AUTONOMOUS NAVIGATION FOR UNSTRUCTURED ENVIRONMENTS

DEEKSHA SETHI [DEESETHI@SEAS], NITHASREE SOMANATHAN [NITHI10@SEAS], SHREYA LANJEWAR [SHREYAL@SEAS], TANAYA GUPTE [TANAYAG@SEAS]

ABSTRACT. Autonomous navigation in unstructured outdoor environments presents significant challenges due to the complexity and variability of natural terrain. This project proposes a vision-based approach to enable a ground robot to navigate through unmapped, off-road terrains safely and efficiently. The system employs a Deep Learning model to segment input images and predict traversable regions, providing a pixel-wise traversability map. These traversability maps are then used to generate waypoints which are subsequently integrated with the Dynamic Path Controller (DPC) algorithm. The DPC algorithm guides our ground robot by continuously steering toward the nearest attainable waypoint. The whole pipeline was implemented on Jetson Xavier NX mounted on an F1Tenth car. A key motivation is to overcome the limitations of traditional sensors like LiDAR, which may incorrectly identify traversable areas like grass as obstacles. The proposed vision-based method aims to recognize the true traversable regions, enabling exploration and navigation through unstructured natural environments without prior mapping. The system's ability to operate safely in unmapped, off-road conditions has applications in fields such as search and rescue, environmental monitoring, and planetary exploration. The proposed pipeline yields a success rate of 85% in terms of waypoint generation and navigation.

INTRODUCTION

Autonomous navigation remains an open challenge, particularly in unstructured outdoor environments where natural terrains exhibit high complexity and variability. Traditional sensors like LiDAR and ultrasonic rangefinders often struggle to accurately distinguish traversable regions from obstacles in such settings, hindering the deployment of autonomous robots for crucial applications like search and rescue, environmental monitoring, and planetary exploration. These applications necessitate the ability to safely explore and navigate unmapped, off-road areas. To address this challenge, we propose a novel vision-based approach that leverages deep learning for terrain assessment and path planning. Our system utilizes visual data from onboard camera to semantically segment the environment and predict traversable regions at a granular, pixel-level resolution. This is done using the ResNet18-based neural network architecture - TRAVNET [10]. The generated traversability maps are then used to plan feasible paths and generate waypoints that guide the robot's navigation via the DPC algorithm. By directly learning to identify true traversable areas from visual input data, our method circumvents the limitations of geometric perception sensors in distinguishing complex natural obstacles like vegetation. This vision-based terrain mapping enables safer and more efficient autonomous exploration without requiring prior maps or terrain models, making it well-suited for operations in previously unmapped environments.

Contributions

(1) Existing datasets for off-roading and unstructured environments are limited, prompting the need for a more comprehensive collection. Recognizing this gap, we meticulously curated a small yet diverse dataset tailored specifically for such challenging terrains. Notably, many available datasets are sourced from larger vehicles, rendering them incompatible with our implementation on an F1tenth-sized car. Thus, our dataset is tailored to small-frame autonomous vehicles.
(2) A novel trajectory generator has been implemented. We generate two trajectories. One takes a more risky and uneven path while the other takes a safe path through more traversable regions. This gives the user a choice in the kind of path the vehicle would be able to take based on its dynamics.
(3) Implemented the entire pipeline on the Jetson Xavier NX with the F1Tenth car, creating a foundation for future work in this domain with a relevant dataset.

RELATED WORK

Bird-eye View (BEV) techniques, also referred to as top-down or overhead view navigation methods, are employed in structured outdoor environments to facilitate navigation [1], [2], [3]. This approach relies on a high-level perspective of the surroundings, often utilizing aerial imagery or maps, to enhance spatial comprehension and path planning.

Nevertheless, it is important to note that these techniques primarily forecast ground-level information and do not encompass 3D terrain data. Consequently, their applicability in unstructured, off-road conditions is challenged due to this limitation.

Learning-based navigation algorithms have shown promise in overcoming the challenges by learning from the data. These methods utilize sensor data to make real-time decisions related to navigation tasks [4]. Certain approaches within this domain focus on directly learning the mapping from sensory input to control commands, facilitating end-to-end navigation without the need for explicitly programmed algorithms. Additionally, navigation algorithms often involve semantic segmentation of the scene to categorize obstacles and terrain. However, these methods encounter limitations due to the reliance on geometric information, which may not provide a comprehensive description of the scene [5], [6], [7]. While Reinforcement Learning-based techniques are employed, they suffer from sample inefficiency, posing challenges for deployment in unstructured environments. Furthermore, realistic simulations may be necessary to generate training datasets for these methods.

