Neural Semantic 3D World Modeling and Mapping summary

S. Papadopoulos, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@aiia.csd.auth.gr
www.aiia.csd.auth.gr
Version 2.2

Artificial Intelligence & Information Analysis Lab

Neural Semantic 3D World Modeling

- **Introduction**
- Disparity/Depth Estimation with NNs
- Joint 3D Scene Geometry and Semantics Estimation
- Semantic 3D World Maps
- Semantic 3D World Map Annotations

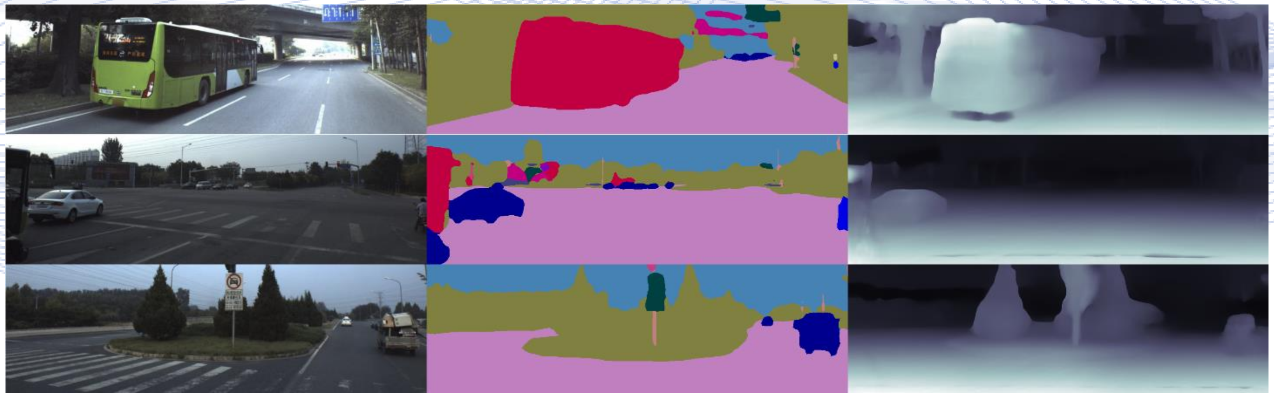
Introduction

- Autonomous/robotic systems (e.g., autonomous cars, drones, etc.) are characterized by their ability to navigate an area on their own, by exploiting sensor data acquired on-the-fly and AI algorithms.

- Knowing the geometry and semantics of a depicted scene/object is a prerequisite for understanding its surroundings and thus, safely navigate therein.

Introduction

- Traditionally, scene geometry was directly sampled using 3D sensors, such as LiDARs.
- Estimation of the underlying scene semantics was limited to object detection with handcrafted features.
- Recently, Deep Neural Networks (DNNs) enabled accurate scene geometry and semantics estimation, using visual sensors only, such as RGB or RGB-D cameras.



Classification/Recognition/ Identification

- Given a set of classes $\mathcal{C} = \{\mathcal{C}_i,\ i = 1, \dots, m\}$ and a sample $\mathbf{x} \in \mathbb{R}^n$, the ML model $\hat{\mathbf{y}} = \boldsymbol{f}(\mathbf{x}; \boldsymbol{\theta})$ predicts a class label vector $\hat{\mathbf{y}} \in [0, 1]^m$ for input sample $\mathbf{x}$, where $\boldsymbol{\theta}$ are the learnable model parameters.

- Essentially, a probabilistic distribution $P(\hat{\mathbf{y}}; \mathbf{x})$ is computed.
- Interpretation: likelihood of the given sample $\mathbf{x}$ belonging to each class $\mathcal{C}_i$.

- Single-target classification:
  - Classes $\mathcal{C}_i, i = 1, \dots, m$ are ***mutually exclusive***: $||\hat{\mathbf{y}}||_1 = 1$.
- Multi-target classification:
  - Classes $\mathcal{C}_i, i = 1, \dots, m$ are ***not mutually exclusive***: $||\hat{\mathbf{y}}||_1 \geq 1$.

Supervised Learning

- A sufficient large training sample set $\mathcal{D}$ is required for Supervised Learning (regression, classification):

$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \dots, N\}.$$

- $\mathbf{x}_i \in \mathbb{R}^n$: n –dimensional input (feature) vector of the i-th training sample.
- $\mathbf{y}_i$: its target label (output).
- Target form $\mathbf{y}$ can vary:
  - it can be categorical, a real number or a combination of both.

