# Week 1 : Literature survey

## Tasks :

- [x] ~~Look for autonomous driving datasets containing 3d models, and images~~
- [x] ~~Literature survey on 3d reconstruction~~
- [x] ~~Find a baseline for monocular / stereo image based 3d reconstruction~~
- [ ] Start tests with baseline model

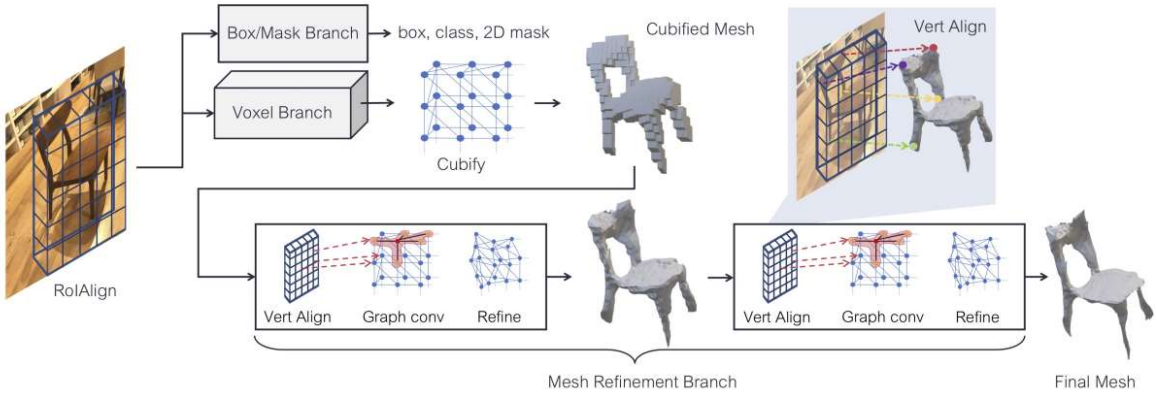
# Summary

The first week consisted mostly of literature surveys. The objective of the practical is to estimate pose and 3d shape from stereo images. As such the prior works were divided as follows:

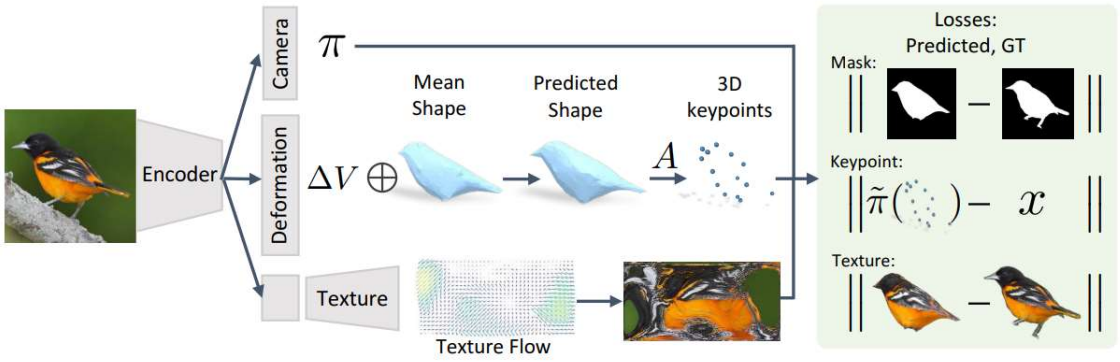
- 3D shape reconstruction : Shape reconstruction based on single images (without pose)
- Pose estimation with fixed shape : Estimate the pose (rotation and translation) of multiple objects when a set of shapes are given.
- 3D shape and pose estimation : Estimate the pose and 3D shape simultaneously in a multi object environment.

