

Road Sight Unit: Edge Image Processing for Vehicle Trajectory and Collision Prediction

José Cutileiro[†], Pedro Rosa[†], Orlando Remédios^{*}, and Miguel L. Pardal[†]

[†] INESC-ID, Instituto Superior Técnico, Universidade de Lisboa

{jose.cutileiro, pmgsrosa, miguel.pardal}@tecnico.ulisboa.pt

^{*} Sensefinity, Lisbon, Portugal

orlando.remédios@sensefinity.com

Abstract—Road accidents remain a leading cause of harm despite advances in vehicle safety systems. Onboard sensors improve awareness but are limited by field of view and positioning. Effective collision warnings require timely and contextualized information for accurate trajectory prediction.

This paper proposes the use of edge computing devices equipped with cameras to detect and track vehicles. It introduces a *similarity-based trajectory predictor* that combines vehicle movement history with road information, including lane structure, curvature, and likely paths. This approach achieved an average positional error of 50 cm (19.69 in) and improves the accuracy of collision prediction, achieving 83% compared to 50% with the baseline Extended Kalman Filter, while requiring less data than machine learning methods. It reduces false positives and enhances confidence in predicted trajectories, enabling earlier and more reliable collision detection.

Index Terms—Collision Detection, Trajectory Prediction, Vulnerable Road Users, Roadside Unit, Edge Computing

I. INTRODUCTION

Many people die in road accidents each year. In 2021, around 1.19 million people lost their lives on the road, according to the World Health Organization (WHO). 56% of people were *vulnerable road users* (VRUs), such as pedestrians and cyclists, who face greater risk of injury in traffic due to limited physical protection. This occurs even though many vehicles are equipped with sensors to predict potential collisions. These sensors perform well but are limited by range, positioning, and line of sight. Incorporating sensor data from external sources with better road visibility can increase local awareness and allow earlier acquisition of safety information.

This paper introduces Road *Sight* Unit, a roadside unit (RSU) device, positioned at the edge of roads, equipped with a single monocular camera and a processing unit, capable of perceiving the current traffic scene. It relies on vision processing algorithms running locally to detect the presence of road users and predict their movements, using also information about the geometry of the road.

A trajectory prediction algorithm was specifically designed for this use case, using historical vehicle movement data from

the area, that can run in a timely way in a limited capability device. The implemented *Similarity-Based Predictor* (SBP) serves as a middle-ground between established methods, such as *Kalman filters* (KF) [1], which do not require prior configuration, and more data-intensive approaches, such as *Long Short-Term Memory* (LSTM) networks, which rely on datasets to make their predictions [2].

To extract meaningful information from captured video frames, we employ object detection and tracking models. Our system predicts future positions using road geometry and considering lane structure, mandatory directions, restricted directions, and common traffic patterns, extending simple road user identification and enabling early detection of collisions between vehicles and VRUs. For example, when a vehicle is in the left lane, it is most likely to turn left. While this pattern is easily recognizable, many algorithms often struggle to detect it. The SBP can recognize such patterns and use them for better predictions.

After predicting collisions, the next step is to warn road users. To reach as many as possible, the information needs to be broadcast, preferably using standardized *Personal Safety Messages* (PSMs) [3], through all available *Vehicular Ad-hoc Networks* (VANETs) [4]. However, before that can be done, the data must be converted from image pixel coordinates of the RSU frame of reference to a global coordinate system to ensure correct spatial interpretation. This transformation relies on multiple reference points and requires prior calibration of the RSU. Our work also handles these challenges.

This paper is structured as follows: Section II reviews related work, Section III presents preliminary experiments, Section IV details the proposed solution, Section V reports results of the prototype, and Section VI concludes.

II. RELATED WORK

Our work builds on traffic monitoring tools and prior research in trajectory prediction.

A. Traffic Monitoring Tool

Cancouët et al. [5] developed an evaluation framework for their roadside infrastructure software designed to detect collision risks between vehicles and VRUs. Their work provides a structured description of the system under evaluation and the corresponding assessment methodology. This methodology

Work supported by national funds through Fundação para a Ciência e a Tecnologia, I.P. (FCT) under projects UID/50021/2025 and UID/PRR/50021/2025 (INESC-ID) and Project Blockchain.PT – Decentralize Portugal with Blockchain Agenda, (Project no 51), WP 1: Agriculture and Agri-food, Call no 02/C05-i01.01/2022, funded by the Portuguese Recovery and Resilience Program (PRR), The Portuguese Republic and The European Union (EU) under the framework of Next Generation EU Program.

begins with a calibration phase for the RSU, enabling the conversion of pixel coordinates into a global coordinate system. Once calibrated, the RSU processes live video frames through several stages. First, object detection identifies elements within the frame. Then, object tracking associates these detections across multiple frames, enabling motion estimation. Finally, the processed information is structured and converted into widely recognized standards, such as the PSM defined in SAE J2735 [3], allowing dissemination via broadcast transmission. When developing such a system, it is crucial to consider *timeliness*, as it must provide information quickly enough for road users to prevent accidents. These principles form the foundation of a system with similar objectives to ours. However, there are significant differences between our solution and the one evaluated by Cancouët et al. [5], primarily due to the fact that our system operates with a single camera.

