005.
- [14] A. Lareshyn and H. Ardö, "Automated video analysis as a tool for analysing road safety behaviour," in *ITS World Congress*, London, 2006.
- [15] B. Maurin, O. Masoud, and N. P. Papanikolopoulos, "Tracking all traffic: computer vision algorithms for monitoring vehicles, individuals, and crowds," *Robotics & Automation Magazine, IEEE*, vol. 12, no. 1, pp. 29–36, Mar. 2005.
- [16] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A real-time computer vision system for measuring traffic parameters," in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. Washington, DC, USA: IEEE Computer Society, 1997, pp. 495–501.
- [17] N. Saunier and T. Sayed, "A feature-based tracking algorithm for vehicles in intersections," in *Third Canadian Conference on Computer and Robot Vision*. Québec: IEEE, June 2006.
- [18] C. Picciarelli and G. Foresti, "On-line trajectory clustering for anomalous events detection," *Pattern Recognition Letters*, vol. 27, no. 15, pp. 1835–1842, Nov. 2006.
- [19] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 28, no. 9, pp. 1450–1464, Sept. 2006.
- [20] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image and Vision Computing*, vol. 14, no. 8, pp. 609–615, Aug. 1996.
- [21] C. Stauffer and E. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, Aug. 2000.
- [22] X. Wang, K. Tieu, and E. L. Grimson, "Learning semantic scene models by trajectory analysis," MIT, Tech. Rep., Feb. 2006. [Online]. Available: <http://hdl.handle.net/1721.1/31208>
- [23] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *The International Journal of Robotics Research (IJRR)*, vol. 24, no. 1, 2005.
- [24] D. Makris and T. Ellis, "Learning semantic scene models from observing activity in visual surveillance," *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, vol. 35, no. 3, pp. 397–408, June 2005.
- [25] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic, "Discovering clusters in motion time-series data," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.
- [26] I. Cadez, S. Gaffney, and P. Smyth, "A general probabilistic framework for clustering individuals," University of California, Irvine, Tech. Rep. UCI-ICS 00-09, Mar. 2000.
- [27] N. Saunier and T. Sayed, "Clustering Vehicle Trajectories with Hidden Markov Models. Application to Automated Traffic Safety Analysis," in *International Joint Conference on Neural Networks*. Vancouver: IEEE, July 2006.
- [28] A. D. Vasquez and T. Fraichard, "Motion prediction for moving objects: a statistical approach," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, New Orleans, LA (US), Apr. 2004, pp. 3931–3936.
- [29] D. Buzan, S. Sclaroff, and G. Kollios, "Extraction and clustering of motion trajectories in video," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 2, Aug. 2004, pp. 521–524.
- [30] M. Vlachos, G. Kollios, and D. Gunopulos, "Elastic translation invariant matching of trajectories," *Machine Learning*, vol. 58, no. 2-3, pp. 301–334, Feb. 2005.
- [31] S. Atev, O. Masoud, and N. Papanikolopoulos, "Learning traffic patterns at intersections by spectral clustering of motion trajectories," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Oct. 2006, pp. 4851–4856.
- [32] S. Khalid and A. Naftel, "Classifying spatiotemporal object trajectories using unsupervised learning of basis function coefficients," in *VSSN '05: Proceedings of the third ACM international workshop on Video surveillance & sensor networks*. New York, NY, USA: ACM Press, 2005, pp. 45–52.
- [33] F. M. Porikli, "Trajectory distance metric using hidden markov model based representation," in *European Conference on Computer Vision (ECCV)*, May 2004.
- [34] L. Chen and R. Ng, "On the marriage of lp-norm and edit distance," in *Proceedings of 30th International Conference on Very Large Data Base*, Toronto, Canada, Aug. 2004, pp. 792–803.