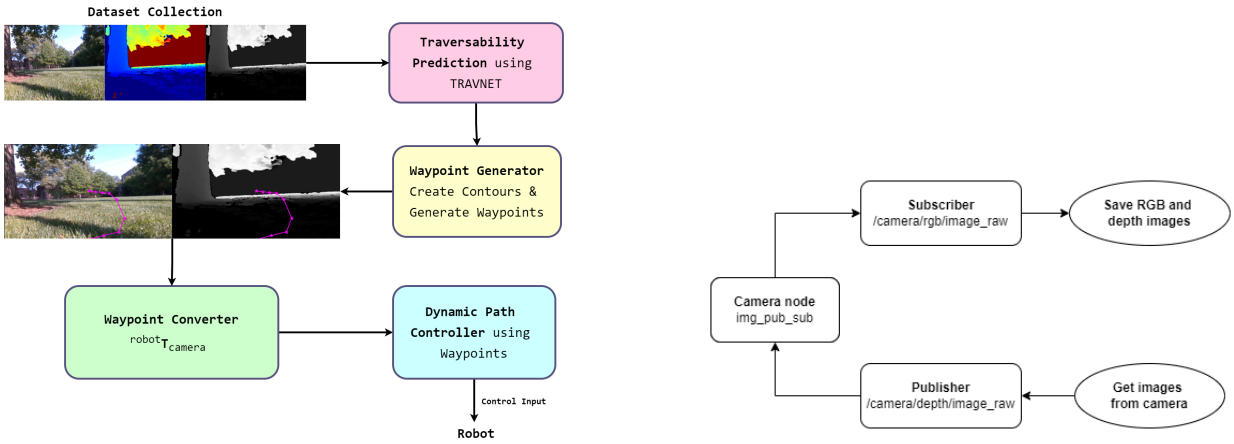
TerrainNet uses a camera-based vision system for off-road autonomous driving [8]. It is a prominent study on using camera-based off-road navigation. However, this approach also uses the BEV semantic and geometric segmentation.

BADGR [9] trains the end-to-end pipeline for generating the labels for defining the traversability which works by collecting off-policy data in real world. However, the data required for training BADGR algorithm is immensely large which makes it a tedious method to run. Instead, **WayFAST** [10] a self-supervised approach for wheeled mobile robots to predict traversable paths in outdoor environments uses RGB and depth data, along with navigation experience, to generate these paths. It is particularly effective in avoiding geometric obstacles and less traversable terrain, such as snow, and is more data-efficient than previous methods. WayFAST's ability to navigate unstructured outdoor environments without predefined waypoints is a significant advancement in the field of autonomous robot navigation. WayFASTER [11] is an extension of the WayFAST paper, eliminating the need for heuristics for generating labels.

APPROACH

Experimental Setup: All experiments were performed on the F1Tenth car which uses a differential drive. We have used Intel RealSense D435i camera which outputs RGBD images. An Nvidia Jetson Xavier NX with a compute power of 384-core Volta GPU and 6-Core Arm v8 CPU is mounted on the car. Dependencies for this project: ROS2 Foxy, Python3.8, OpenCV, CUDA.

The pipeline framework and dataset collection publisher-subscriber node are provided in Figure 1.



(A) Framework Architecture

(B) Dataset Collection: Node Tree

FIGURE 1. Framework pipeline and dataset collection publisher-subscriber node.

Data collection: Our objective is centered on validating our entire pipeline in an outdoor setting using the F1tenth Autonomous Racing car. Both data collection and algorithm validation were executed using the same vehicle. The dataset was meticulously gathered at Shoemaker Green, UPenn, owing to its ideal unstructured terrain, aligning perfectly with the challenges our project addresses. The environment boasted uneven grassy terrain dotted with natural obstacles like trees and patches of rough mud. This selection factored in the car's dynamics, ensuring our trajectory remains

feasible for navigation. In total, we collected 1067 images, capturing them at a frame rate of 30 frames per second. To strike the right balance between data quantity and quality, we opted to record images at 1 Hz. This deliberate approach ensures the feasibility of stitching consecutive frames to construct a seamless trajectory in the future. Continuous trajectories were saved for future analysis. For data acquisition, we developed a publisher-subscriber node in ROS2, facilitating the collection of both RGB and depth images at the desired rate. These images were saved within the callbacks of the respective RGB and depth callback functions. Additionally, we recorded ROSbags to store data from other pertinent topics beyond the camera topics (/camera/rgb/image_raw and /camera/depth/image_raw), such as /odom, enriching our dataset for potential future analysis. Figure 1b is a visual representation of the comprehensive data collection process.

