3D Multi-modal Multi-object Tracking via Machine Learning and Analytic Collision Risk Calculation for Autonomous Vehicles Navigation: The Analytic Collision Risk Module

Spathoulas Dimitrios
Department of Industrial Engineering and Management
Democritus University of Thrace

CONTENTS

I INTRODUCTION AND GENERAL SOLUTION 1

II TRAJECTORY FORMULATION AND ROBOT'S SAFE ZONE 1

III ANALYTIC COLLISION STATE PROBABILITY 2

IV RESULTS 2

V CONCLUSIONS AND FUTURE WORK 3

References 3

LIST OF FIGURES

1 Minkowski sum for an AV-obstacle pair . . . 2
2 Intergral boundaries. Collision octagon is shown in red, 1-Sigma ellipsis of Gaussian state distribution of obstacle in blue. . . . 2
3 Collision risk calculation. Octagon below the x-axis. Reversal of contributions. . . . 2
4 Collision risk calculation. Octagon segments above and below the horizontal axis. . . . 3

3D Multi-modal Multi-object Tracking via Machine Learning and Analytic Collision Risk Calculation for Autonomous Vehicles Navigation: The Analytic Collision Risk Module

***Abstract*—This report focuses on the Analytic Collision Risk Module, which is the last of the three main components in my master's thesis [1] [2] [3] [4]. This module is dedicated to the analytical calculation of collision risk between the ego vehicle and all surrounding track-obstacles within a defined perimeter of its center over a finite time horizon. Collision detection plays a critical role in ensuring the safety of autonomous vehicle systems, serving as a key component that relies on accurate detection and tracking. It provides essential information for both short- and long-term decision-making in path planning and mission control. Without this capability, autonomous driving would be extremely dangerous and practically impossible [1]. We leverage data from seven sensors—one LiDAR and six cameras—obtained from the NuScenes dataset [2]. The LiDAR point cloud data is processed using the 3D detector CenterPoint [3], while the camera data is analyzed using the 2D detector and segmentor Mask R-CNN [4]. Tracklets are generated through our multi-modal multi-object tracking module, which integrates both geometric and appearance-based features for robust object tracking.**

I. INTRODUCTION AND GENERAL SOLUTION

To ensure accurate and robust collision calculations, it is essential to detect and track all objects of interest providing the critical information required for precise analysis. By doing so, the future positions of each tracked object can be predicted, taking into account the uncertainties in their movements. Finally, the calculations must be designed to meaningfully model the collision risk between each pair of objects, ensuring that every obstacle is individually considered and that the results are both meaningful and actionable. Instead of relying on traditional measures such as Time to Collision (TTC), Time to React (TTR), or Time to Brake (TTB)—which fail to account for the variances in the calculated times or the overall probability of a collision occurring—the implemented paper [5] adopts the concept of Collision State Probability and extends its practicality proposing an analytical formulation prior to relevant work [6] that relied on Monte Carlo simulations.
Collision state probability (CSP) is the probability of spatial overlap of two objects at a certain point in time, which simplyfies to the probability, that the centroid of the obstacle is somewhere inside the octagon created by the convolution between the obstacle around the ego vehicle [5]. Assuming the states of the two objects are independent of each other, one has to solve the integral over their common state distribution to compute the collision risk:

$$\int_{x_1}\int_{y_1}\int_{\varphi_1}\int_{x_2}\int_{y_2}\int_{\varphi_2} I_C(x_1,y_1,\varphi_1,x_2,y_2,\varphi_2) \\ p\,(x_1,y_1,\varphi_1)\;\, p\,(x_2,y_2,\varphi_2)\;\, dx_1\,dy_1\,d\varphi_1\,dx_2\,dy_2\,d\varphi_2 \quad (1)$$

With the indicator function (2) yielding 1, if the two rectangles overlap at least partially:

$$I_C(x_1,y_1,\varphi_1,x_2,y_2,\varphi_2)=1 \quad \text{if} \quad S(x_1,y_1,\varphi_1)\cap S(x_2,y_2,\varphi_2)\neq 0 \quad (2)$$