B. Trajectory Prediction

Real-time forecasting of vehicle positions is a challenging task, especially in urban environments. Factors such as traffic signs, pedestrian presence, and congestion can unpredictably alter vehicle motion, Jaiswal et al. [6] propose a solution utilizing *Extended Kalman Filters* (EKF) [7] to mitigate this challenge, demonstrating that EKF offers improvements over the standard KF [1]. However, such prediction methods remain limited when compared to approaches that incorporate road geometry. For instance, *Path-Based Trajectory Prediction* [8] (PBP) enhances forecasting accuracy by constraining predictions using both the agent's previous positions and the surrounding map. PBP achieves better results on standard trajectory prediction metrics than the baselines. In PBP, the authors utilize the Argoverse dataset [9], granting them access to a high-definition vectorized map and enabling the dynamic extraction of reference paths from the map. However, because an RSU is fixed in place, it can locally store information about common paths, which reduces data needs, improves efficiency, and makes the solution more scalable. This requires certain adaptations, which will be discussed in Section III-B.

III. PRELIMINARY EXPERIMENTS

We conducted several experiments to understand how to perform coordinate conversion and how to predict object trajectories effectively. Additionally, we aimed to assess whether image processing models could generate sufficiently accurate information within the time constraints imposed by our use case. These tests were conducted on a computer running Windows 11, equipped with an Intel Core i7 10th Gen processor, an Intel(R) UHD Graphics integrated GPU, and an NVIDIA GeForce GTX 1650 Ti graphics card.

A. Image Processing

Working with object detection and tracking models yields approximations of reality, and relying on a single video frame is often limiting. Evaluating these models remains challenging and is an active area of research [10].



Fig. 1: Example of three detection frames in sequence.

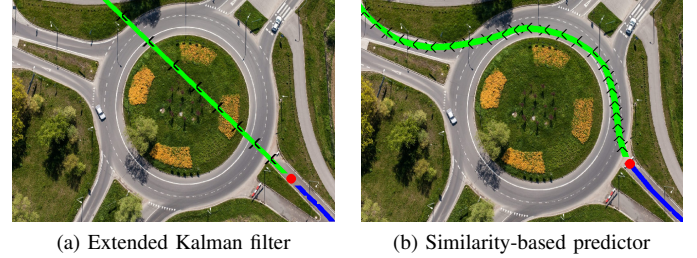


Fig. 2: Comparison of the outputs of different algorithms

We combined YOLO [11] with DeepSORT [12], which have been previously applied in road traffic environments [13]. Through testing under various conditions, we observed a real-time failure rate of 4.79%. Fig. 1 illustrates the issue where, even without obstructions, the pedestrian is not detected.

In an exploratory phase, we tested various trajectory prediction models under noisy conditions. Among the tested models were EKF, KF, and Markov models, all of which exhibited performance degradation when exposed to noisy inputs. Based on these observations, we recognized the need for an alternative approach, less sensitive to noise and more tailored to the specific road environment.

B. Similarity-Based Prediction

Based on the same principles as PBP [8], the goal is to develop a trajectory prediction system that accounts for road constraints. In an earlier phase of our project, predictions were generated using EKF [7]; however, this resulted in forecasts similar to those shown in Fig. 2 (a). The predictions, represented as green line with arrows, disregard the presence of a roundabout, predicting that the vehicle will continue straight ahead. Predictions that ignore typical vehicle behavior reflect a fundamental gap in understanding. To address this limitation, we introduce Similarity-Based Predictor (SBP), an algorithm that, when provided with a historical sequence of positions, predicts future locations by matching them to preconfigured trajectories. This approach leverages lane structures, mandatory and prohibited directions, and even road curvature. Our use case assumes that this type of data is static, meaning the road layout does not change. As shown in Fig. 2 (b), the predictions take into account the curvature of the roundabout to significantly improve the forecast accuracy.