Traversability Prediction: For the computation of traversability coefficients, we employed the TRAVNET framework [10]. To facilitate this computation, we trained a model on the dataset supplied by the authors of TRAVNET. This decision was motivated by two primary factors. Firstly, the dataset encompassed a diverse range of environments, spanning locations in Puerto Rico and the USA, with terrain ranging from grassy landscapes to snowy regions. Therefore, training the model on this dataset ensured better generalization across various environmental conditions. Additionally, the dataset provided by the authors comprised approximately 16,000 images collected over two hours for training purposes. In contrast, our dataset consisted of around 1,067 images acquired within 15 minutes, encompassing both color and depth images. This model is a U-Net architecture with a ResNet18 backbone, combining information from RGB and depth images. It consists of an encoder, a bottleneck layer, and a decoder. The encoder extracts features from both RGB and depth images through ResNet blocks, the bottleneck compresses the features into a latent space, and the decoder reconstructs the final output by upsampling and concatenating features from different encoder stages.

Waypoint Generator: The traversability prediction model outputs an image with each pixel containing the value of "Traversability". This is an output is an HxW image with values ranging from 0 to 1. Our approach now generates waypoints to guide the robot by leveraging the above traversability map from the vision-based terrain assessment model. The key steps involve extracting traversable regions through image processing techniques like contour detection and computing the centroids of these regions as intermediate waypoints. To ensure smooth navigation, we interpolate additional waypoints between consecutive centroids using linear interpolation, creating a sequence of evenly spaced points along the line segments connecting the centroids. The number of interpolated points can be adjusted based on the desired path resolution and computational constraints. The resulting set of waypoints, comprising both centroids and interpolated points, defines the planned path for the robot to follow, enabling safe navigation within identified traversable areas of the unstructured environment as shown in 4a and 4b.

Waypoint Converter Waypoints generated using the traversability map would be pixels in the image. These were converted into the car frame with the integration of the intrinsic and extrinsics obtained during the calibration process of the RealSense 435i mounted on the F1Tenth. This is done by using the formula:

$$\begin{bmatrix} x_{car} \; y_{car} \; z_{car} \end{bmatrix}^T = R_{cam}^{car} \begin{bmatrix} K \end{bmatrix}^{-1} \begin{bmatrix} u \; v \; 1 \end{bmatrix}^T$$

The calibration matrix K was derived by capturing images of a checkerboard pattern from various angles using the RealSense camera. Utilizing the "calibrateCamera" function from the OpenCV (cv2) library facilitated this process. For extrinsic parameters, specifically the rotational component, we leveraged frames obtained through the URDF model of the car. This allowed us to calculate the corresponding rotation matrix. Since there is little offset between the camera and the baselink of the car, calculating the translational component of the transformation matrix was unnecessary.

Dynamic Path Controller (DPC): The DPC algorithm is a geometric path-tracking method that enables accurate and responsive steering control for mobile robots. It operates by continuously steering the robot towards a dynamically assigned target point on the desired path. The target point is selected as the point on the path that lies within a predefined look-ahead distance from the robot's current position. In our implementation, we integrated the DPC algorithm with the traversability maps generated by the TRAVNET model. We leverage these traversability maps to extract feasible paths and generate waypoints for navigation. The Dynamic Path Controller operates as follows:

Proximal Waypoint Calculation: At each control cycle, the algorithm calculates the proximal waypoint by projecting a look-ahead distance (L) along the planned path, starting from the robot's current position. The look-ahead distance is dynamically adjusted based on the robot's speed (v) to ensure stable and smooth tracking, using the formula:

$$L = k_l \cdot v + L_0$$

where k_l and L_0 are tunable constants.

