Vehicular pose estimation and shape reconstruction

Suren Sritharan

1. Introduction

In recent years, the emergence of new techniques in computer vision, sensor technology, and machine learning has led to innovations in a wide range of domains for understanding and interacting with the physical world. One such remarkable applications of this convergence is the field of vehicular pose estimation and 3D shape reconstruction. This discipline lies at the intersection of computer vision, robotics, and automotive engineering, and offers solutions for a wide range of industries including autonomous driving, augmented reality, digital twins etc. The problem of vehicular pose estimation and 3D shape reconstruction is composed of 2 sub problems which are solved in parallel.

First, Vehicular pose estimation refers to the process of determining the precise position and orientation of a vehicle in its environment. On the other hand, 3D shape reconstruction plays a crucial role in comprehending the surrounding environment in three dimensions. This information is pivotal for various tasks such as navigation, obstacle avoidance, and accurate map generation. Traditional methods often relied on GPS data, inertial sensors, and laser scanning of 3D models. However, these approaches are limited in accuracy, reliability and efficiency. However, through the use of advanced deep learning techniques and and sensor fusion, vehicles can now utilize cameras, LiDAR (Light Detection and Ranging), radar, and IMUs (Inertial Measurement Units) to achieve higher precision in pose estimation and 3D object reconstruction.

Many recent works have shown that fusion techniques perform better in comparison. However, the use of fusion techniques pose other challenges in regards to the feasibility, communication and efficiency. Furthermore, some prior works have relied on specific devices such as LiDAR and stereo vision. However, as the system becomes complex and costly, it quickly reaches the limit to scalability. In comparison, 3D perception using monocular vision has shown to overcome these issues, and has gained interest due to its simplicity and feasibility.

While many works on 3D perception focus on bounding box detection, this project focuses on reconstructing exact object shapes. Joint pose estimation and shape reconstruction is vital in recovering fine object details, which can significantly improve downstream tasks such as digital twin creation. Many of the recent works have similar foundational ideas in which they use supervised training methodology. This creates a dependence on the availability of labeled dataset which contains exact 3D shape annotations. However, the lack of extensive labeled dataset with a wide range of classes has made shape reconstruction challenging task for diverse conditions. Therefore, this project focuses on moving towards a semi-supervised / self-supervised learning methodology, which would pave the way for unsupervised techniques.

The rest of this report is organized as follows. Section 2 presents the related work in this field. Next, section 3 gives a brief overview of the available datasets. Section 4 discusses the short comings in existing works and describes the proposed solutions to overcome these shortcomings. Section 5 and 6 describes the experimental setup, and the results and analysis of theses experiments respectively. Finally, the report is concluded by summarizing the findings and discussing potential future directions.

2 Related work

Majority of the works in the realm of 3D vehicular object detection focus on identifying bounding boxes at scale. These methods can be categorized as either depth-assisted or image-only approaches. Depth-assisted methods use depth maps to aid 3D object detection, transforming them into 3D point clouds for LiDAR-based detectors. Various techniques such as CMKD [1], PatchNet [2], DDMP-3D [3], and DD3D [4] utilize depth information for improved

detection quality. Image-only methods without depth information focus on geometry priors for object detection. Methods like Deep3DBox [5], GUPNet [6], and MonoFlex [7] incorporate geometric relationships and uncertainty modeling to estimate object depth. Despite the abundance of these types of techniques, they fail to model detailed realistic 3D shapes which provide an exact intuitive representations of the objects.

In contrast, reconstructing the exact 3D mesh provides detailed information of each. instances' scales and orientations As specified previously, the problem of vehicular pose estimation and 3D shape reconstruction is composed of two sub problems, namely (1) pose estimation, and (2) shape reconstruction. These problems have been studies under different domains. 6D pose estimation refers to the problem of estimating the 6 degrees of freedom of an object given by rotation and translation components from a monocular image. In addition to monocular images, 6D pose estimation techniques often utilize fixed 3D models [8, 9, 10]. Different approaches have been proposed to solve the 3D shape reconstruction problem. The reconstruction could be either voxel-based [11] or mesh-based [12, 13], and the techniques can take different approaches such as offset based reconstruction [14], completely deformable spheres [13] or feature extraction based reconstruction [12, 11].