**<u>3D shape reconstruction</u>**

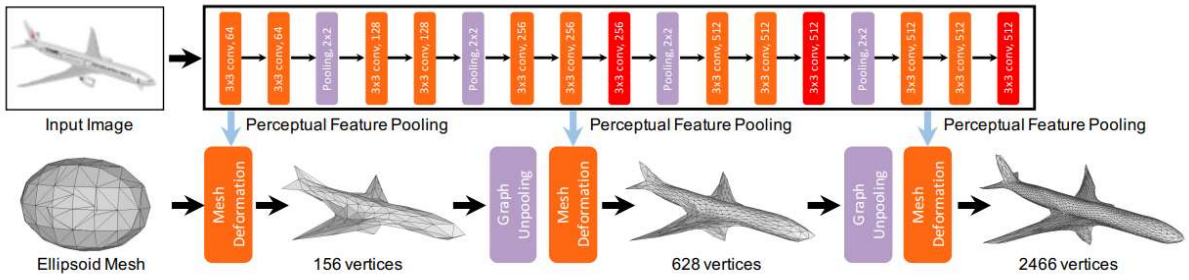
- **Mesh R-CNN**
  - Dataset : Pix3D, ShapeNet
  - Method : Mask R-CNN based backbone -> Voxel prediction -> Mesh optimization
  - Links : https://gkioxari.github.io/meshrcnn/




- **Learning Category-Specific Mesh Reconstruction from Image Collections**
  - Dataset :
  - Method : Deformable mesh model
    - Learn pose, shape, and texture by projecting 3D reconstruction to image
    - Shape expressed as mean class shape + <u>deformation</u>
  - Notes :

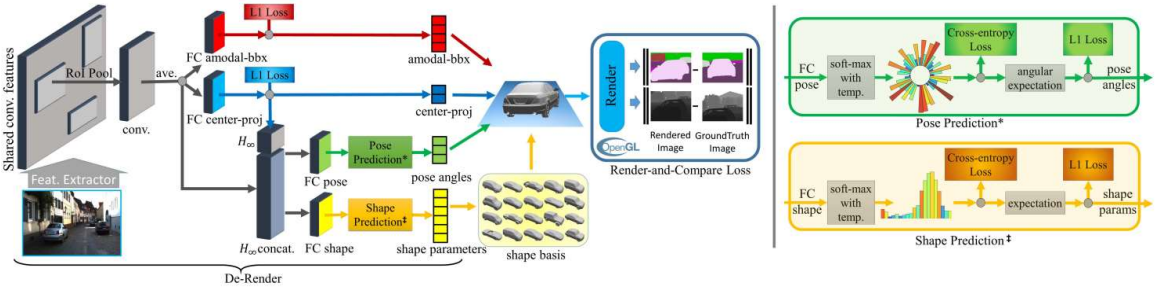



- **Pixel2Mesh**: Generating 3D Mesh Models from Single RGB Images
  - Dataset : ShapeNet
  - Method :2D feature extractor + Deformable mesh learned from GCNI
    - 2D CNN that extract perceptual feature from the input image
    - GNN based mesh deformation network to progressively deform an ellipsoid mesh into the desired 3D model
  - Key contribution : R-GNN
  - Notes :



## Pose and shape estimation

- **3D-RCNN**: Instance-level 3D Object Reconstruction via Render-and-Compare (2018)
  - Dataset : KITTI, PASCAL3D+
  - Method : Shape and orientation prediction
    - Shape prediction based on shape basis + render and compare loss
  - Links : https://openaccess.thecvf.com/content_cvpr_2018/papers/Kundu_3D-RCNN_Instance-Level_3D_CVPR_2018_paper.pdf




- **GSNet**: Joint Vehicle Pose and Shape Reconstruction with Geometrical and Scene-aware Supervision (2020)
  - Dataset : Apolloscape, PASCAL3D+
  - Method :
  - Link : https://arxiv.org/pdf/2007.13124v1.pdf

- **BAAM**: Monocular 3D pose and shape reconstruction with bi-contextual attention module and attention-guided modeling
  - Dataset : Apolloscape
  - Method :
  - Link : https://openaccess.thecvf.com/content/CVPR2023/papers/Lee_BAAM_Monocular_3D_Pose_and_Shape_Reconstruction_With_Bi-Contextual_Attention_CVPR_2023_paper.pdf

- Holistic 3D Scene Understanding from a Single Image with Implicit Representation
  - Dataset : Pix3D
  - Method : TODO

- Learning Monocular 3D Vehicle Detection without 3D Bounding Box Labels
  - Dataset : KITTI
  - Method :TODO