Steering Control: The DPC algorithm computes the curvature (κ) required for the robot to reach the proximal waypoint from its current position and orientation. This curvature is calculated as:

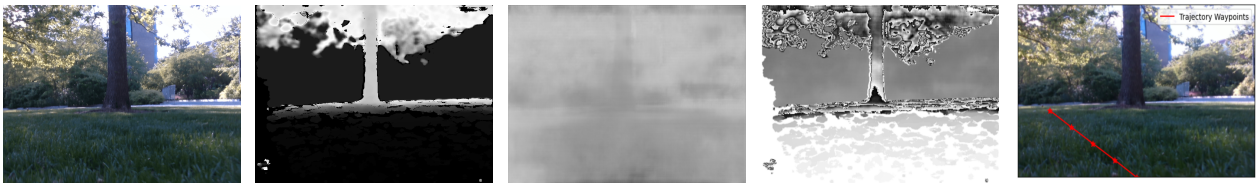
$$\kappa = \frac{2\sin(\theta)}{L_d}$$

where θ is the angle between the robot's current orientation and the line connecting its position to the pursuit point, and L_d is the distance to the pursuit point. The curvature value is then used to determine the appropriate steering angle (δ) for the robot's wheels using the relation:

$$\delta = \tan^{-1}(\kappa \cdot L)$$

Where κ is the curvature and L is the wheelbase of the robot.

Path Updating: As the robot navigates through the environment, the traversability maps are continuously updated based on the latest camera input. Consequently, the planned path and the corresponding waypoints are dynamically adjusted to account for changes in the terrain or the presence of new obstacles. By tightly coupling the DPC algorithm with the vision-based traversability prediction, our system can effectively navigate through unstructured outdoor environments without relying on predefined maps or geometrically defined obstacles.



(A) RGB (B) Depth (C) Inference (D) Post-processed (E) Waypoints

FIGURE 2. Sample images from the self-collected dataset, corresponding inference image from trained TRAVNET model, post-processed image after inference, and way-point generated image.

EXPERIMENTAL RESULTS

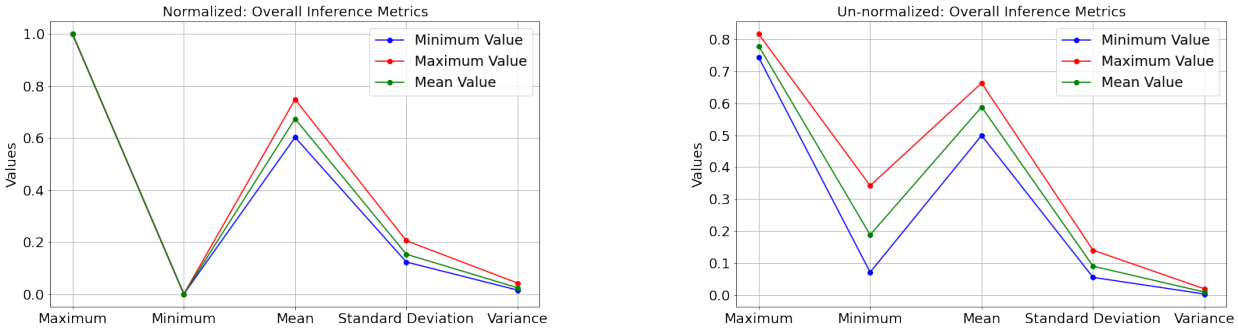
The pipeline successfully generated traversable waypoints for approximately 85% of the test images (self-collected data). An example output is shown in 4a. Traversability coefficients span from 0 to 1 for each pixel. We designate regions with coefficients above 0.65 as traversable. Given the varying traversability of terrain types, such as grassy lawns being more traversable than snowy regions, it is imperative to adjust the traversability allowance parameter accordingly based on the specific terrain to be traversed (Figure 3).

In 3, we present the statistics of the TRAVNET output images. We highlight the difference between the two sets: un-normalized and normalized images. The difference in the mean, standard deviation, and variance for the two sets is negligible. These graphs are also provided to statistically back our decision of choosing 0.65 as the traversability allowance parameter.

After obtaining the waypoints and trajectory from the trajectory generator, we used Dynamic Path Control (DPC) to navigate these points effectively. The DPC has some adjustable settings we needed to fine-tune. Firstly, we adjusted the K_p parameter to control steering. Increasing the K_p made turns sharper while decreasing it made turns smoother. Secondly, we played with the lookAhead setting. A smaller value made our car react more quickly to changes in the path, while a larger value made the car follow a smoother path but might lead to some tracking errors. Lastly, we adjusted the speed of the vehicle to match our exploration goals. We chose a slower speed to ensure the car navigated the waypoints accurately. After testing, we settled on these settings: K_p = 0.5, lookAhead = 1.2m and speed=1.0m/s.