Classification/Recognition/ Identification

- ***Training***: Given N pairs of training samples $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \ldots N\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in [0,1]^m$, estimate $\boldsymbol{\theta}$ by minimizing a loss function: $\min_{\boldsymbol{\theta}} J(\mathbf{y}, \hat{\mathbf{y}})$.

- ***Inference/testing***: Given N_t pairs of testing examples $\mathcal{D}_t = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \ldots, N_t\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in [0,1]^m$, compute (***predict***) $\hat{\mathbf{y}}_i$ and calculate a performance metric, e.g., classification accuracy.

Regression

Given a sample $\mathbf{x} \in \mathbb{R}^n$ and a function $\mathbf{y} = \boldsymbol{f}(\mathbf{x})$, the model predicts ***real-valued quantities*** for that sample: $\hat{\mathbf{y}} = \boldsymbol{f}(\mathbf{x}; \boldsymbol{\theta})$, where $\hat{\mathbf{y}} \in \mathbb{R}^m$ and $\boldsymbol{\theta}$ are the learnable parameters of the model.

- **Training**: Given N pairs of training examples $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \ldots N\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in \mathbb{R}^m$, estimate $\boldsymbol{\theta}$ by minimizing a loss function: $\min_{\boldsymbol{\theta}} J(\mathbf{y}, \hat{\mathbf{y}})$.

- **Testing**: Given N_t pairs of testing examples $\mathcal{D}_t = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \ldots, N_t\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in \mathbf{y}_i \in \mathbb{R}^m$, compute (predict) $\hat{\mathbf{y}}_i$ and calculate a performance metric, e.g., MSE.

Semantic Image Segmentation

- CNN Semantic image segmentation typically uses a cascade of an ***encoding*** and a ***decoding subnetwork***.
- The final output of the decoder is a ***semantic image map***, having:
  - ***same spatial resolution*** as the input and
  - as ***many channels*** as the object class number.
- ***Per-pixel*** image classification is performed.

Disparity/Depth Estimation with NNs

Disparity/Depth estimation using Neural Networks (NN) can be divided into four categories:

- NNs for stereo image pair patch matching.
- NN computation of the dense disparity (or depth) map directly from stereo image pairs (without any explicit feature matching).
- Monocular supervised disparity/depth estimation.
- Unsupervised NN disparity/depth estimation methods.

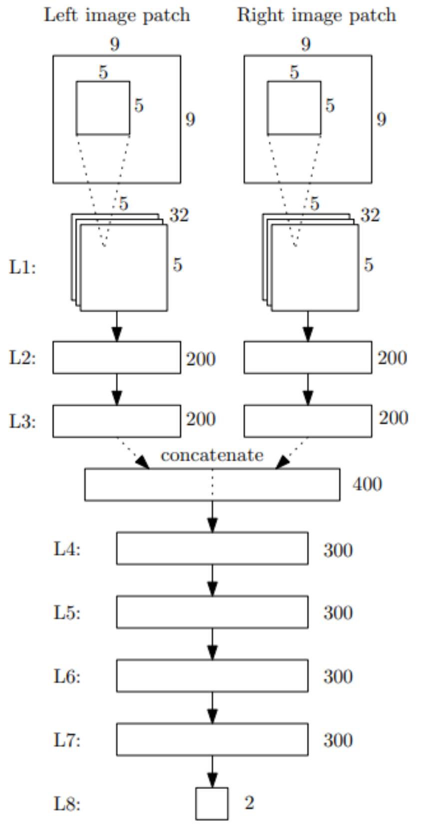
Disparity/Depth map Estimation with NNs

CNN architecture for ***patch comparison*** [ZBO2015]:

- CNN is trained to predict how well two image patches match and use it to compute the stereo matching cost:

$$SAD(\mathbf{p},\mathbf{d}) = \sum_{\mathbf{q}\in\mathcal{N}_\mathbf{p}} |f_l(\mathbf{q}) - f_r(\mathbf{q}-\mathbf{d})|.$$

- $f_l(\mathbf{p}), f_r(\mathbf{p})$: image intensities at position $\mathbf{p}$ in the left and right image.
- $\mathcal{N}_\mathbf{p}$: image neighborhood at pixel $\mathbf{p}$.
- $\mathbf{d} = [d, 0]^T$: stereo disparity.