II. TRAJECTORY FORMULATION AND ROBOT'S SAFE ZONE

The Autonomous Vehicle (AV) is assumed to follow a deterministic trajectory (as designed by the trajectory planner), thereby deriving its future states from the dataset. For each obstacle-track, the next position is predicted using a dynamic system with constant linear velocity and fixed orientation, incorporating process noise. This noise accounts for uncertainties in position and velocity, as reflected in the precomputed covariance matrices based on the selected detector. Since there is no correction step, the obstacle's covariance gradually increases over time. For the state transition model:

$$\mathbf{F}=\begin{bmatrix}1&0&t&0\\0&1&0&t\\0&0&1&0\\0&0&0&1\end{bmatrix} \quad (3)$$

Where $t = 0.5$ (based on the dataset's sensor synchronization). The state vector is updated as:

$$\mathbf{x}_t=\mathbf{F}\cdot\mathbf{x}_{t-1} \quad (4)$$

And the uncertainty increases as:

$$\mathbf{P}_t=\mathbf{F}\cdot\mathbf{P}_{t-1}\cdot\mathbf{F}^\top+\mathbf{Q} \quad (5)$$

Here, $\mathbf{Q}$ and $\mathbf{P}_{t=0}$ are obtained from the precomputed covariance matrix $\mathbf{\Sigma}_0$. After each such prediction, a new polygon is computed for each AV-obstacle pair, with the reference center

[1] https://github.com/DimSpathoulas/PointCloud_Feature_Extractor
[2] https://github.com/DimSpathoulas/2D_FEATURE_EXTRACTOR
[3] https://github.com/DimSpathoulas/GeomApp_3MOT
[4] https://github.com/DimSpathoulas/Collision_Risk_Calculation

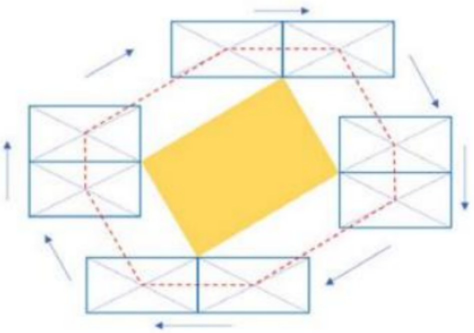


Fig. 1. Minkowski sum for an AV-obstacle pair

being the robot-centric coordinates, through their convolution, i.e., the Minkowski sum. The result of the Minkowski sum between two rectangles is an octagon (if they do not overlap).

III. ANALYTIC COLLISION STATE PROBABILITY

To calculate the CSP for each AV-obstacle pair, the coordinate system is first transformed based on the obstacle's position in space and its orientation. This ensures that the principal axes of the covariance align with the orthogonal axes of the reference frame. All line segments between consecutive vertices of the polygon are constructed, each assigned a value of -1 or 1 depending on whether it lies below or above the horizontal axis. Line segments that are nearly parallel to the vertical axis are excluded according to the methodology.

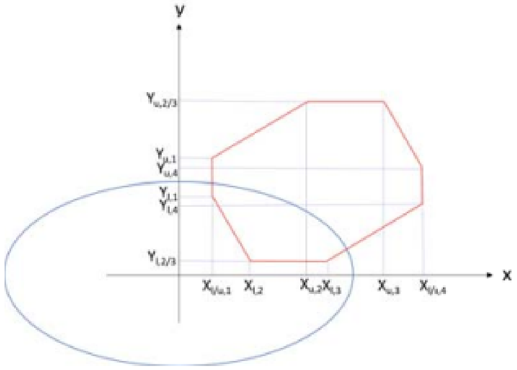


Fig. 2. Intergral boundaries. Collision octagon is shown in red, 1-Sigma ellipsis of Gaussian state distribution of obstacle in blue.

The volume under the intersection area between the Gaussian covariance and the octagon represents the CSP between the specific obstacle and the AV at the given time P:

$$P(C,T_p) = \int_y \int_x l_C(x,y) \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)} \, dx \, dy \quad (6)$$

Decomposing the octagon into C_n segments:

$$P\Big(C_l, T_p\Big) = \frac{1}{\sqrt{8\pi}\sigma_x} \int_{x=x_{l,i}}^{x=x_{u,i}} \operatorname{erf}\left(\frac{m_i x + b_i}{\sqrt{2}\sigma_x}\right) e^{-\frac{x^2}{2\sigma_x^2}} \, dx \quad (7)$$

Here erf is the Gaussian error function, which gives the probability of the Gaussian within the specified limits. For the computation of the integral, the 'quad' function from the 'scipy.integrate' package is used.