The pipeline was deployed offline. Once the waypoints required for the DPC were generated offline, we deployed the DPC online to validate the waypoints generated and control inputs from the DPC. An additional camera mounted on the F1Tenth car while this experiment shows the performance of our pipeline. Videos of the same are available on the project GitHub repository provided in the Appendix section. From our experiments on the real car, we inferred that the range of waypoints generated covered a minimal distance. This requires improvement by stitching the sequence of images in the trajectory for longer traversal. Moreover, the car could comfortably traverse through muddy regions around trees. We noticed that sometimes the car would stop unexpectedly due to mechanical limits and debris getting

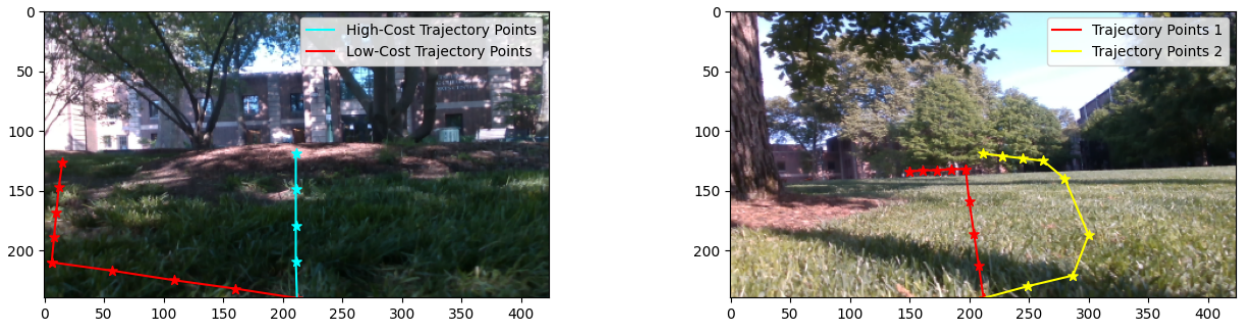
stuck in the axle of the car. Also, the speed of the car near the muddy region was high and caused the car to skid/topple when the obstacle was large. We plan to integrate adaptive speed control to address this limitation.



(A) Normalised Inference Metrics

(B) Un-normalised Inference Metrics

FIGURE 3. Inference data metrics for the self-collected dataset from the trained TRAVNET.



(A) Waypoint path generated by the framework considering horizon as the goal.

(B) Multiple trajectories generated by the framework and their respective way-points for a given RGBD image.

**The 'High-Cost Trajectory Points' refer to the path generated with relatively small traversability coefficients. This implies that the region these waypoints cover is relatively rough and tough to traverse. The 'Low-Cost Trajectory Points' refer to the path generated with relatively higher traversability coefficients. The region these waypoints cover is relatively smooth and easy to traverse.*

FIGURE 4. Comparison of waypoint paths and multiple trajectories.

FUTURE WORK AND DISCUSSION

While our vision-based approach for autonomous navigation in unstructured outdoor environments has shown promising results for short distances (limited waypoints), several areas require further exploration and enhancement. Key challenges include the occasional generation of waypoints in non-traversable regions and the presence of noise, leading to waypoints extending beyond the visible horizon. These issues likely stem from limitations in the traversability prediction model or the waypoint generation algorithm. Future work should aim to improve the robustness and accuracy of the traversability prediction model by incorporating additional input modalities and exploring advanced neural network architectures. Currently, traversability maps are generated offline on a separate system, and the planned trajectories are then fed to the robot. To enable truly autonomous operation, we plan to deploy the entire pipeline, including traversability prediction and path planning, on the robot itself. This will involve optimizing algorithm efficiency and leveraging hardware acceleration techniques, such as TensorRT, for real-time inference and path planning on the embedded platform. Currently, we focus on exploratory path generation without specific start or endpoints. In the future, we aim to integrate GNSS data to provide initial and goal waypoints, enabling optimal trajectory planning while accounting for local obstacles. Incorporating local obstacle avoidance will further enhance the system's ability to navigate safely in dynamic environments, mitigating potential errors in the traversability prediction model. This project has provided a foundation for future work in the field of unstructured environment traversal using the F1Tenth car setup.

ACKNOWLEDGEMENTS

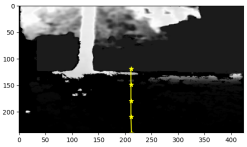
We extend our gratitude to Prof. Pratik Chaudhari for directing us towards pertinent resources to shape our project framework. Our appreciation also goes to the Teaching Assistants of ESE6500 for their invaluable feedback on our project proposal. Additionally, we express our thanks to Prof. Rahul Mangharam and the XLab team member for furnishing us with resources and technical assistance to execute our project on the F1Tenth car.