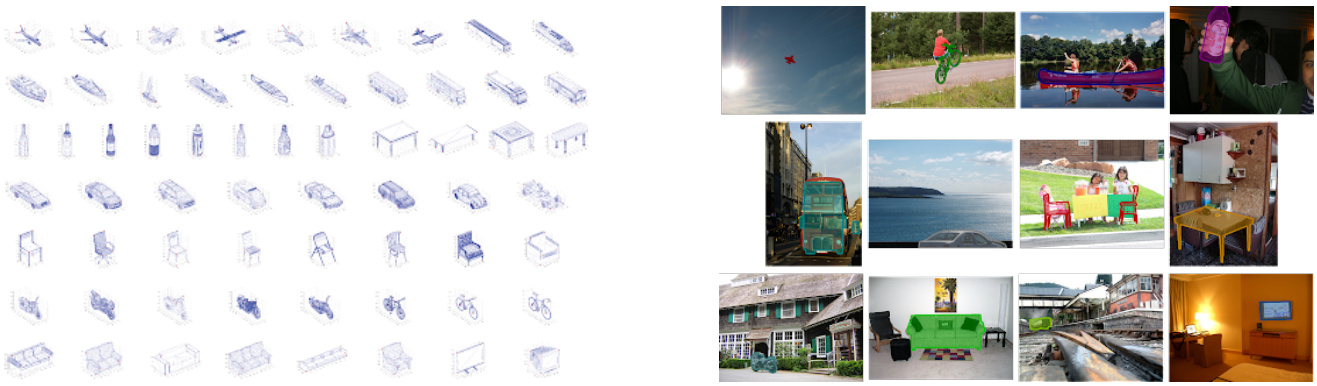
The techniques presented above solve the respective problems individually. However, recent advances in human objects have led to more interest in joint vehicular pose and shape estimation for 3D traffic scene understanding. DeepMANTA [15] reconstructs non-detailed skeleton through a coarse-to-fine retrieval strategy. 3D-RCNN [16] encodes the template 3D shapes as PCA parameters. These parameters are then concatenated with positional information and decoded to voxel representation. The reconstructed voxels are then projected to the image and trained using a render-and-compare scheme. In [17], the direct-based approach extends 3D-RCNN to utilize attention mask and offset flow. Similar to previous works, GSNet [18] uses 2D keypoints as additional features for the backbone and generates 3D shapes by blending multiple meshes from different PCA-basis. BAAM [19] uses shape priors encoded into template features instead of a PCA-basis which often leads to loss of details of object shape. In addition it proposes the use of attention mechanism to find the relevance between object and shape prior, and the use of global features for globally aware pose estimation. All of these works are dependant on 3D mesh annotations which are often unavailable in large scale vehicular datasets. Furthermore, the models utilize keypoints and shape templates which are specific to the car models, and such cannot be extended to other object classes. Therefore, the aim of this project is to remove this dependency and extend the applicability of these models to generic datasets consisting multiple classes and object types.

3 Dataset

With the rise of autonomous driving research, a wide variety of datasets have been introduced for 3D vehicular object detection and tracking. The most widely used datasets such as KITTI [20], NuScenes [21], Waymo [22], etc. contain a wide variety of image data, taken under different conditions. The annotations present in these datasets vary depending on the probleme at hand. However, these datasets do not contain detailed 3D shape annotations, and as such cannot be used for supervised training. On the other hand, the Pascal3D+ [23] dataset, and ShapeNet [24] dataset contains 3D CAD models of multiple object classes. The Pascal3D+ dataset also contains images of these objects taken in the wild, along with the object pose and shape annotations. The Apollo3D [17] dataset is the most commonly used dataset used for joint pose estimation and 3D shape reconstruction since it contains all the necessary annotations. As opposed to Pascal3D+ dataset, the images in Apollo3D dataset were taken from an ego-vehicle which is vital in the context of autonomous driving and scene understanding. A brief description of some of these datasets, along with their usage in prior works is described below.

3.1 KITTI

The KITTI dataset is the most well known autonomous driving dataset used for tasks such as object detection, tracking, and scene understanding. It contains a diverse set of sensor data collected from a moving vehicle equipped with multiple sensors, including high-resolution cameras, Velodyne lidar, and GPS/INS systems. However, in the context of this project, only RGB images taken from the camera are considered. The dataset provides a large number of annotated images of different urban driving scenarios. It includes labelled bounding boxes, segmentation masks, depths, etc. for objects of interest such as vehicles (cars, trucks, etc), pedestrians, and cyclists. In addition to the object annotations, the KITTI dataset provides other valuable information, such as precise 3D object poses, calibration data for sensor alignment, and camera parameters. Though the dataset does not contain 3D mesh



(a) Sample CAD models

(b) Annotations projected onto input images

Figure 1: Pascal3D+ dataset

annotations, prior works have used it in combination with other datasets such as Pascal3D+ [16] or Apollo3D [19] for training and evaluation.

3.2 Pascal3D+

The Pascal3D+ dataset is an extension of the Pascal VOC dataset to address the shortcomings in regards to 3D pose estimation and shape reconstruction. Pascal3D+ contains 12 categories of rigid objects selected from the PASCAL VOC 2012 dataset including cars, chairs, bicycles, and airplanes, among others as shown in Figure 1. These objects are annotated with pose information (azimuth, elevation and distance to camera). Prior works [16] have filtered out the cars category and used it for training and evalutaion.

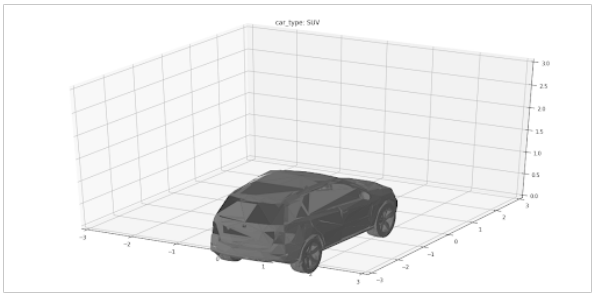
3.3 ShapeNet

The ShapeNet dataset is a significant and widely-used dataset for many computer vision research problems including 3D shape recognition, reconstruction, and generation. ShapeNet offers an extensive collection of labeled 3D models spanning a diverse range of object categories more than that offerred by Pascal3D+ dataset. A few sample images are shown in Figure 2, however, the lack of real-life images for the corresponding object shapes makes this dataset unsuitable for diverse applications. Prior works [11] have leveraged this dataset for shape reconstruction tasks.