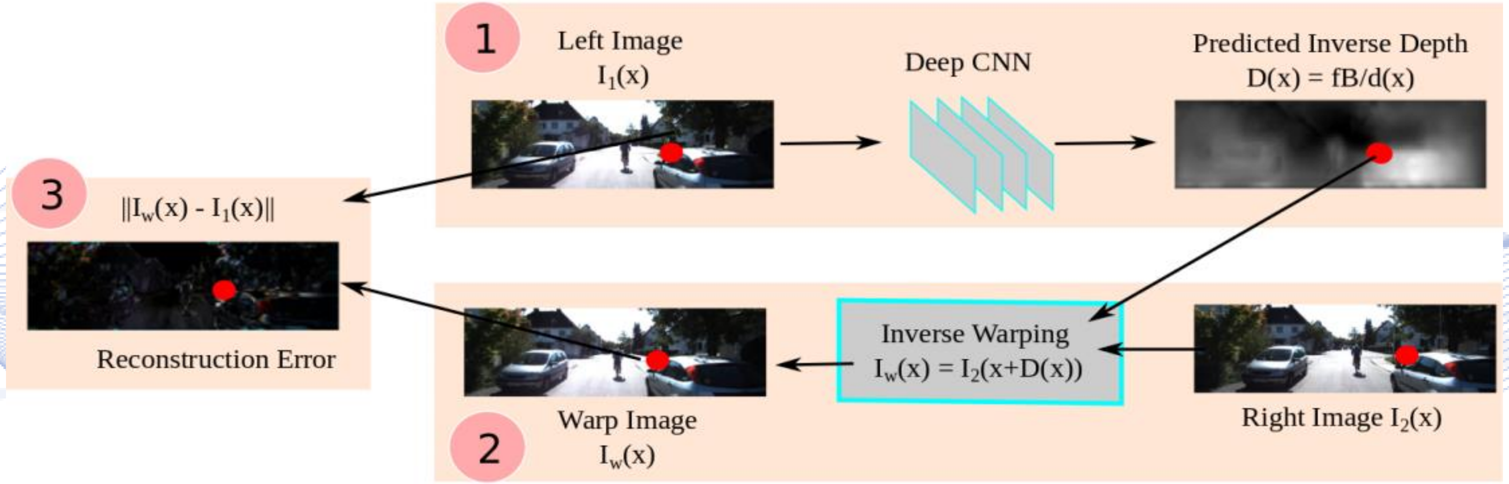


[ZBO2015]

Disparity/Depth map Estimation with NNs

- Next, image f_r is warped to form an approximation f'_l of f_l, such that:

$$f_l(\mathbf{p}_l) \approx f'_l(\mathbf{p}_l) = f_r(\mathbf{p}_r).$$



Disparity/Depth map Estimation with NNs

- Then, the photometric loss function J_p is minimized for optimal depth estimation:

$$J_p = \sum_{\mathbf{p}_l, \mathbf{p}_r \in \mathcal{X}} \| f_l(\mathbf{p}_l) - f_r(\mathbf{p}_r) \|^2 .$$

- $\mathcal{X}$: image domain.
- During DNN training using stereo image pairs, DNN learns to estimate $D(\mathbf{p}_l)$, by minimizing J_p.
- During testing, a monocular image $f(\mathbf{p})$ is fed to DNN to produce the desired depth map $\mathbf{D}$.