**Pose estimation with fixed shapes**

- **6D-VNet**: End-to-end 6DoF Vehicle Pose Estimation from Monocular RGB Images
  - Dataset : Apolloscape + other
  - Method : TODO
  - Link : https://github.com/stevenwudi/Kaggle_PKU_Baidu https://openaccess.thecvf.com/content_CVPRW_2019/papers/Autonomous%20Driving/Wu_6D-VNet_End-to-End_6-DoF_Vehicle_Pose_Estimation_From_Monocular_RGB_Images_CVPRW_2019_paper.pdf

- **Disp R-CNN:** Stereo 3D Object Detection via Shape Prior Guided Instance Disparity Estimation
  - Dataset : KITTI

- **RNNPose**: Recurrent 6-DoF Object Pose Refinement with Robust Correspondence Field Estimation and Pose Optimization
  - Dataset : LineMOD

- **EfficientPose**: An efficient, accurate and scalable end-to-end 6D multi object pose estimation approach

**Dataset**

- **ApolloCar3D**: A Large 3D Car Instance Understanding Benchmark for Autonomous Driving
  - Dataset for multiple tasks
    - 3D object detection from monocular
  - Includes 3D models (mesh)
  - Links :
    - https://apolloscape.auto/car_instance.html
    - https://github.com/ApolloScapeAuto/dataset-api/tree/master/car_instance
- **PASCAL3D+** : Beyond PASCAL: A benchmark for 3D object detection in the wild
  - CAD models of multiple classes. Classes include cars, bikes, etc.
  - Links :
    - https://cvgl.stanford.edu/papers/xiang_wacv14.pdf
    -
- **KITTI**
  -
  - Notes : Does not contain 3d models

# Week 2 : Data exploration

## Tasks :

- [x] ~~Read most recent works on 3D reconstruction + pose estimation~~
  - [x] ~~GSNet~~
  - [ ] BAAM
- [x] ~~Find shortcomings in existing work - 3D~~
  - [ ] BAAM
  - [x] ~~GSNet~~
  - [ ] 3D - RCNN
- [x] ~~Find possible improvements on existing works / other experiments~~
  - [ ] Transformers head for 3D reconstruction
    - 3D-RETR: End-to-End Single and Multi-View 3D Reconstruction with Transformers (BMVC 2021)
  - [ ] Improve efficiency
    - How to change the backbone to something faster than mask RCNN which obtains similar information.
  - [ ] Domain adaptation for infrastructure camera data
    - Use Rope 3D,  A9-collaborative or DAIR-V2X dataset
  - [ ] Incorporate stereo camera to extract scale
- [x] ~~Visualise data~~
  - [x] ~~Apollo3D~~
  - [ ] Pascal3D +
- [ ] Start tests with baseline model

# Summary :

Most of this week was spent on obtaining the data / analysing it.

## Datasets

In the context of shape and pose estimation in autonomous driving, there are quite a few datasets which offer diverse types of data for shape estimation. However, this project mainly focuses on 3D shape reconstruction from stereo images, as such the following datasets would be useful in this scenario.

**Apollo 3D**

The dataset contains 60,000 labeled 3D car instances from 5,277 real-world images, based on industry-grade CAD car models. Each image is labelled with the type of object in the image together with the absolute 6D pose of vehicles (yaw, pitch, roll, x, y, z). The dataset contains images taken from 2 such cameras. Many of the prior works on 3D shape and pose estimation are based on this model including BAAM which is the SOTA in this field.

**Pascal3D+**

The dataset is an extension of the Pascal VOC dataset to address the shortcomings in CV research 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. These objects are annotated with pose information (azimuth, elevation and distance to camera). Certain prior works have used the cars category together with other accompanying datasets such as KITTI for training and evaluation.

**KITTI**

The KITTI dataset is an autonomous driving dataset which is useful for computer vision tasks such as 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 annotations, previous works have used this in combination with other datasets such as Pascal3D+ for training and evaluation.