Figure 2: Sample CAD models in ShapeNet dataset

3.4 Apollo 3D

The Apollo3D dataset contains 60,000 labeled 3D car instances from 5,277 real-world images, based on industry-grade CAD car models of 79 types. On average, each ApolloCar3D image contains 11.7 car objects described by 2D keypoints, 3D translation (x, y, z), and rotation (yaw, pitch, roll) labels. Each object is one of 79 car classes (e.g. sedan, coupe, SUV, and so on) and one such sample is shown in Figure 3. Many of the prior works [19, 18] on 3D shape and pose estimation have been trained and evaluated on this dataset.



(a) One sample mesh model of 79 classes

(b) Sample annotations projected onto input images

Figure 3: Apollo3D car instance dataset

4 Methodology

The objective of this project was to move towards an unsupervised methodology for vehicular pose estimation and shape reconstruction. To achieve this objective the tasks carried out are broken to 3 stages.

1. Remove dependency on keypoint labels and template shapes which are only present in Apollo3D dataset.
2. Update model architecture to achieve similar performance without these additional annotations.
3. Utilize semi-supervised training to remove the need for exact 3D shape labels.

To achieve each of these targets, the baseline model was modified in a three step process. First the use of keypoints and vehicular templates was removed from the model and training process. Next the architecture was updated to accommodate for the loss in information from the previous stage. Finally, the training process was updated, such that the model can be trained from segmentation labels instead of 3D shapes. While the first step was completed during the length of the project. However, due to lack of time and resources, the latter stages are still work in progress. However, the steps taken to achieve this is explained briefly.

4.1 Baseline model

All prior works have a general structure in which they're made of two components : (1) Feature extraction backbone and (2) detection / reconstruction head. The feature extraction backbone is based on mask R-CNN [25] and extracts individual object features. The detection / reconstruction head uses these extracted information to predict the pose which is made of the rotation and translation components, and the 3D shape through separate components. The major difference between these works lies in the 2D object features which were extracted by the backbone, and the approach taken by the head for shape reconstruction / pose estimation. The relative quantitative performance[1] of these models are given in Table 1[2].

We observe that the BAAM performs better in general compared to other techniques. DeepManta often performs better in the strict criteria (c-s) metric since it uses non-detailed coarse shapes with fewer vertices. However, this

[1]Measured in terms of A3DP metric. Refer Appendix A for more details

[2]Best performing models are bold, and next best is underlined

Table 1: A3DP measure of most recent join shape reconstruction and pose estimation.

Method	A3DP - abs			A3DP - rel		
	mean	c-l	c-s	mean	c-l	c-s
DeepManta [15]	<u>20.10</u>	30.69	**23.76**	16.04	23.76	**19.80**
3D-RCNN [16]	16.44	29.70	19.80	10.79	17.82	11.88
Direct approach [17]	15.15	28.71	17.82	11.49	17.82	11.88
GSNet [18]	18.91	<u>37.42</u>	18.36	<u>20.21</u>	<u>40.50</u>	<u>19.85</u>
BAAM [19]	**23.82**	**46.60**	<u>21.09</u>	**21.16**	**43.68**	18.52

is counter intuitive to our objective since more fine grained shapes would be preferred. Thus BAAM was chosen as the baseline model. The model architecture and loss function of the BAAM model are briefly described below.

Model architecture : The BAAM model, similar to other works, uses a Mask R-CNN as the feature extracting backbone. The backbone extracts 2D features such as bounding boxes, box features and keypoint features. The keypoint features are made as a combination of the location and the visibility. These three 2D features are then reshaped and concatenated to form the object features. In addition, the global features of the image are extracted using another CNN based feature extractor. The global features encode the scene context whereas, each local feature focuses on an object. Then the object features are passed into the three separate modules used for shape reconstruction, rotation estimation, and translation estimation. In addition to the object features, the global features are also passed into the translation estimation module. Once the objects have been reconstructed and projected onto 3D space based on the estimated pose, a 3D non maximum suppression (NMS) technique is used to prune detections. The overall model architecture is shown in Figure 4.

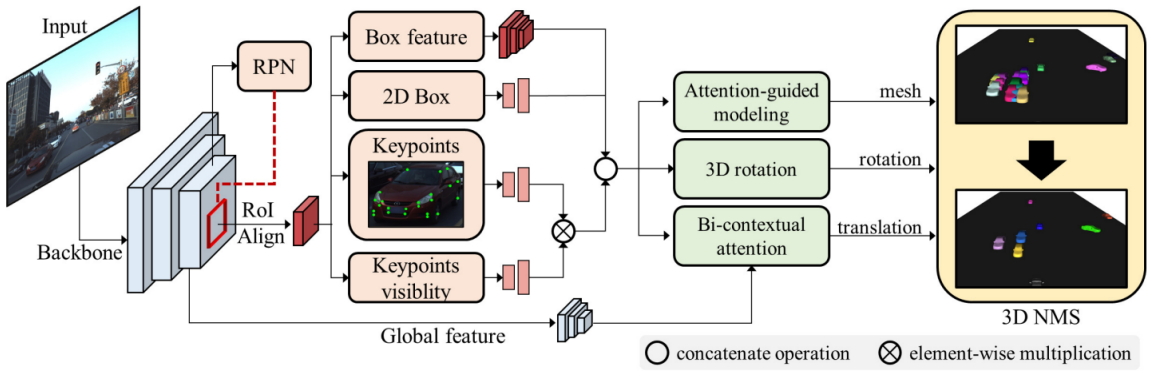


Figure 4: BAAM model architecture

The rotation estimation component consist only of a fully connected network, since this was straightforward. However, the translation estimation and mesh reconstruction is done through attention mechanism. The mesh reconstruction module is referred to as the attention guided modelling (AGM) and the translation estimation component is called Bi-contextual attention (BCA). The global features are only used by the BCA module as seen in the figure.

