Twitter: **T**opdown **WIT**h **T**ransformation Using **E**gocentric **R**GB Images

Sundar, Akash Goel, Ayush Xie, Jason

Venkatesh, Sharanya

Project Summary

In video games, we often view the scene from the top as if we are birds observing the environment. We can also do this in the real world with some clever camera tricks. Many automobile manufacturers in recent years have been using top-down 360 views to assist the driver in parking. Top-down views can also be used for downstream automation tasks such as parking assist and autonomous driving. Thus, this project pertains to the transformation of multiple images captured in the world to a single bird-eye view.

Keywords— Bird's Eye View (BEV), Homography, Inverse Perspective Mapping

Goals and Objectives

- Given a single image from an egocentric RGB camera, transform the image to its corresponding top-down view.
- From the top-down view of multiple images, stitch these images and apply blending to form a single coherent top-down view.

Related Works

There is plenty of research on Advanced Driver Assistance Systems (ADAS) using both classical and neural approaches. [1] provides an overview of the hardware and software involved in implementing driver assistance systems. The authors in that paper also discuss applying vision for downstream tasks such as blindspot monitoring, parking spot recognition, and pedestrian tracking. One work that is directly related to our project proposal is [2], where birds-eye-views are generated via a Generative Adversarial Network trained on egocentric/top-down image pairs taken from GTA5.

Methodology

Our main objective of providing the birds eye view of a scene was tackled in multiple stages. The first step was to perform extensive literature review and identify related

works. We then chose one of the papers [3] to be a good starting point. It was then important to identify a good dataset that consisted of the required images. We settled on two popular open source datasets, namely, Kitti and Apollo. We used both over various stages of the processing pipeline to obtain intermediate results.

We proceeded with an initial homographic transformation of the scene onto the birds eye view plane. One problem seen with this approach is that the planar assumption breaks down when cars and non-planar objects are present in the scene. The drawbacks of the results could be tackled by approximating the data in the image using template images as observed in [4]. Hence, the transformation problem now simplifies to object identification and depth estimation. In our final pipeline, we also used a lane detection estimator along with a bounding box object detector to estimate the features on the road and approximated their depths using a depth map.

Inverse Perspective Transform

We considered a classical approach to the problem of BEV generation using the inverse perspective transform as proposed by Reiher et al. [3].

Let the original camera frame be $\mathbf{C}$, and the world frame be $\mathbf{W}$. We define the transformation from $\mathbf{C}$ to $\mathbf{W}$ to be $[\mathbf{R}^{\mathbf{C}}_{\mathbf{W}}|\mathbf{t}^{\mathbf{C}}_{\mathbf{W}}]$. Let $\mathbf{p_w}$ be an imaged point in world coordinates and let $\mathbf{p_i}$ be the image pixel. The camera projection equations is:

$$\mathbf{p_i} = \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{p_w}. \tag{1}$$

Assuming a flat road plane, the transformation from the road plane to a point in the world frame is given by:

$$\mathbf{p_w} = \mathbf{M}\mathbf{p_r}, \tag{2}$$

Where

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

Substituting this expression into the projection equation, we see that the operation is analogous to a homography transformation:

$$\mathbf{p_r} = (\mathbf{PM})^{-\mathbf{1}}\mathbf{p_i}. \tag{4}$$

Following re-projection of images into the top-down view, we can attempt to stitch images together using feature matching. Further blending can be done to the stitched images to enhance quality.

We have implemented the inverse perspective transform on sample images taken from the Kitti Dataset. However, as is clearly visible, the resultant output suffers from major distortion. Objects farther from the camera are often mapped into the same pixel and hence information is lost and objects closer to the camera are elongated to unrealistic dimensions. Hence, there was need to consider alternate approaches.

Alternate Approach - Recreate the scene

From extensive literature review and based on our results from the previous section, we concluded that there was no foolproof method to reproject the scene. Hence, we

Figure 1: Original Sample Image from Dataset

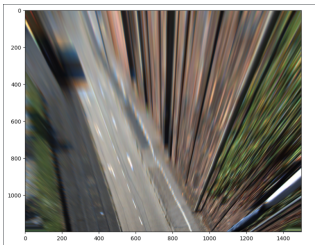

Figure 2: Projected Image

followed an approach of functionality over accuracy. We chose a template image of a car and accurately determined its position in the top view using methodologies as explained in further sections. Following the basic thought process of Palazzi et al. [4], the final image consisted of car images in the same location as in the real world but not images of the same cars, hence giving the user an accurate description of the real world scenario.

Using Depth Maps and Segmentation Masks

The base approach of homographic reprojection of the scene was retained. The input to the algorithm was the segmented background area. This effectively removes all objects over the road plane from the image. The effective reprojection then simply consists of unadulterated picture of the road alone.

Onto this it was important to determine the positions of the cars in the birds eye view of the image. The depth map of the image was generated from disparity maps generated from the OpenCV function *cv.StereoBM_create* . A horizon thereshhold was defined and the intensity values in the depth map very mapped to real world lengths in the top view. The centres of each of the cluster of intensity values that were used to represent each car were identified using *KMeans* and this was used as the effective depth to represent the entire template image.

