BIRD'S EYE VIEW GENERATION AND LANE DETECTION USING MULTI-CAMERA SETUP

CH SRICHERAN
Artificial Intelligence and Data Science
22011101023
22110383
sricheran22110383@snuchennai.edu.in

ADITYA V
Artificial Intelligence and Data Science
22011101007
22110420
aditya22110420@snuchennai.edu.in

BAVATHAYINI N
Artificial Intelligence and Data Science
22011101021
22110283
bavathayini22110283@snuchennai.edu.in

***Abstract*—In autonomous driving, gaining a top-down understanding of a vehicle's environment significantly improves decision-making and safety. This paper introduces a computer vision approach to generate Bird's Eye View (BEV) images from six camera inputs using the nuScenes dataset. By utilizing intrinsic and extrinsic camera calibration parameters, combined with a perspective transformation, our method projects 2D camera frames into top-down BEV representations. The system supports real-time video display of both original and transformed frames, offering enhanced visualization of road geometry and object layout without relying on LiDAR or 3D point cloud data.**

***Index Terms*—Bird's Eye View (BEV), Perspective Transformation, Camera Calibration, nuScenes Dataset, Autonomous Driving, Computer Vision, Homography Matrix, Image Warping, Extrinsic and Intrinsic Parameters, Surround-View Cameras.**

I. Introduction

The development of autonomous vehicles has driven innovation in perception systems capable of interpreting real-world scenes in real time. A top-down or Bird's Eye View (BEV) representation provides a more intuitive spatial layout for vehicle navigation, lane detection, and path planning. In this work, we propose a method to transform monocular images from six surround-view cameras (CAM_FRONT, CAM_FRONT_LEFT, CAM_FRONT_RIGHT, CAM_BACK, CAM_BACK_LEFT, CAM_BACK_RIGHT) into BEV format using geometric transformations and camera calibration data from the nuScenes dataset. The approach bridges the gap between 2D image capture and 3D scene reasoning without relying on expensive sensors such as LiDAR.

II. Dataset Description

This project uses the `v1.0-mini` version of the **nuScenes dataset**, a publicly available benchmark for autonomous driving research. The dataset provides synchronized data from multiple sensors mounted on a vehicle, including cameras, LiDAR, RADAR, and GPS/IMU.

For this study, we focus on the six surround-view cameras: `CAM_FRONT`, `CAM_FRONT_LEFT`, `CAM_FRONT_RIGHT`, `CAM_BACK`, `CAM_BACK_LEFT`, and `CAM_BACK_RIGHT`. Essential metadata files such as `sensor.json`, `calibrated_sensor.json`, `ego_pose.json`, and `sample_data.json` are utilized to extract camera parameters and maintain frame-wise synchronization. These parameters enable the application of homographic transformation to generate Bird's Eye View (BEV) visualizations from monocular images captured from all six directions.

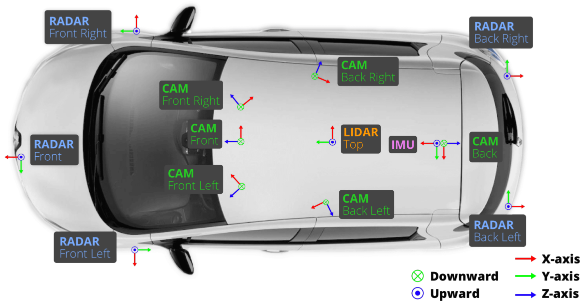

Fig. 1. Car Setup

Camera Specifications

The dataset utilizes a 6-camera setup consisting of Basler acA1600-60gc cameras. The specifications of the cameras are as follows:

- **Capture Frequency:** 12 Hz
- **Lens:** Evetar Lens N118B05518W, F1.8, f = 5.5 mm, 1/1.8"
- **Sensor:** 1/1.8" CMOS sensor with a native resolution of 1600 $\times$ 1200 pixels
- **Image Format:** Bayer8 format (1 byte per pixel encoding)
- **Region of Interest (ROI):** Cropped to 1600 $\times$ 900 pixels from the original resolution to reduce processing and transmission bandwidth
- **Exposure:** Auto exposure enabled with exposure time limited to a maximum of 20 ms
- **Image Processing:** Captured images are unpacked to BGR format and compressed using JPEG

The orientation and overlap of the six camera views are illustrated in the figure below.