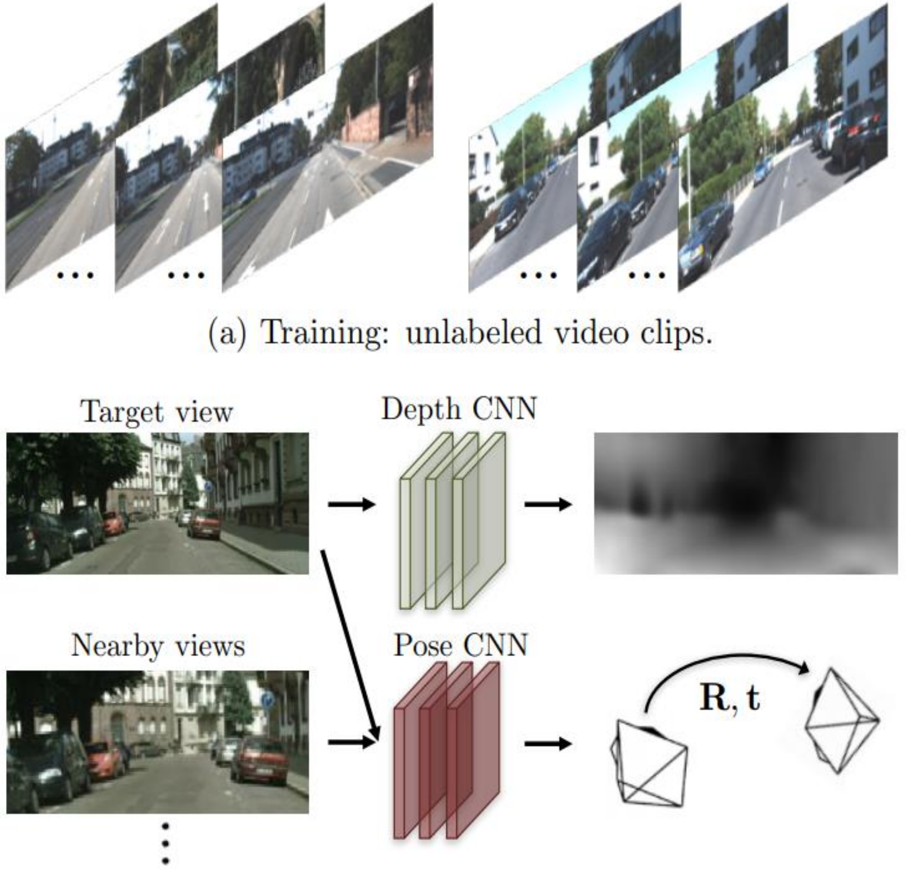
Disparity/Depth map Estimation with NNs

- Then the photometric loss function is computed:

$$J_p(n) = \sum_{\mathbf{p}_n, \mathbf{p}_{n+1} \in \mathcal{X}} \| f(\mathbf{p}_n, n) - f(\mathbf{p}_{n+1}, n+1) \|^2.$$

- The two DNNs are trained to minimize J_p.
- During testing, a monocular image $f(\mathbf{p}, n')$ is fed to the depth estimation DNN to produce the desired depth map $\mathbf{D}$.

Disparity/Depth map Estimation with NNs



(a) Training: unlabeled video clips.

(b) Testing: single-view depth and multi-view pose estimation.

Depth and pose estimation DNNs.

Disparity/Depth map Estimation with NNs

Depth image from monocular video [APOLLO].

Point Cloud Generation

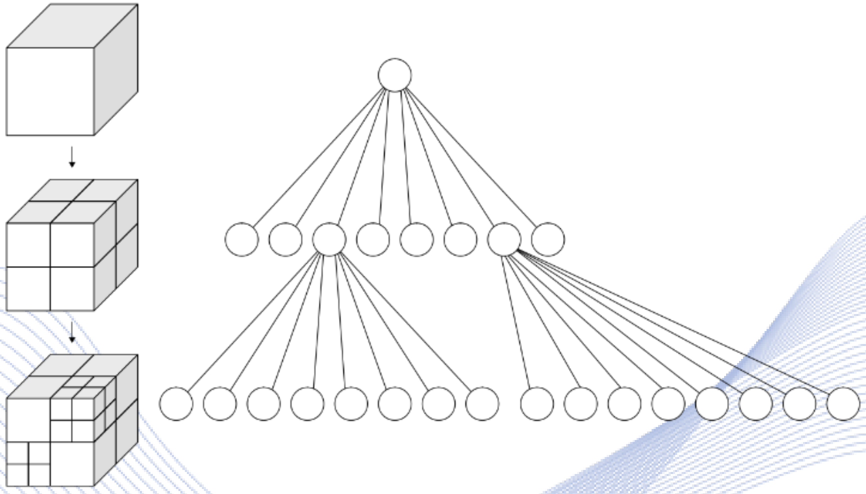
- CNNs (***U-nets*** in particular) can generate 3D point cloud coordinates, if given a single image as input [FAN2017].
- The encoder predicts embeddings from the image and a random vector to perturb the prediction (inspired from GANs).
- The predictor outputs a $N \times 3$ $(N = 1024)$ coordinate matrix having entries $[X_i, Y_i, Z_i]$, $i = 1, \dots, N$.