REFERENCES

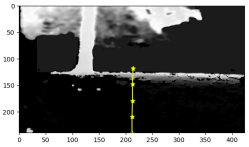
(1) Philion, Jonah, and Sanja Fidler. "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d." Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. Springer International Publishing, 2020.
(2) Hu, Anthony, et al. "Fiery: Future instance prediction in bird's-eye view from surround monocular cameras." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
(3) Shore, Tavis, Simon Hadfield, and Oscar Mendez. "BEV-CV: Birds-Eye-View Transform for Cross-View Geo-Localisation." arXiv preprint arXiv:2312.15363 (2023).
(4) Mirowski, Piotr, et al. "Learning to Navigate in Complex Environments." arXiv preprint arXiv:1611.03673, 2016.
(5) Bowman, Sean L., et al. "Probabilistic data association for semantic slam." 2017 IEEE international conference on robotics and automation (ICRA). IEEE, 2017.
(6) McCormac, John, et al. "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks." 2017 IEEE International Conference on Robotics and automation (ICRA). IEEE, 2017.
(7) Wang, Yan, et al. "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.
(8) Meng, Xiangyun, et al. "Terrainnet: Visual modeling of complex terrain for high-speed, off-road navigation." arXiv preprint arXiv:2303.15771 (2023).
(9) Kahn, Gregory, Pieter Abbeel, and Sergey Levine. "Badgr: An autonomous self-supervised learning-based navigation system." IEEE Robotics and Automation Letters 6.2 (2021): 1312-1319.
(10) Gasparino, Mateus V., et al. "Wayfast: Navigation with predictive traversability in the field." IEEE Robotics and Automation Letters 7.4 (2022): 10651-10658.
(11) Gasparino, Mateus Valverde, Arun Narenthiran Sivakumar, and Girish Chowdhary. "WayFASTER: a Self-Supervised Traversability Prediction for Increased Navigation Awareness." arXiv preprint arXiv:2402.00683 (2024).

APPENDIX

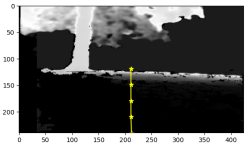
(1) **Trajectory Generation** These are the consequent frames and our trajectory/ waypoint generation algorithm in action.



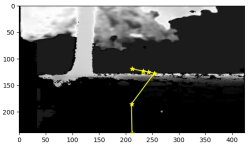
(A) Frame 1



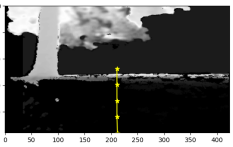
(B) Frame 2



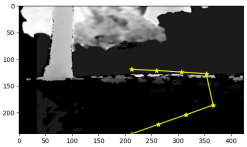
(C) Frame 3



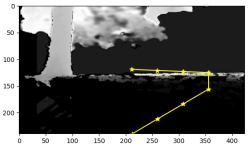
(D) Frame 4



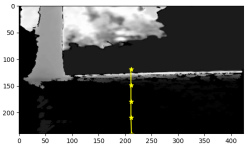
(E) Frame 5



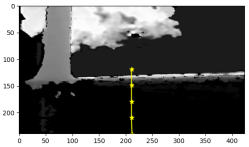
(A) Frame 6



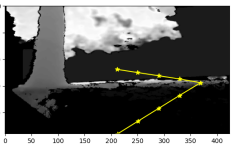
(B) Frame 7



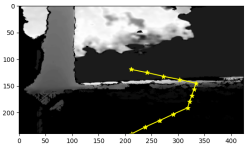
(C) Frame 8



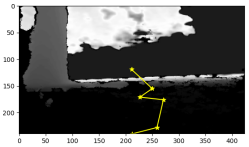
(D) Frame 9



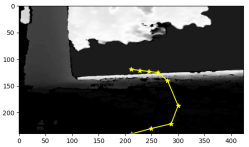
(E) Frame 10



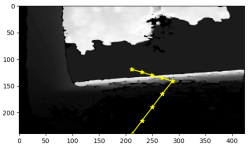
(A) Frame 11



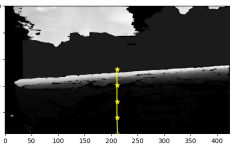
(B) Frame 12



(C) Frame 13



(D) Frame 14



(E) Frame 15

(2) You can find the code repository on GitHub: GitHub Repository.