Algorithm 1 Similarity-Based Predictor

Require: current_traj, predefined_trajs, min_points, max_points, score_threshold

Ensure: Continuation of the trajectory most similar to the current one

```

1: len  $\leftarrow$  size(current_traj)
2: if len < min_points then
3:   return []
4: end if
5: current_arr  $\leftarrow$  convert_to_array(current_traj)
6: best_score  $\leftarrow$   $\infty$ 
7: best_match  $\leftarrow$  []
8: for each traj in predefined_trajs do
9:   if size(traj)  $\leq$  len then
10:    continue
11:  end if
12:  for  $i = 0$  to (size(traj) - len) do
13:    reference_window  $\leftarrow$  sublist(traj,  $i$ ,  $i + \text{len}$ )
14:    diff  $\leftarrow$  current_arr - reference_window
15:    score  $\leftarrow$  sum(norm(diff))
16:    if score < best_score then
17:      best_score  $\leftarrow$  score
18:      start_pred  $\leftarrow$   $i + \text{len}$ 
19:      end_pred  $\leftarrow$  start_pred + max_points
20:      best_match  $\leftarrow$  sublist(traj, start_pred, end_pred)
21:    end if
22:  end for
23: end for
24: if best_score < score_threshold then
25:   return best_match
26: else
27:   return EKF(current_traj, max_points)
28: end if

```

Regarding Algorithm 1:

- `current_traj` represents the history of positions in pixel coordinates of the object;
- `predefined_trajs` refers to the pre-installed trajectories within the RSU during the configuration phase, also in pixel coordinates;
- `min_points` is the minimum number of points a trajectory must have for the SBP to generate meaningful results;
- `max_points` denotes the maximum number of points that the algorithm can predict;
- `score_threshold` represents the maximum acceptable distance in pixel space for a trajectory to be considered valid.

C. Coordinate Transformation

To enable data sharing across different road safety systems, the local coordinates must be accurately mapped to global ones. Given the static position of a RSU, *Ground Control Points* (GCPs) can be utilized to assist in mapping. GCPs are represented in both pixel coordinates and the *Global*

TABLE I: Mean Error and Standard Deviation of Test Samples

Location	Mean Error (m)	Standard Deviation (m)
Michigan, USA	0.532	0.455
Chengdu, China	0.471	0.392

Positioning System (GPS), enabling the computation of a homography matrix that serves as a mapping between these two reference frames. However, a key challenge is that both the collection of GCPs and the calibration of the RSU require manual procedures, limiting the broader applicability of this solution.

We conducted early tests to ensure the quality of our solution. When constructing a homography matrix, the general rule is that the more GCPs are used, the better, assuming that they are accurately measured. In our tests, we used 13 calibration points, a number that allows for a well-distributed set of points in the application area, and 10 test points, which were widely dispersed to identify regions where the transformation yielded poor results. The outcomes were generally positive, with an average error of 50 cm (19.7 in) when compared to reference values as shown in Table I.

IV. PROPOSED SOLUTION

Our preliminary results showed promising outcomes: we achieved an average error of 50 cm (19.7 in) using a single camera within a global coordinate system; and we also succeeded in predicting some movements using preconfigured trajectories. Next, we developed a system that integrates all components. The system follows an architecture similar to other systems mentioned in Section II. As shown in Fig. 3, it includes processing stages for object detection, object tracking, trajectory prediction, and conversion into GPS coordinates.

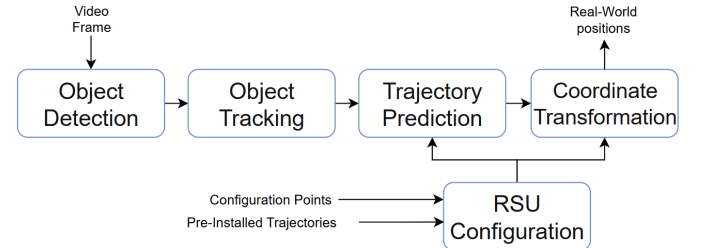


Fig. 3: Road Sight Unit image processing pipeline

Once the information is processed, it is important to determine which data should be shared. This requires encoding messages according to well-defined standards to ensure global interpretation of the data. A central aspect of the system is determining what information should be shared over the network. For testing the communication, we used *Bluetooth Low Energy* (BLE). In an ideal scenario, all data computed by the RSU would be transmitted, consisting of both current and predicted future positions of detected road users, along with their respective classifications. This can be achieved using PSMs, encapsulating information in a well-defined protocol,



Fig. 4: Downtown Coldwater, MI, USA traffic camera with trajectory visualization over 5 minutes

to facilitate the integration with other systems. However, transmitting all computed data requires sending large amounts of data simultaneously and handling message synchronization issues. Therefore, our setup limits the transmitted data to collision risk positions between vehicles and VRUs.