Drawbacks of this approach: The approach requires a well built depth map and its cluster centres are based on running the KMeans algorithm. If the actual boundaries of the car are not well defined in the figure, the clusters formed are not fully representative of the scene. Hence when mapping a scene, two or more clusters could be generated on objects closer to the observer, covering a larger area while some others may be clubbed together to form a single cluster. Hence a more accurate method of representing each

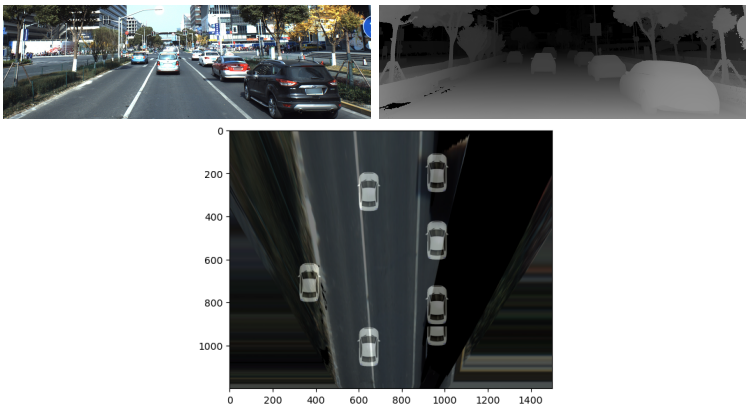

Figure 3: Scene Reconstruction using Depth Maps and Segmentation Masks (a) Input Image (b) Depth Map (c) Image Reconstruction

object in the scene separately was needed.

Using Bounding Boxes and Instance Segmentation

Instance Segmentation was implemented using *ResNet50* to segregate each object in the image separately. Simultaneously, to overcome the problem faced using depth maps of being unable to identify the geometric centre of some objects due to occlusion, a bounding box was built over the estimated geometry of the object and its centre was used to identify the distance of the object in the birds eye view. However, drawing bounding boxes over images created dark patches in the final image. Instead of bounding boxes, we opted for instance segmentation, which creates less dark areas in the result image by masking out only the object of interest in the scene. To determine the correct location of the car in the top down view, we can create a bounding box from the image mask and remap its coordinates to the final image.

Lane Detection and Segmentation of Driveable Area

In addition to car detection, we also incorporated drivable area segmentation and lane detection into the top down image using YOLO P. YOLOP takes a RGB image as input, and outputs lane markings and the drivable area as separate boolean masks. We can further process this output by applying image dilation and erosion, removing noisy outputs. Finally, the inverse perspective transform is applied to the lane and drivable area image to produce an annotated birds-eye view image. This can be useful for downstream tasks such as lane keep assist.

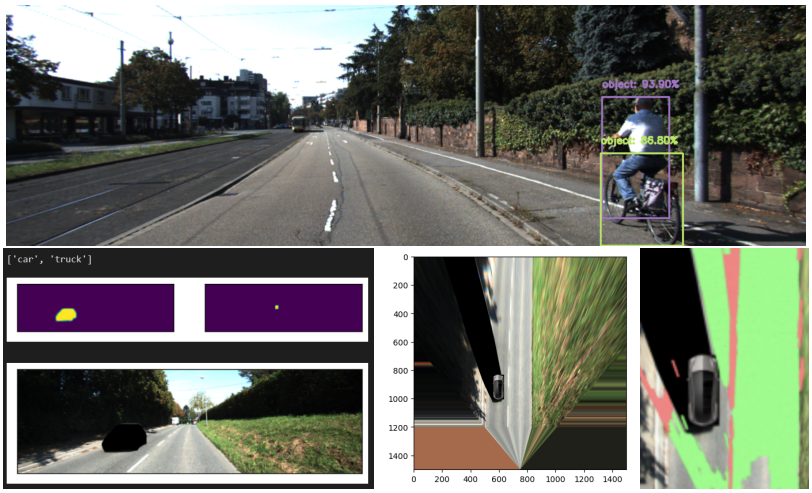

Figure 4: Scene Reconstruction using Bounding Boxes and Instance Segmentation (a) Bounding Box (b) Instance segmentation (c) Reprojection with instance segmentation mask (d) Final output with lane detection and drivable area segmentation

Consolidated Pipeline and Final Results

By combining instance segmentation, car reconstruction, lane detection, and drivable area segmentation, we were able to reconstruct the car's forward view from a top-down perspective using egocentric images from the Kitti dataset.

Author Contribution

All authors contributed equally towards the development of this pipeline.

Future Plans

In our current results, the direct stitching of projected images does not make semantic sense because the frames are temporally spaced and the overlap is minimal to extract any meaningful information and perform the stitching. To overcome this, we are currently exploring NeRF based synthesis to attempt to reconstruct novel views and potentially use it to synthesise information to make up for the loss of information in between frames. In parallel, we will also carry out work in the direction of extracting depth information from the images so as to perform locally suitable projective transformations.

References

[1] M. Heimberger, J. Horgan, C. Hughes, J. McDonald, and S. Yogamani, "Computer vision in automated parking systems: Design, implementation and challenges," *Image and Vision Computing*, vol. 68, pp. 88–101, 2017, automotive Vision: Challenges, Trends, Technologies and Systems for Vision-Based Intelligent Vehicles. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885617301105

[2] "Generating bird's eye view from egocentric rgb videos," *Generative Adversarial Networks for Multi-Modal Multimedia Computing*, vol. 2021, 2021. [Online]. Available: https://www.hindawi.com/journals/wcmc/2021/7479473/

[3] L. Reiher, B. Lampe, and L. Eckstein, "A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–7.

[4] A. Palazzi, G. Borghi, D. Abati, S. Calderara, and R. Cucchiara, "Learning to map vehicles into bird's eye view," in *Image Analysis and Processing - ICIAP 2017*, S. Battiato, G. Gallo, R. Schettini, and F. Stanco, Eds. Cham: Springer International Publishing, 2017, pp. 233–243.