The AGM in BAAM leverages the relevance between objects and shape priors to estimate object shapes. Shape-aware attention decomposes an object's shape into three components: mean shape (M_s) , template offsets (O_s), and object offsets (O_o). The mean shape and template offsets are calculated as follows:

$$M_s = \frac{1}{79} \sum_{\forall\, s} V_s$$

$$O_s = V_s - M_s \quad \forall\, s \in [1, ..., 79]$$

where V_s is a $3 \times v$ matrix representing the vertex position of the template shapes $s = [1, ..., 79]$ in the dataset. The shape-aware attention estimates the attention scores and object offsets by leveraging the relationship between objects and shape priors. This decomposition simplifies the problem of determining vertex coordinates directly, considering that vehicles are rigid bodies with limited possible object offsets. To measure the relevance between an object and the templates, the template offsets are mapped to template features using a learnable embedding scheme. Object offsets are then predicted using a multi-head cross-attention (MCA) mechanism. The MCA takes queries from object features and keys and values from template features. The object features and template features are learnable embeddings learned from an encoder architecture. The attention score indicates the similarities between objects and shape priors. Then, the final object shape M is represented as follows,

$$M = M_s + A\,O_s + O_o \tag{1}$$

where A is the attention score, and O_o is the object offset calculated as

$$O_o = \text{MLP}(C_A + x)$$

where C_A is the context obtained from the attention mechanism, and x is the object features and MLP is the function of a 2 layer fully connected network which converts the object features to the dimension of vertices. The architecture of the AGM module is shown in Figure 5

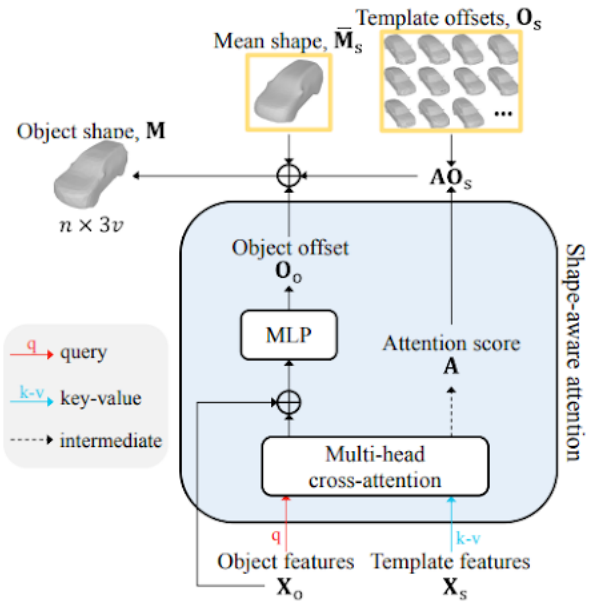


Figure 5: Attention guided modelling (AGM) architecture.

Loss function The loss function was directly adapted from the baseline BAAM model [19]. It consists of three components namely the regression loss, 2D detection loss and the 3D space loss. The regression loss computes the loss at the final detection head. It's the summation of three components namely, the rotation, translation and reconstruction loss. The rotation loss L_{rot} is measured in terms of L1 loss, and the shape reconstruction loss L_{shape} is measured as the L2 loss between prediction and ground truth values. The translation loss L_{tran} is given as a combination of L1 loss, and depth based uncertainty regression loss. Thus, the total regression loss is given as:

$$L_{reg} = L_{rot} + L_{tran} + L_{shape}$$

The 2D keypoint extractor predicts the bounding boxes and keypoints. As such, the 2D detection loss L_{det} is given as the summation of these individual losses as,

$$L_{det} = L_{rpn} + L_{bbox} + L_{kpts}$$

where $L_{rpn}, L_{bbox}, L_{kpts}$ are obtained directly from the implementation of Mask R-CNN [25]. The final component is the 3D space loss L_{3D}. Though the first loss component captures the losses of the three prediction, the losses are calculated independant of each other. However, when projected onto a 3D space, the error can be of a higher degree. To include the effect of this, the prediction and ground truth data are projected onto 3D space, and the difference between the 3D point clouds are calculated using L1 loss. The total final loss is given as a summation of the three losses:

$$L_{tot} = L_{reg} + L_{det} + L_{3D} \tag{2}$$

More information regarding these loss functions can be found in the original BAAM paper [19].