A. Configuration Phase

In this initial phase, the roadside unit must be configured to enable coordinate conversion. Therefore, data collection from real-world scenarios is required, with the collected data being mapped to corresponding points in the image.

In this phase, we also need to provide the possible trajectories within the application area, enabling the use of SBP during the monitoring phase. To determine possible trajectories, a tracking tool should be executed, capturing historical paths as shown in Fig. 4. As an additional step, these paths can be merged and smoothed, improving the efficiency of the SBP and enhancing the input quality by reducing the noise.

B. Monitoring phase

The goal of this stage is to process video frames and extract information. We deploy the processing pipeline on edge computing devices, ensuring that analysis occurs close to the data source and minimizing latency.

We begin by utilizing YOLO. At this stage, we obtain object classifications and positions. These detections are then fed into our tracking system, DeepSORT, which means that issues such as object occlusion or data loss have a reduced impact on the tracker's output [12]. Nevertheless, as observed in the preliminary results, these factors must still be taken into account. The tracker enables the system to track movements across multiple frames. This provides a historical sequence of positions for each object in the image, which can be leveraged to predict future locations. For this purpose, we employ SBP, specifically designed for our use case. However, to achieve good performance, SBP requires a limited number of predefined trajectories. This constraint aligns well with vehicle movement patterns, as they follow structured paths, but it is less suitable for VRUs such as pedestrians, whose movement are more unpredictable and have greater degrees of freedom. To simplify implementation, VRU trajectory prediction will be handled using the EKF. Using the predicted

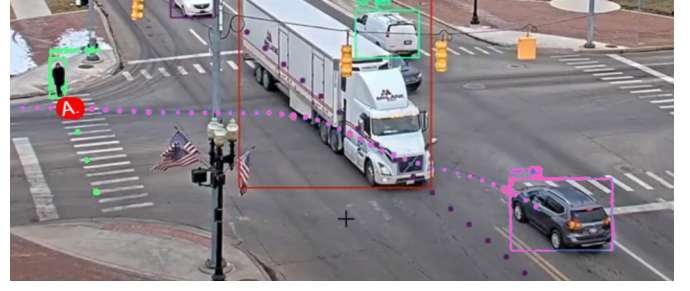


Fig. 5: Downtown Coldwater, MI, USA traffic camera: Trajectory prediction and real-time overlap verification



Fig. 6: Coordinates obtained from trajectory overlay according to Google Earth

trajectories, potential collisions between vehicles and VRUs can be identified by detecting trajectory overlaps.

After all the information is processed, it must be transformed into the global coordinate system using the previously computed homography matrix and subsequently encoded for transmission.

V. EXPERIMENTAL RESULTS

Following the architecture outlined in Section IV, we obtained several promising results. The tests were conducted using a public video feed from Downtown Coldwater, Michigan, in the United States of America. The feed was selected for its high image quality and availability. As shown in Fig. 5, the SBP effectively identifies vehicle trajectories. In this example, although the vehicle is still in a pre-turn position, the SBP correctly predicts that, given its position in the left lane, it is likely to turn left. Looking again at Fig. 5, we can see that even in a scenario where the vehicle lacks direct visibility of the pedestrian due to the obstruction caused by the truck, our solution was able to detect the risk of collision.

Regarding collision detection, our algorithm successfully identifies trajectory overlap at the marked point 'A'. Using the homographic matrix and providing the detected collision point, we obtain the coordinates "41.940508, -85.000814". As shown in Fig. 6, these coordinates align closely with reality.

Testing the effectiveness of the Road Sight Unit, particularly the SBP, is challenging due to the lack of videos featuring actual accidents between vehicles and pedestrians. To address this limitation, we employed video overlay techniques, combining one video containing the vehicle component with

TABLE II: Performance Comparison of EKF and SBP Using Video Overlay Data from Downtown Coldwater

Method	TP	TP (> 4s)	NM	FP	FP (Sidewalk)	FN	Precision
EKF	4	2	4	8	4	3	50%
SBP	6	4	4	2	0	1	83%

another depicting the pedestrians. This approach enabled the creation of synthesized collisions. Through this method, we measured the anticipation time of the system and also the performance of different predictors, assessing the SBP improvements.