## Data Analysis

The focus was mainly on the Apollo3D dataset since it meets many of the requirements for this project

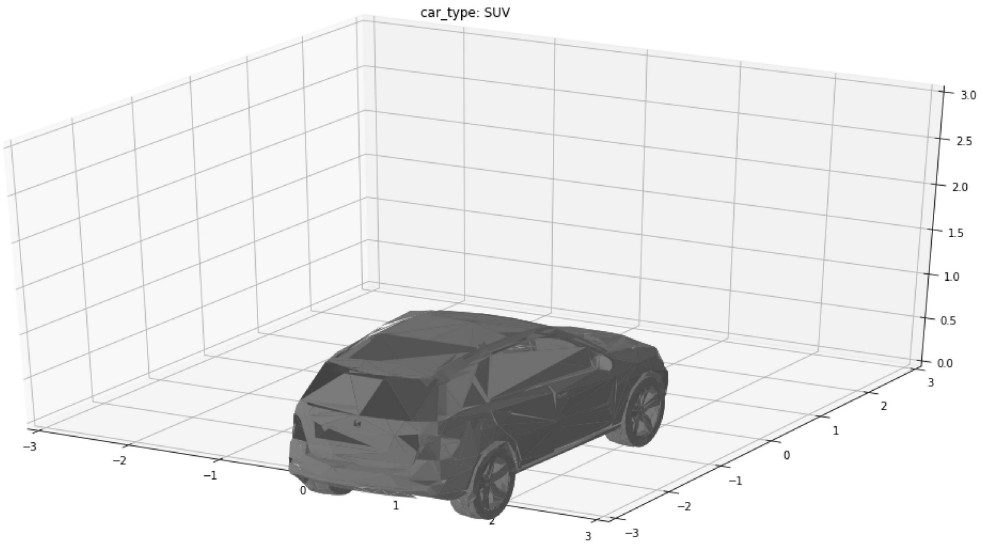
- Stereo images annotated with pose information
- 3D mesh models

The original dataset can be obtained from the following link:
https://apolloscape.auto/car_instance.html

Analysis of the dataset posed certain challenges

- .pkl file issue : The models available on the original dataset contains pkl files which were not readable, as such thus the kaggle dataset was used instead: https://www.kaggle.com/competitions/pku-autonomous-driving
- version mismatch : The code was based on python 2.7 on Ubuntu 14.04. To make it compatible with newer models minor modifications were made. The dockerized file based on this can be found here.

The dataset contains 79 car models separated into 3 classes. One such model is shown below.




A sample image from the dataset is shown below, together with the corresponding 3D models, the models rendered onto the image and the depth map

# Week 3 : Baseline results

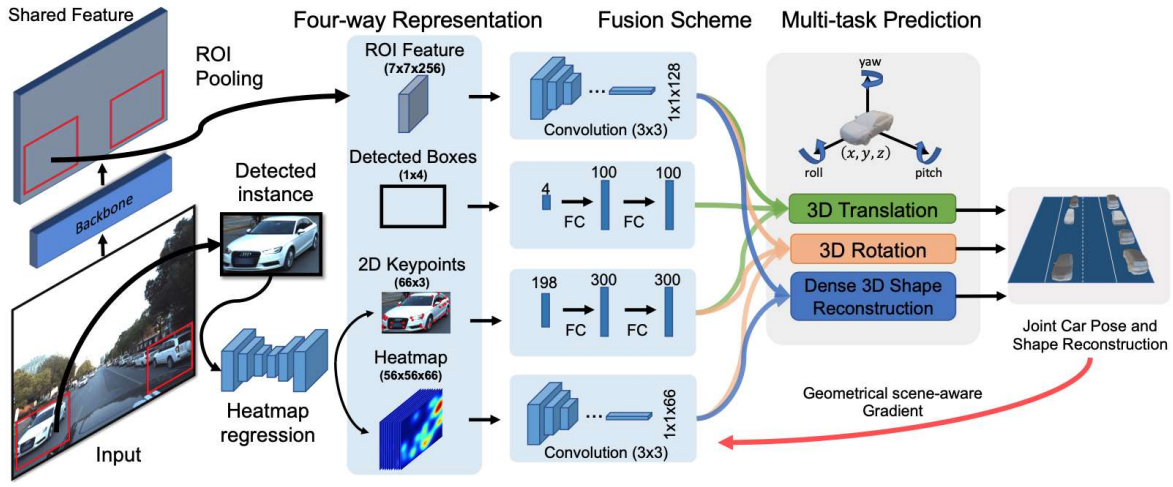
## Tasks :