Finally, the total probability at this specific time T_p for this AV-obstacle pair is:

$$P(C,T_p) = \sum_{i=0}^{i<m} P(C_i,T_p) - \sum_{i=m}^{i<n} P(C_i,T_p) \quad (8)$$

Where $i < m$ represents regions of positive contribution. If the octagon is below the x-axis, the positive and negative contributions are reversed, which is why the line segments have the aforementioned property. If the analysis were performed with respect to the y-axis, the contribution would take into account the relative position with respect to the x-axis, with positive contributions being those to the right of it. There may be no negative contribution when some segments are above the y-axis and the rest are below it.

IV. RESULTS

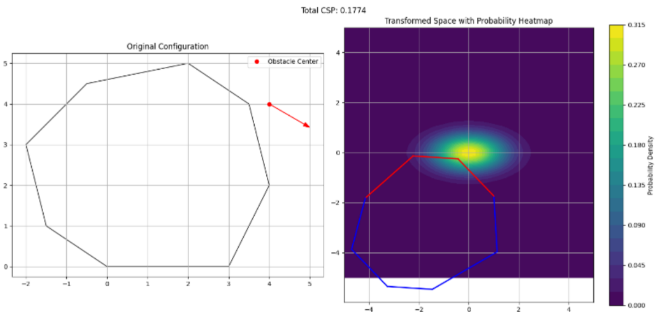
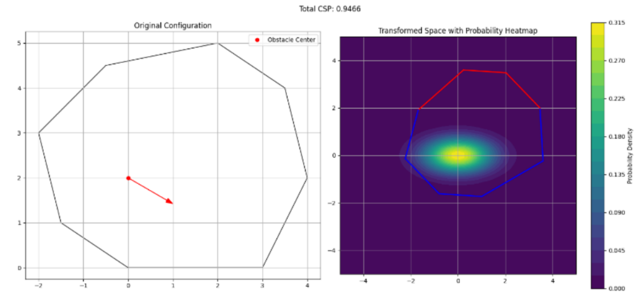


Fig. 3. Collision risk calculation. Octagon below the x-axis. Reversal of contributions.

It is important to note that as the obstacle's covariance increases and the octagon approaches the center, the collision risk probability does not increase linearly. This is because the growing uncertainty spreads the probability mass of the obstacle's position over a larger area, potentially diluting the likelihood of the obstacle being within the fixed-size octagon.



Εικόνα 7-4: Υπολογισμός κινδύνου σύγκρουσης (τμήματα οκταγώνου άνω και κάτω οριζόντιου άξονα)

Fig. 4. Collision risk calculation. Octagon segments above and below the horizontal axis.

V. CONCLUSIONS AND FUTURE WORK

Due to the nature of the dataset and the detector's accuracy, risk values exceeding 10% are rare. For the calculation, knowledge of the relative velocity of the objects is not required. An increase in covariance does not necessarily translate to a higher collision risk, even if the objects remain stationary relative to each other. However, the risk increases geometrically when penetration occurs. In specific projection scenarios (depending on the prediction time in seconds), the octagon may fully intersect the obstacle, causing the collision probability to first rise and then fall. This is because there is no direct control over the autonomous vehicle (AV), and the prediction model becomes less reliable over time. The collision probability between the octagon and covariance, as well as between the corresponding rectangle and covariance, is similar in certain scenarios, depending on the size difference resulting from geometric simplification. Currently, there is no real-time trajectory planning to fully model and utilize the algorithm.

To enhance its effectiveness, the system would greatly benefit from simulation scenarios. Additionally, implementing a collision probability calculation that accounts for the dynamic state of the objects would be particularly useful. Finally, extending the algorithm to three dimensions would further improve its applicability in air and water environments. With appropriate modifications, the system can be adapted for use in air (e.g., UAVs), water (e.g., submarines), and even space applications (e.g., space debris avoidance) through tailored datasets and simulations.

REFERENCES

[1] A. Houenou, P. Bonnifait, and V. Cherfaoui. Risk assessment for collision avoidance systems. In *2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*, pages 386–391. IEEE, 2014.

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[3] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[5] A. Philipp and D. Goehring. Analytic collision risk calculation for autonomous vehicle navigation. *2019 International Conference on Robotics and Automation (ICRA)*, pages 1744–1750, 2019.

[6] A. Lambert, D. Gruyer, and G. Saint Pierre. A fast monte carlo algorithm for collision probability estimation. In *2008 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 406–411. IEEE, 2008.