4.2 Removal of keypoints and templates

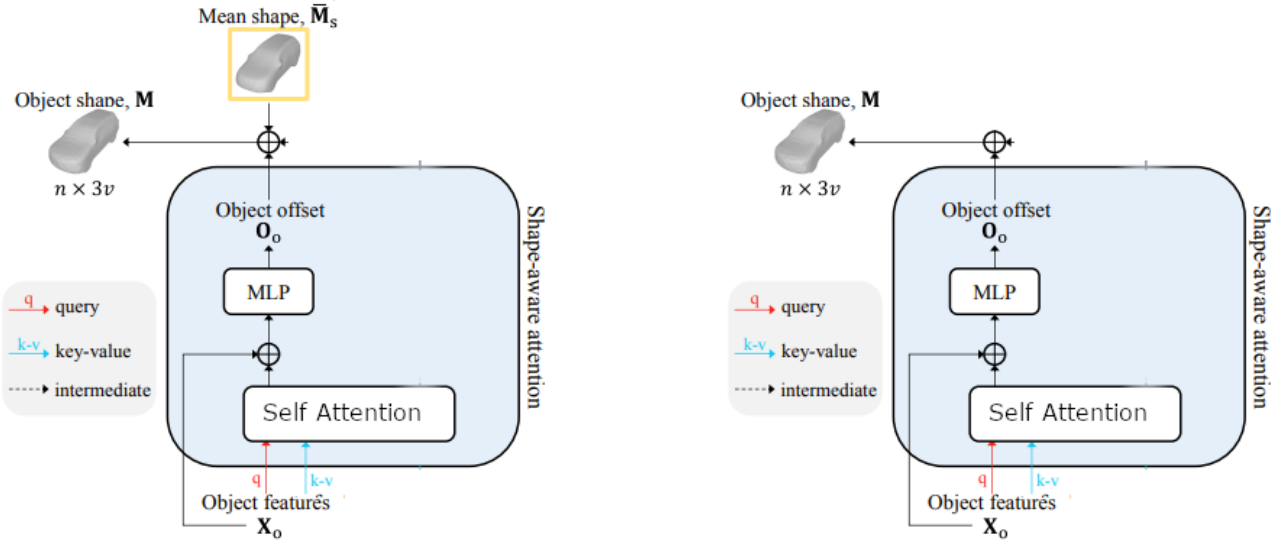
As the first step, the keypoints and templates were removed. To remove the keypoints, the feature extractor was updated such the backbone only predicts the bounding box shape and features. Thereby, the object feature is a concatenation of only the 2D boundinf box information and it's features, but not of the keypoints as presented in prior works [19, 18]. In addition, the component corresponding to keypoints was also removed from the loss function.

Next, To remove the templates from shape prediction head, the attention guided modelling (AGM) component was modified. First, to remove the template features the multi-head cross attention component was replaced by a multi-head self-attention (MSA) mechanism which attends only within the object features. Thus, the MSA takes queries, keys and values from object features only. Now, the equation 1 is modified to obtain the final object shape as,

$$M = M_s + O_o \tag{3}$$

However, the mean shapes are still calculated from the template shapes V_s. To remove this dependency the shape reconstruction equation is finally updated as follows

$$M = O_o \tag{4}$$



(a) Removal of template features

(b) Removal of mean shape

Figure 6: Updated AGM module

The modification used in these two steps are shown in Figure 6. Note that this modification leads to loss of information which leads to the object feature size being halved. As such the dimension of the feature extractors

corresponding to bounding boxes and its features were doubled in size as a followup experiment to visualize the change in performance.

4.3 Updating the model architecture

The removal of keypoints and templates in the previous experiment leads to a loss of performance. To compensate for this loss, the architecture of the shape reconstruction component was modified to a transformer based architecture. Many prior works utilize transformer backbones for shape reconstruction, and have shown similar / superior performance in comparison to CNN based models. Based on prior works, the encoder-decoder architecture of BAAM was replace by the encoder-decoder architecture of the 3D-RETR network [11]. The shape reconstruction pipeline of BAAM consists of a mask R-CNN based encoder which act as the feature extractor backbone, and a attention based decoder which acts as a detection head. These components were replaced by the voxel encoder-decoder architecture in 3D-RETR. However, the CNN decoder of the 3D-RETR model reconstructs voxels as shown in Fig. 7. On the contrary, the Apolloscape dataset contains meshes, and as such the voxel reconstruction head was replaced by mesh reconstruction head, by using 1×1 convolution, 2D convolutions and linear layers.

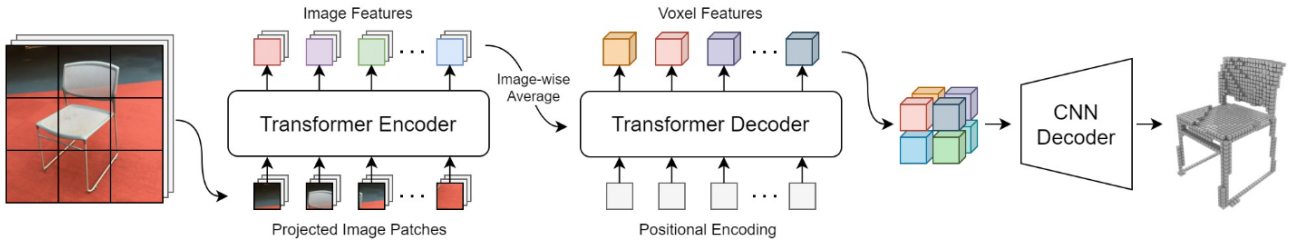


Figure 7: 3D-RETR (Reconstruction transformer) model architecture

Despite these modifications, the entire model cannot be directly trained from scratch since transformer based models need a large amount of data, which the Apollo3D dataset lacks. In addition, it should be noted that that the mask R-CNN backbone was also pre-trained based on the MS COCO 2017 dataset, and as such retraining the transformer encoder-decoder model from scratch would lead to a significant loss in performance. Thus, the model was first pre-trained based on the ShapeNet dataset for voxel reconstruction. Then the transformer encoder-decoder was kept fixed while the reconstruction head was replaced with a mesh reconstructor, and the model was retrained in an end-to-end manner.