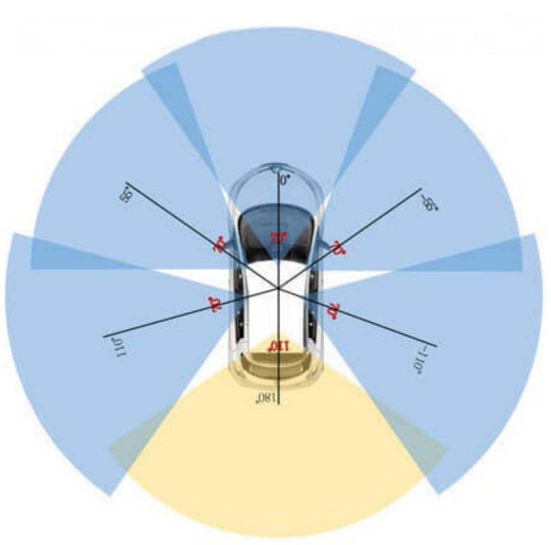

Fig. 2. Camera orientation and overlap

Pixel to World Coordinate Transformation

The workflow converts 2D pixel coordinates from a front-facing camera image into 3D world coordinates using data from the *nuScenes* dataset. The process begins by loading essential metadata files (`sensor.json`, `calibrated_sensor.json`, `ego_pose.json`, and `sample_data.json`) to access sensor calibration parameters and ego vehicle poses. A camera image from the sensor is selected, and its intrinsic parameters (camera matrix) along with extrinsic parameters (rotation and translation between camera and ego vehicle) are retrieved. The global position and orientation of the ego vehicle are also extracted from the ego pose data. Pixel coordinates are first projected into 3D camera rays using the inverse of the intrinsic matrix. These rays are then transformed into the ego vehicle coordinate frame using the camera's rotation and translation, and further mapped into the global world frame using the ego vehicle's pose. The Euclidean distance between each mapped world point and the ego vehicle's global position is calculated to determine spatial relationships. Finally, the selected image is visualized by marking the input pixels and annotating each with the computed distance, effectively demonstrating the transformation from image space to world space.

Image Stitching Using Feature-Based Homography

Image stitching is achieved by aligning and merging two adjacent camera views.The process performs a comprehensive image stitching operation using three camera inputs from a multi-sensor autonomous vehicle setup: **CAM_FRONT**, **CAM_FRONT_LEFT**, and **CAM_FRONT_RIGHT**. The objective is to generate a panoramic view of the front region by aligning and blending these images. Initially, images from all three cameras are loaded and verified. Then, the Scale-Invariant Feature Transform (SIFT) algorithm is applied to detect keypoints and compute feature descriptors, which are matched using a FLANN-based matcher. Lowe's ratio test filters the best matches, and if enough good matches are found, homography matrices are estimated using RANSAC to map the perspectives of CAM_FRONT_LEFT and CAM_FRONT_RIGHT to that of CAM_FRONT. These matrices are further modified using a translation matrix to place the front camera image in the center of a larger canvas, leaving space for the other views. The left and right images are warped accordingly and aligned beside the front image. A custom blending function overlays the warped images onto the canvas using binary masks to avoid visual artifacts. The final output is a seamlessly stitched image showing a wide-angle view combining the left, front, and right perspectives, aiding better environmental perception for autonomous systems.

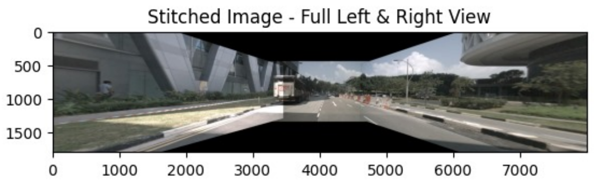

Fig. 3. Stitched Image

Bird's Eye View Generation using Perspective Transformation

To generate a Bird's Eye View (BEV) from the stitched multi-camera image, a perspective transformation is applied. A trapezoidal region representing the road surface in the stitched image is defined using four source points. These points are carefully selected based on the region of interest captured from the vehicle's front and side-facing cameras. A corresponding rectangular region in the destination plane is specified to represent the top-down BEV perspective. Using OpenCV's `getPerspectiveTransform()` function, a homography matrix is computed to map the trapezoid to the rectangle. The stitched image is then warped using `warpPerspective()` to obtain the final BEV output. Additionally, the selected trapezoid is visually overlaid on the stitched image for reference. This transformation provides a geometrically corrected top-down view of the surroundings, which is essential for autonomous vehicle perception and scene understanding.

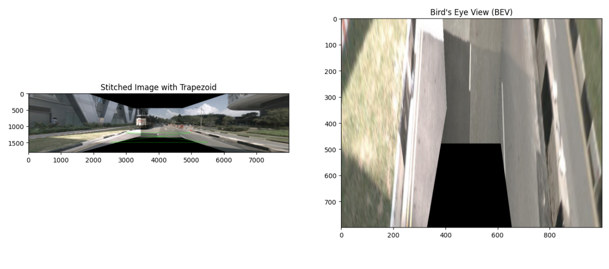

Fig. 4. Birds Eye View (BEV) Image of Front Camera

Multi-Camera Stitching and BEV Generation

Real-time BEV generation is achieved by stitching images from the front, front-left, and front-right `nuScenes` cameras.