- [x] ~~Start with GSNet~~
  - [x] ~~Read the paper~~
  - [x] ~~Go through codebase to make changes~~
- [x] ~~Get a baseline model working~~
- [ ] Check unsupervised learning
- [ ] Check implementation of 3D-RCNN

# Summary :

Initial focus was on GSNet paper and architecture. Notes on the paper:

- Mask R-CNN based feature extractor with ResNet101 backbone for RIO feature extraction and object detection




The code was analysed to update the backbone and head. There were certain issues with the codebase

- Undocumented code
- To training code
- Pretrained models do not exist
- Issues installing required libraries

# Week 4 : Analyse code for improvement

## Tasks :

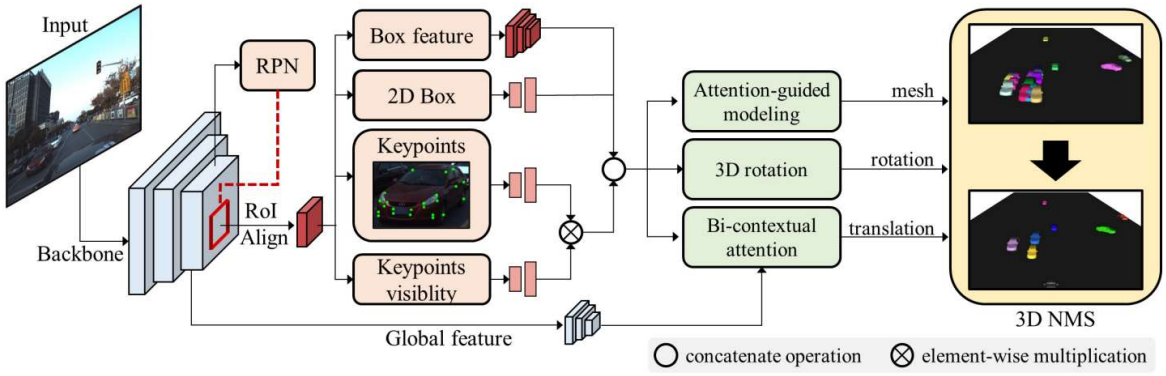
- [ ] Qualitative analysis
  - [ ] Test on KITT / waymo
  - [x] ~~Visualise rendering~~
- [ ] Quantitative evaluation
  - [x] ~~Semantic segmentation measure vs visual rendering~~
- [ ] Measure FPS
  - Started on measure (How to set a baseline device?)
- [ ] BAAM
  - [x] ~~Read the paper and check for improvements~~
    - Faster backbone than mask RCNN
  - [ ] Check code for optimization
- [ ] Check unsupervised learning
- [ ] Check implementation of 3D-RCNN

# Summary:

The objective of this week was to understand the BAAM model and to make changes to it. For this purpose, the paper was first read then the code was analysed to understand the components of the model. In addition to this, it was planned that certain measures would be introduced to evaluate the performance of the model both qualitatively and quantitatively.

## BAAM

The BAAM model, similar to GSNet, extends Mask R-CNN. 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 features together with global features are then used as the input features for 3D pose and shape estimation. The global features encode the scene context whereas, each local feature focuses on an object.