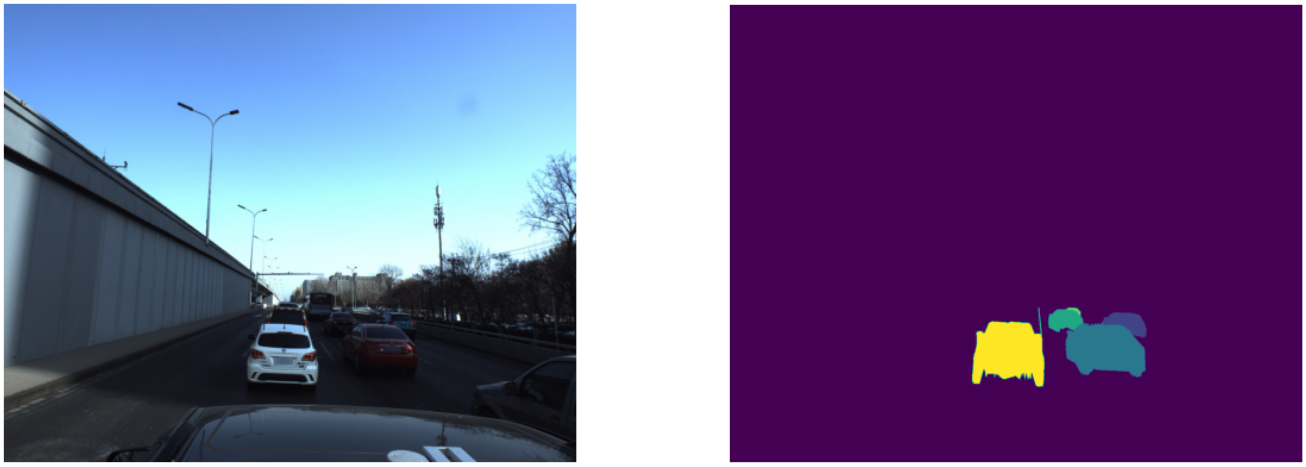
The loss function was kept as it is since the reconstruction head produces similar output. Similarly, the input of the reconstruction head is now the cropped image from bounding box. As such the bounding box, and feature component of the loss function is also kept the same.

4.4 Segmentation label based training

The major shortcomings of many of these works, are the need for 3D shape labels which are not widely present. As such it would be important to move towards a semi-supervised technique where the exact 3D shape labels are not required. This idea has been previously introduced in the context of vehicular shape reconstruction in [16], where the authors propose the use of "render-and-compare" loss which does not require 3D shape labels, but only a set of generic 3D shapes and segmentation labels.

The core idea behind "Render and Compare" is to leverage the consistency between the synthetic renderings and the ground truth data to guide the learning process. Once the pose and shape have been estimated by the model, if the estimated 3D pose or geometry is accurate, the renderings of the 3D object onto the image plane should closely resemble the actual objects in the real images. An example of this is shown in Figure 8, where the predicted 3D shape have been projected onto a 2D plane based on the predicted pose and camera parameters. Therefore, by minimizing the discrepancy between the synthetic and real images, the model can be trained to improve its accuracy in estimating object poses or shapes. The comparison between renderings and ground truth is often done using various image similarity metrics, such as pixel-wise L2 distance, structural similarity index (SSIM), or

perceptual similarity based on features extracted from deep neural networks. In the case of segmentation labels, this has been done using the dice score as the metric. Though this idea has existed before, the complexity of the problem has made it impractical in the past, but the authors have proposed an OpenGL based technique in [16] which leads to regressable end-to-end training mechanism.



(a) Input image

(b) Predicted models rendered onto the image plane

Figure 8: Segmentation prediction for "Render and compare" loss

Through this methodology, the training can be made independent of 3D shape labels, and the training would made be possible through other datasets such as KITTI. In addition, the model can be first pretrained using the Apollo3D dataset, and then be retrained on KITT after removing all the additional annotation related components which are absent in the latter dataset. However, this architectural change is quite significant and the effect should be studied separately.

5 Experimental setup

The modifications proposed in section 4.2 were conducted as separate experiments as follows.

1. **Baseline model** : First the baseline model was trained as it is without any modifications.
2. **Remove keypoints** : Next the keypoints were removed from the model, and the corresponding loss function component was updated.
3. **Remove template features** : Template offsets which are calculated based on template features are then removed and the MCA is replaced with SCA. The shape is thereby reconstructed according to equation (3).
4. **Remove mean shape** : Finally, the mean shapes are also removed, and the shape is reconstructed according to equation (4).

Similar to the BAAM experiments, the Apollo3D car instance dataset was used here for training and evaluation. The dataset contains 4036, 200, and 1041 high resolution images for training, validation, and testing respectively. The input images were resized to a shape of (1355,1692) as the first step. The models were then trained with loss functions specified in 4.2 and the adam optimizer with a learning rate of $1e-3$ for 10 epochs. All the models were trained on a server with an V100 (16GB) GPU, and the total training time is approximately 9 hours with this setup.

The evaluation was performed both quantitatively and qualitatively. Qualitative evaluation was performed by observing the reconstructed shapes in their corresponding orientation from the point-of-view of the camera. This output is compared to the input image to visualize and obtain an estimative performance of the model. However,

the output consists of three components (reconstruction, rotation, translation) and as such the visualization doesn't provide additional details regarding the actual performance of the model. As such, in addition quantitative evaluation was performed by considering **A3DP** metric on the test dataset. The A3DP[3] metric is originally defined in [17], and jointly measures 3D translation, rotation, and shape reconstruction accuracy. The absolute translation error and a relative one are given as A3DP-abs and A3DP-rel, according to the same convention used.

The experiments related to the modifications presented in sections 4.3 and 4.4 were still in progress at the end of the project timeline and as such the results are not presented in this report.

6 Results and discussion

To obtain the baseline performance of the model, it was retrained without any modifications. The corresponding input and output images are show in Figure 9. The quantitative metrics obtained were similar to that in the the BAAM paper.