Frames are grouped by timestamp and loaded using metadata from `sample_data.json`. SIFT and FLANN-based feature matching are employed to compute homographies that align the side views with the front view. Each image is warped using `warpPerspective()` and blended onto a larger canvas using a translation matrix to maintain spatial alignment. After stitching, a fixed perspective transformation using `getPerspectiveTransform()` maps a defined trapezoidal road region to a rectangular BEV space. The final top-down view is generated by applying this transformation to the stitched image and displayed frame-by-frame in a real-time OpenCV window.

Similarly, the same process is performed for the rear-facing cameras `CAM_BACK`, `CAM_BACK_LEFT`, and `CAM_BACK_RIGHT` to generate the rear Bird's Eye View.

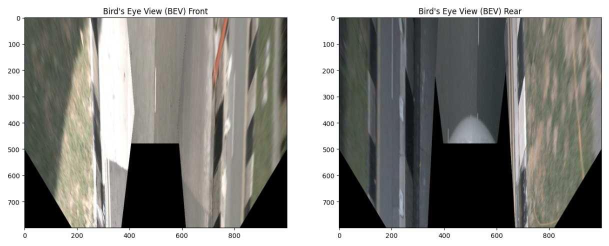

Fig. 5. BEV (Front and Rear) Image

These homographies are then adjusted using a translation matrix T to map all camera views to a larger panoramic canvas. Each frame undergoes perspective transformation via OpenCV's `warpPerspective()` and `getPerspectiveTransform()` to generate a top-down Bird's Eye View (BEV). The warped left/right images are blended with the base front/back images using bitwise masking, and the two BEV halves (front and flipped back) are stacked vertically. An ego-vehicle overlay (transparent PNG) is resized and positioned at the center of the BEV to simulate the car's location. Lastly, the image is cropped and resized to a square resolution for visualization.

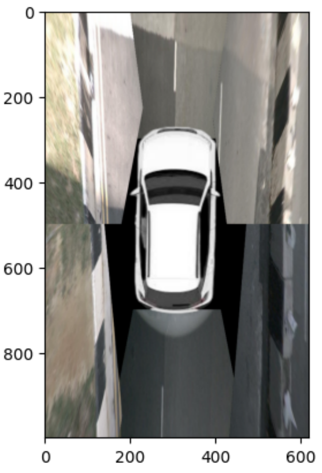

Fig. 6. Birds Eye View (BEV) Image of 6 Cameras

A. Lane Detection and Overlay

To enhance perception, the BEV image undergoes a lane detection pipeline. A central horizontal region from the image is cropped and processed to highlight lane markings. The pipeline converts the region to grayscale, applies Gaussian blur to reduce noise, and detects edges using Canny edge detection. Subsequently, the Hough Line Transform (`HoughLinesP`) identifies linear lane segments, which are redrawn over the original BEV using `cv2.line()` with green strokes. The overlayed result shows the detected lanes within the BEV, providing critical geometric cues for vehicle path planning and autonomous driving perception tasks.

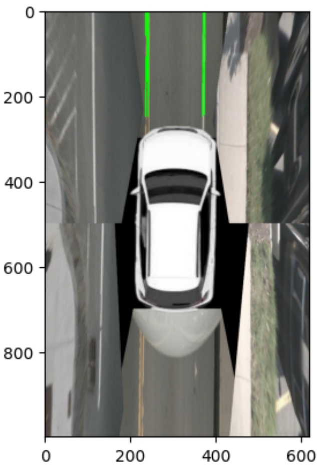

Fig. 7. BEV with lane detection

Future Works

Future improvements aim to make the system more efficient and suitable for real-time applications. Optimizing the pipeline to be lightweight will support deployment on embedded automotive hardware. Deep learning models like **U-Net** can be integrated to perform more accurate segmentation of **lanes, parking slots, and pedestrian crossings**.

Additionally, incorporating **temporal information** across frames can improve detection stability, and **sensor fusion** with LiDAR or RADAR can enhance spatial awareness. These upgrades will help create a more seamless, accurate, and robust BEV perception system for autonomous driving.

References

[1] H. Qin, Y. Zhang, X. Zhou, M. Wang, H. Zhu, and D. Dai, "PanoBEV: Panoramic Bird's-Eye-View for Surround View Perception," *arXiv preprint arXiv:2409.13912*, 2024.

[2] MathWorks, "Create 360-Degree Bird's-Eye View Image," *MATLAB & Simulink Documentation*.

[3] Z. Chen, K. Li, and Z. Wang, "Surround-View Camera System Calibration and 3D Measurement for ADAS Applications," *arXiv preprint arXiv:2007.01813*, 2020.