Analyzing the videos generated using video overlay, we obtained the results shown in Table II. The generated videos contain seven collisions. The scenarios were designed to cover diverse situations in order to avoid favoring any particular algorithm. The dataset emphasizes scenario diversity rather than scale, given that the video editing process is performed manually. Each entry in Table II is examined in detail:

- *TP (true positives)*: represent actual collisions that did occur and were successfully detected;
- *TP (more than 4s anticipation)¹*: a subset of TP, denotes collisions detected with more than 4 seconds in advance;
- *NM (near miss)*: situations that did not result in actual collisions due to sudden braking or evasive maneuvers, but which could realistically have led to collisions.
- *FP (false positives)*: detections of collisions that were incorrect or baseless;
- *FP (Sidewalk)*: a subset of FP, indicating false detections occurring on the sidewalk. This field highlights that SBP does not produce collision warnings on the sidewalk, owing to its understanding of road geometry;
- *FN (false negatives)*: collisions that were not detected.
- *Precision*: overall quality of the obtained results.

$$\text{Precision} = \frac{\text{TP} + \text{NM}}{\text{TP} + \text{NM} + \text{FP}}$$

According to Table II, we observe that when using the EKF, only 50% of the detected collisions are meaningful. The fragility of the EKF in our scenario is amplified by the object detection model. In contrast, SBP, leveraging its knowledge of the operational environment, is less affected by this limitation. Knowing the operational environment requires only a small amount of data compared to models that aim to be generic like PBP, which contributes to the efficiency of the model when delivering results. However, it is important to note that obtaining information in a timely manner remains challenging and the field of view of the camera is a critical factor.

The tests show that results depend on the specific scenario and camera field of view. In cases where detection occurs well in advance of the collision, such as when the vehicle approaches from top to bottom, the outcomes are highly promising. Conversely, when the vehicle moves from bottom

to top, detection occurs closer to the collision event, producing results that are reasonable but delayed due to the limited field of view. Still, the analysis of results shows that SBP reduces false positives and detects hazards earlier than EKF.

VI. CONCLUSION

An individual vehicle perspective is often insufficient for perceiving the full road environment, even with high-end sensors. This leaves other road users at risk, especially VRUs.

This work proposed a RSU equipped with a single camera to monitor the road from a vantage point, to predict road user movements, and to broadcast collision warnings. For this purpose, we developed SBP, an algorithm that integrates each vehicle's historical data with local road characteristics, such as lane geometry, mandatory and restricted directions, to achieve more accurate trajectory prediction. SBP is a balance between classical methods like EKF and data-heavy approaches such as PBP or LSTMs, prioritizing simple installation of the camera device and minimal data needs to enable broader deployment.

REFERENCES

- [1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, 1960.
- [2] F. Althché and A. de La Fortelle, "An lstm network for highway trajectory prediction," in *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*. IEEE, 2017, pp. 353–359.
- [3] SAE International, "V2X Communications Message Set Dictionary J2735_202409," in *SAE International Standard J2735*, 2024.
- [4] C. K. Toh, *Ad hoc mobile wireless networks: protocols and systems*. Pearson Education, 2001.
- [5] M. Cancouët, R. Bellessort, E. Nassor, H. Ruellan, and J.-M. Bonnin, "Evaluation of multi-camera-based localization for accurate collision risk detection," in *2024 IEEE 100th Vehicular Technology Conference*.
- [6] R. K. Jaiswal and C. Jaidhar, "Location prediction algorithm for a nonlinear vehicular movement in vanet using extended kalman filter," *Wireless Networks*, vol. 23, pp. 2021–2036, 2017.
- [7] J. H. Lee and N. L. Ricker, "Extended kalman filter based nonlinear model predictive control," *Industrial & Engineering Chemistry Research*, vol. 33, no. 6, pp. 1530–1541, 1994.
- [8] S. Afshar, N. Deo, A. Bhagat, T. Chakraborty, Y. Shao, B. R. Bud-dharaju, A. Deshpande, and H. C. Motional, "Pbp: Path-based trajectory prediction for autonomous driving," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 12 927–12 934.
- [9] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.
- [10] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, "Evaluating the evolution of yolo (you only look once) models: A comprehensive benchmark study of yolo11 and its predecessors," *arXiv preprint arXiv:2411.00201*, 2024.
- [11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [12] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3645–3649.
- [13] M. A. B. Zuraimi and F. H. K. Zaman, "Vehicle detection and tracking using yolo and deepsort," in *2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*. IEEE, 2021, pp. 23–29.
- [14] J. Wan, C. Wu, and Y. Zhang, "Effects of lead time of verbal collision warning messages on driving behavior in connected vehicle settings," *Journal of Safety Research*, vol. 58, pp. 89–98, 2016.

¹A study by Wan et al. [14] demonstrated that accident anticipation between 4 and 8 seconds led to greater reductions in kinetic energy.