(a) Input image

(b) Predicted object shapes projected to image plane

Figure 9: Sample input and output from baseline model

6.1 Remove keypoints

As the first experiment, the keypoints were removed from the model and loss function (w/o keypoints). After removing these information, the dimension of the extracted object features was doubled (w/o keypoints + x2), and the model was retrained again to study the change in performance. The quantitative metrics from these experiments are given in Table 2.

Table 2: A3DP measure after removing keypoints from BAAM

Method	A3DP - abs			A3DP - rel		
	mean	c-l	c-s	mean	c-l	c-s
baseline	23.82	46.60	21.09	21.16	43.68	18.52
w/o keypoints	20.97	42.36	18.93	15.50	35.22	11.98
w/o keypoints + x2	21.19	41.76	18.43	16.03	34.83	12.03

From these results, we observed that as soon as the keypoints were removed, the model experienced a drop in performance of around -3.0 for A3DP-abs value and -6.0 for A3DP-relative value. Then when the dimension of object features was increased there was a slight increase in the performance of around +0.2, but this is negligible.

[3]Refer Appendix A for more information

6.2 Remove template features

Next, the template features were removed from the AGM module (w/o templates). Then the keypoints were also removed (w/o templates, keypoints) and finally, the dimension of the extracted object features was doubled (w/o templates, keypoints + x2), and the model was retrained again to study the change in performance. The quantitative metrics from these experiments are given in Table 3.

Table 3: A3DP measure after removing keypoints and template features from BAAM

Method	A3DP - abs			A3DP - rel		
	mean	c-l	c-s	mean	c-l	c-s
baseline	23.82	46.60	21.09	21.16	43.68	18.52
w/o templates	21.67	41.44	20.06	17.20	36.61	13.63
w/o templates, keypoints	17.21	39.68	13.36	11.78	29.57	7.87
w/o templates, keypoints + x2	19.27	41.25	17.20	12.77	31.34	9.41

From these results, we observe that as soon as the template features are removed, the model experienced a drop in performance of around -2.0 for A3DP-abs value and -4.0 for A3DP-relative value. Then when the keypoints are removed the performance drops further (-2.0 and -4.0 respectively). However, in this case when the dimension of the object features are increased, there is a slight increase in performance of around +2.0 which is non-negligible.

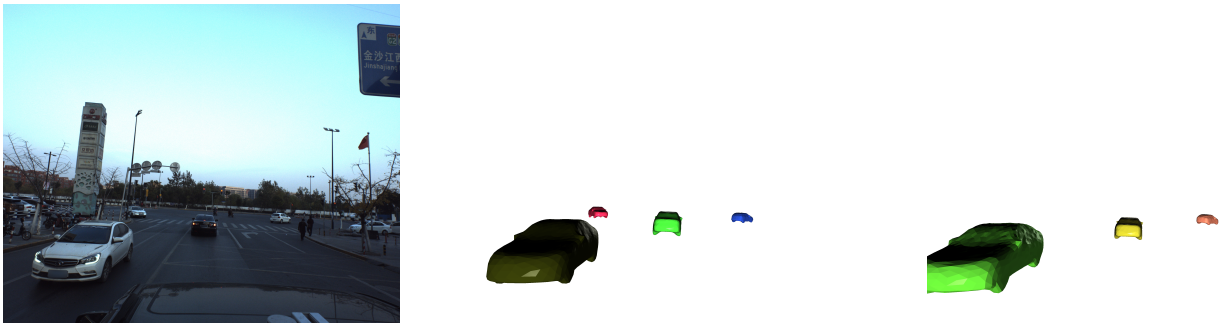
6.3 Remove mean shape

Finally, the mean shape was also removed from the the AGM module (w/o template, mean shape). Then similar to the last experiment the keypoints were also removed (w/o template, mean shape, keypoint) and finally, the dimension of the extracted object features was doubled (w/o template, mean shape, keypoint + x2), and the model was retrained again to study the change in performance. The quantitative metrics from these experiments are given in Table 4.

Table 4: A3DP measure after removing keypoints, templates and mean shape from BAAM

Method	A3DP - abs			A3DP - rel		
	mean	c-l	c-s	mean	c-l	c-s
baseline	23.82	46.60	21.09	21.16	43.68	18.52
w/o mean shape, template	22.12	44.15	20.32	20.98	43.66	18.22
w/o template, mean shape, keypoint	18.47	40.95	14.37	12.76	31.14	8.35
w/o template, mean shape, keypoint + x2	<u>19.21</u>	<u>41.18</u>	<u>15.05</u>	15.54	34.43	10.57

From these final results, we observe that as soon as the mean shape, and template features are removed, the model experienced slight drop in performance. Then when the keypoints are removed, the performance drop was significant (approximately -3.0 for A3DP-abs value and -6.0 for A3DP-relative value). Then, when the dimension of the object features are increased, there is a slight increase in performance of around +1.0. This final model which we obtain after removing the keypoints and template shape information still provides similar performance in comparison to the next best performing model in 1 which uses both these details during training. As such it is evident that the attention mechanism significantly improves the performance despite the lack of keypoint and template information. Furthermore, the qualitative results are shown in Figure 10. We can observe that one of the cars are missing in the final proposed model, and in addition, the position of the car (translation) is also inaccurate in comparison to the baseline model. Thus, the performance drop could be due to many reasons such as fewer vehicular predictions, lower translation accuracy, incorrect shape reconstruction, etc., and the effect of these modifications should be studies in details with respect to individual metrics in future studies.