**3D shape estimation through AGM**

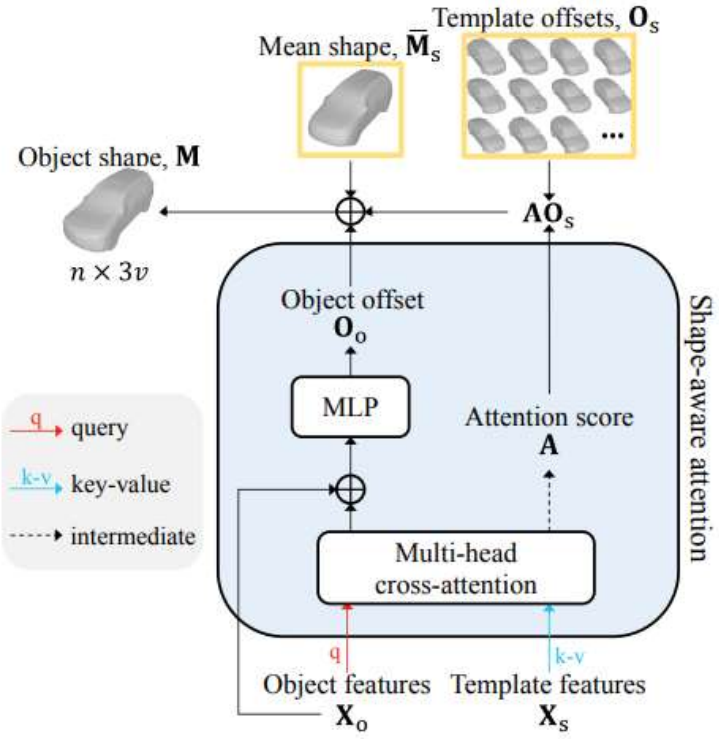
The attention-guided modelling (AGM) in BAAM leverages the relevance between objects and shape priors to estimate object shapes.
Shape-aware attention decomposes an object's shape into three components: mean shape, template offsets, and object offsets. Then, the object shapes are represented as a combination of the mean shape (Ms), attention scores (A), and object offsets (Oo).

$$M = M_s + AO_s + O_o$$

The mean shapes and template offsets are obtained from the 79 object models in the dataset. The shape-aware attention estimates the attention scores (A) and object offsets (Oo) 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, similar to the multi-head self-attention (MSA) approach. 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.




The next step would involve replacing the encoder + MCA block with a transformer architecture to estimate the 3D shape.

**3D pose estimation through BCA**

In addition to the shape, the pose is estimated in terms of rotation and translation. The rotation is directly obtained from the object features through a FCN. On the other hand the translation component is estimated through a Bi-contextual attention (BCA) component which uses global features in addition to object features. However, these 2 components are not analysed in detail as the focus is mainly kept on the reconstruction component.

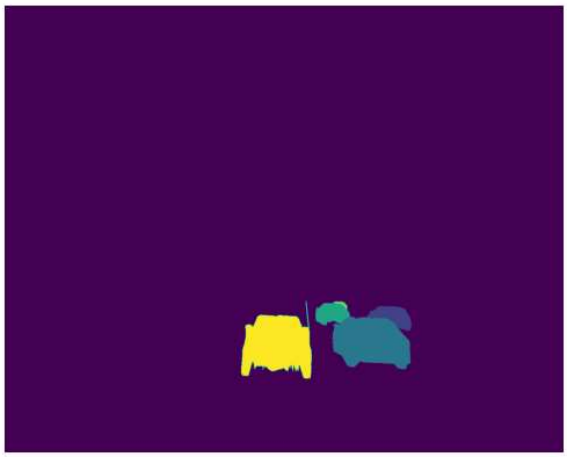
## Evaluation metrics

The performance of the model has to be measured in terms of both accuracy as well as efficiency. As the overall performance metric, average 3D precision (A3DP), which jointly measures 3D translation, rotation, and shape reconstruction accuracy, is used in BAAM. However, in addition to this the 3D IOU, rotation error and translation error can be considered individually.

In addition to this, the following metrics are could be introduced for other datasets where 3D labels are not available

**Accuracy**

The accuracy can also be measured in terms of the 2D mAP based on the segmentation maps. This is done by rendering the object onto the image and then measuring AP of the 2D segmentation map. The figure below shows the input image and the corresponding segmentation map obtained through rendering.



The accuracy can be qualitatively measured by visualising the overlap between the rendering and the image. It can also be quantitatively measured as the average precision of the prediction when compared with the ground truth instance segmentation labels.

Note that while the Apollo3D dataset does not contain segmentation labels, other datasets such as KITTI contain this information, and in the absence of 3D mesh annotations, this metric can be used as an alternative evaluation metric for the reconstruction accuracy.

**Efficiency**

In general, the efficiency of the model is measured in terms of FPS. This is an important metric, specially in the context of real time object detection which is vital autonomous driving tasks. The FPS is measured as the number of inferences that can be made by the model per second on a standard device.