3D Surface Mesh Estimation

- ***Triangular meshes*** can also be inferred from single images.
- An effective way [WAN2018] is to:
  - progressively deform 3D object mesh using a Graph CNN, starting from trivial mesh, e.g., an ellipsoid;
  - produce the mesh that corresponds to the depicted object, by directly inferring mesh (graph) coordinates.
- The 3D mesh can also be formulated as a set of deformable 2D squares that covers a point cloud [GRO2018].

3D Volumetric Model Estimation

- ***Voxel*** grid object representations can be generated given a single-view depth map, multiview images or a single image as input.
- Suitable networks: 3D CNNs [GAL2017], 3D Recurrent Neural Networks [CHO2016].
- For higher resolution, without further memory needs, ***octree*** object representations have been explored [RIE2017].



3D object octtree.

Neural Semantic 3D World Modeling

- Introduction
- Disparity/Depth Estimation with NNs
- **Joint 3D Scene Geometry and Semantics Estimation**
- Semantic 3D World Maps
- Semantic 3D World Map Annotations

Joint 3D Scene Geometry and Semantics Estimation

- Semantic image segmentation and 3D geometry estimation are highly correlated tasks.
- Simultaneous execution of both tasks allows the creation of a ***semantic 3D map***.

- Further gains:
  - ***Accuracy***: the two tasks can reinforce one another.
  - ***Speed***: possible use of common computational modules (e.g. common image feature extractors) instead of totally separate networks.

Joint 3D Scene Geometry and Semantics Estimation

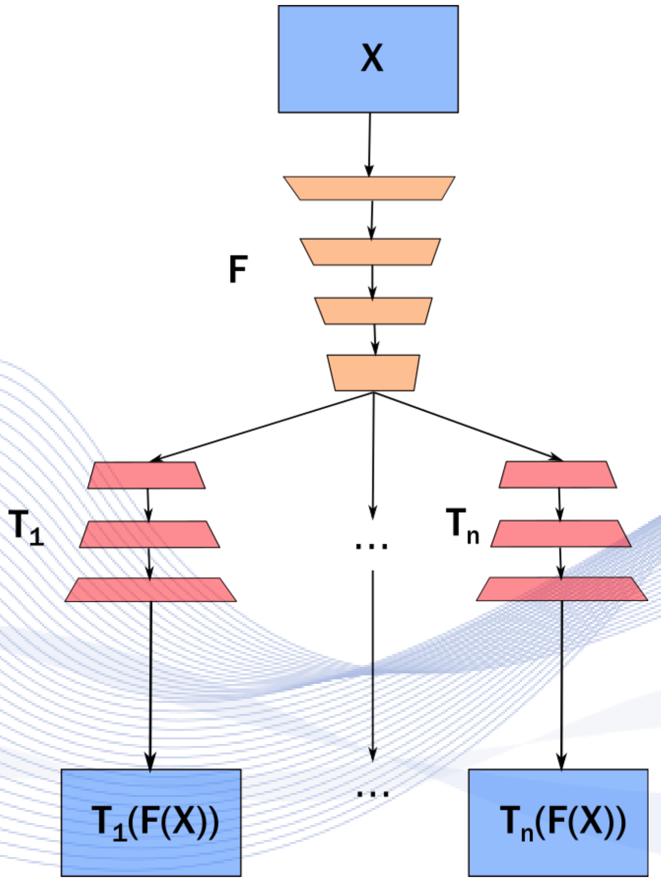
Typical multitask networks have:

- Common input $\mathbf{X}$.
- Common feature extraction operator $\mathbf{F}$.
- n concurrent task operators:

$$\mathbf{T}_1, \dots, \mathbf{T}_n,\ n \geq 2.$$

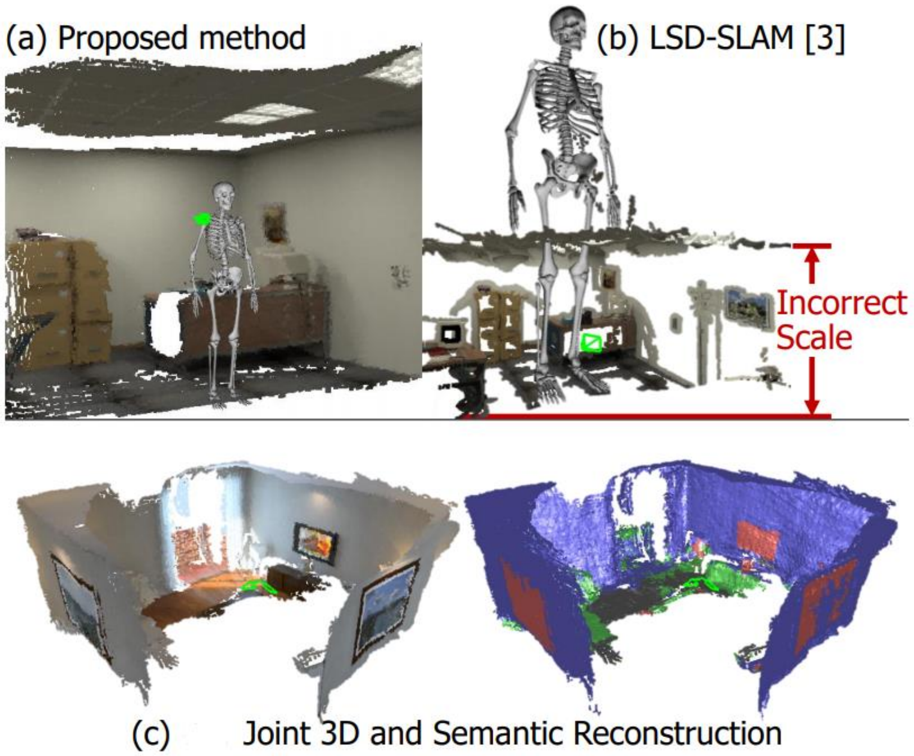
- The **multitask network** output is the set:

$$\mathcal{T} = \{\mathbf{T}_1(\mathbf{F}(\mathbf{X})), \dots, \mathbf{T}_n(\mathbf{F}(\mathbf{X}))\}.$$