(a) Input image (b) Predictions from baseline model (c) Predictions from final model

Figure 10: Qualitative comparison of baseline model and final model performance

7 Conclusion

3D shape reconstruction and pose estimation is a vital problem and helps us with many tasks related to vehicular perception. The SOTA model uses datasets which contain 3D shape labels and other annotiaition informtaion such as 2D keypoints which are not widely available for other datasets. This project presented a few modifications of overcoming these shortcomings along with the variation of the performance of the model due to these modifications. Although the final model has suffered a significant drop in performance in comparison to the baseline, it still performs well in comparison to other prior works. Furthermore, it would be an interesting study to experiment on the the proposed modifications in relation to the change in model architecture and training mechanism, which would completely remove the dataset specific training process and enable the model to be trained on different vehicular perception dataset.

In addition to the proposed techniques, moving towards a completely unsupervised technique would be important for robustness of the model, and this should be studied in detail. Furthermore, cooperative perception techniques have gained interest in recent years, and as such it would be worthwhile to study the performance of these proposed model on images taken at different point of views, especially through infrastructure cameras. Finally, by incorporating stero images, the scale of the detected objects could be recovered, and investigating the possibility of incorporating this information would be beneficial for many applications.

References

[1] Yu Hong, Hang Dai, and Yong Ding. "Cross-Modality Knowledge Distillation Network for Monocular 3D Object Detection". In: *ECCV*. Lecture Notes in Computer Science. Springer, 2022.

[2] Chien-Yi Wang et al. "Patchnet: A simple face anti-spoofing framework via fine-grained patch recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 20281–20290.

[3] Li Wang et al. "Depth-conditioned Dynamic Message Propagation for Monocular 3D Object Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 454–463.

[4] Dennis Park et al. "Is Pseudo-Lidar needed for Monocular 3D Object detection?" In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[5] Arsalan Mousavian et al. "3D Bounding Box Estimation Using Deep Learning and Geometry". In: *CoRR* abs/1612.00496 (2016). arXiv: 1612.00496. URL: http://arxiv.org/abs/1612.00496.

[6] Yan Lu et al. "Geometry Uncertainty Projection Network for Monocular 3D Object Detection". In: *arXiv preprint arXiv:2107.13774* (2021).

[7] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. "Objects are different: Flexible monocular 3d object detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 3289–3298.

[8] Di Wu et al. "6D-VNet: End-To-End 6DoF Vehicle Pose Estimation From Monocular RGB Images". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, pp. 1238–1247. DOI: 10.1109/CVPRW.2019.00163.

[9] Yan Xu et al. "Rnnpose: Recurrent 6-dof object pose refinement with robust correspondence field estimation and pose optimization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 14880–14890.

[10] Yannick Bukschat and Marcus Vetter. "EfficientPose: An efficient, accurate and scalable end-to-end 6D multi object pose estimation approach". In: *arXiv preprint arXiv:2011.04307* (2020).

[11] Zai Shi et al. "3D-RETR: End-to-End Single and Multi-View3D Reconstruction with Transformers". In: *BMVC*. 2021.

[12] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. "Mesh r-cnn". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 9785–9795.

[13] Nanyang Wang et al. "Pixel2mesh: Generating 3d mesh models from single rgb images". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 52–67.

[14] Angjoo Kanazawa et al. "Learning category-specific mesh reconstruction from image collections". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 371–386.

[15] Florian Chabot et al. "Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2040–2049.

[16] Abhijit Kundu, Yin Li, and James M Rehg. "3d-rcnn: Instance-level 3d object reconstruction via render-and-compare". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3559–3568.

[17] Xibin Song et al. "Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5452–5462.

[18] Beibei Wang et al. "GSNet: learning spatial-temporal correlations from geographical and semantic aspects for traffic accident risk forecasting". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 5. 2021, pp. 4402–4409.

[19] Hyo-Jun Lee et al. “BAAM: Monocular 3D pose and shape reconstruction with bi-contextual attention module and attention-guided modeling”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 9011–9020.

[20] Andreas Geiger et al. “Vision meets robotics: The kitti dataset”. In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.

[21] Holger Caesar et al. “nuscenes: A multimodal dataset for autonomous driving”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11621–11631.

[22] Pei Sun et al. “Scalability in perception for autonomous driving: Waymo open dataset”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 2446–2454.

[23] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. “Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild”. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2014.

[24] Angel X Chang et al. “Shapenet: An information-rich 3d model repository”. In: *arXiv preprint arXiv:1512.03012* (2015).

[25] Kaiming He et al. “Mask R-CNN”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

A. Appendix - A3DP evaluation metric

Similar to the detection task, the average precision (AP) is usually used for evaluating 3D object understanding. However, the similarity is measured using 3D bounding box IoU with orientation (average orientation similarity (AOS)) or 2D bounding box with viewpoint (average viewpoint precision (AVP) [23]). Unfortunately, those metrics can only measure very coarse 3D properties, yet object shape has not been considered jointly with 3D rotation and translation.

Mesh distance and voxel IoU are usually used to evaluate 3D shape reconstruction. In our case, a car model is mostly compact, thus we consider comparing projection masks of two models following the idea of visual hull representation. Specifically, we sample 100 orientations at yaw angular direction and project each view of the model to an image with a resolution of 1280×1280. We use the mean IoU over all views as the car shape similarity metric. For evaluating rotation and translation, we follow the metrics commonly used for camera pose estimation. In summary, the criteria for judging a true positive given a set