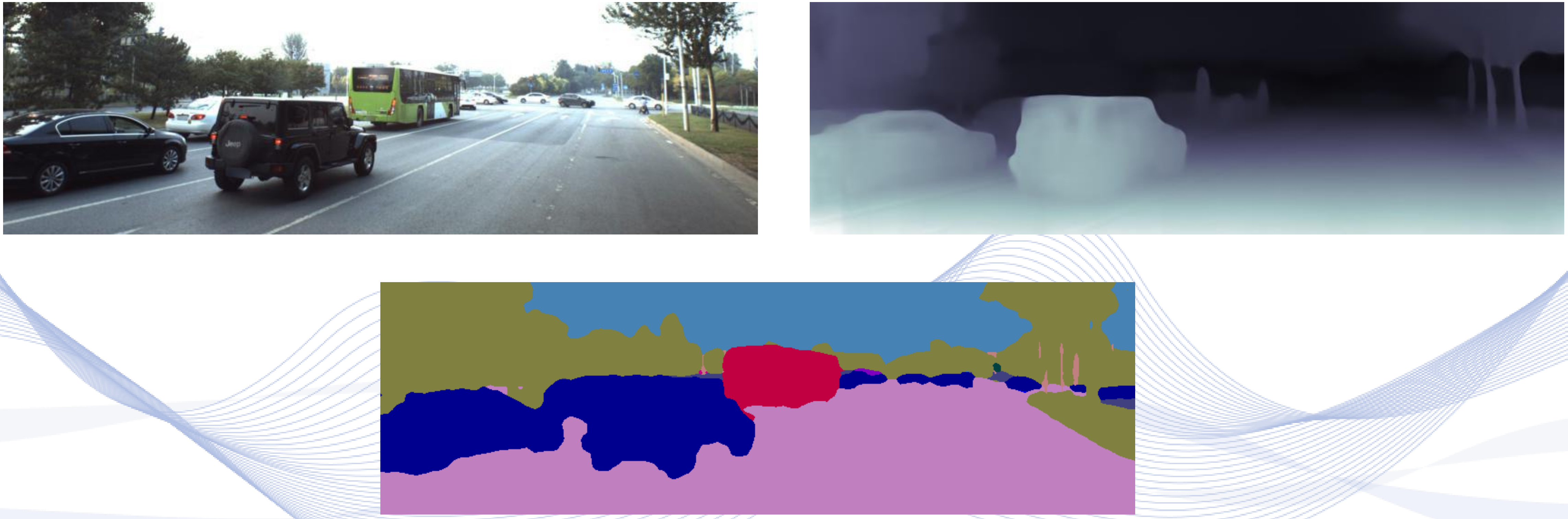


Joint 3D Scene Geometry and Semantics Estimation

- CNN-predicted dense depth maps can be ***fused*** together with depth measurements directly obtained from monocular SLAM [TAT2017].
- CNN-predicted semantic segmentation can be coherently fused with the global 3D scene model.
- It can overcome problems, such as good estimation of the absolute scale, depth prediction in texture-less areas, etc.



Joint 3D Scene Geometry and Semantics Estimation

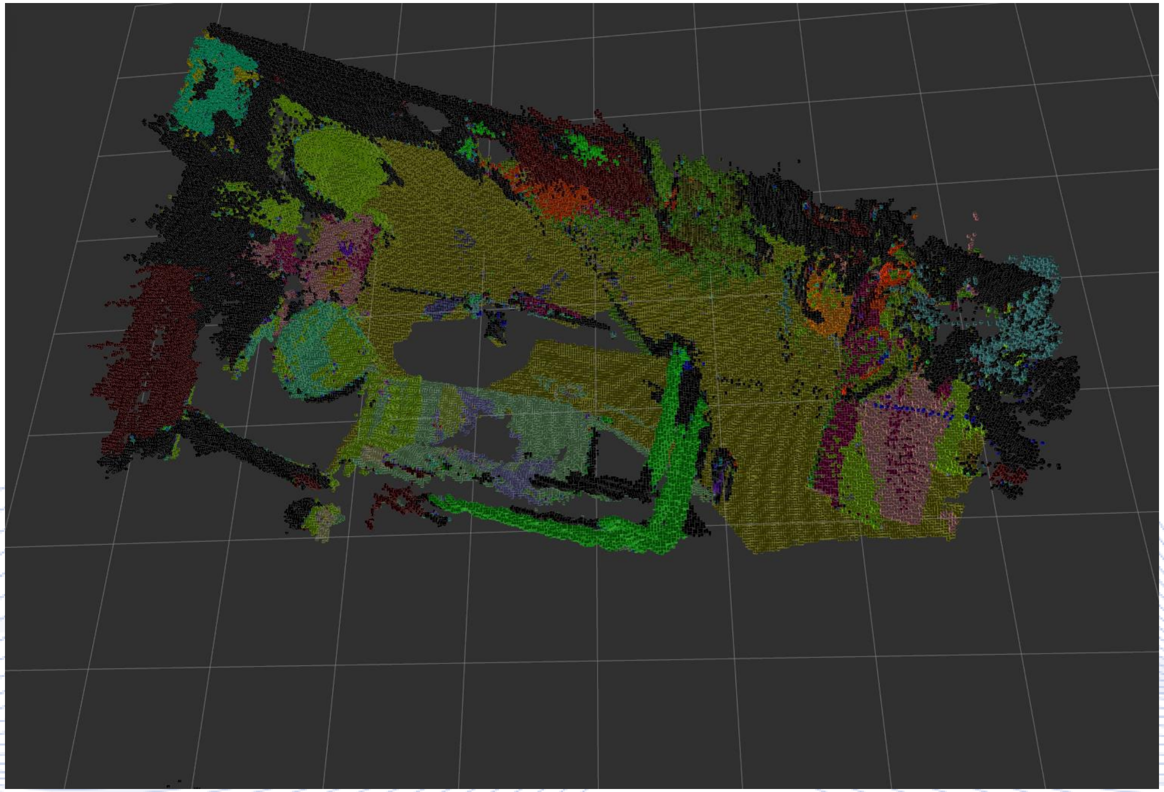


Neural depth image estimation and semantic image segmentation [APOLLO].

Neural Semantic 3D World Modeling

- Introduction
- Disparity/Depth Estimation with NNs
- Joint 3D Scene Geometry and Semantics Estimation
- **Semantic 3D World Maps**
- Semantic 3D World Map Annotations

Semantic 3D World Maps



Semantic octomap [ZHA2018].

Neural Semantic 3D World Modeling

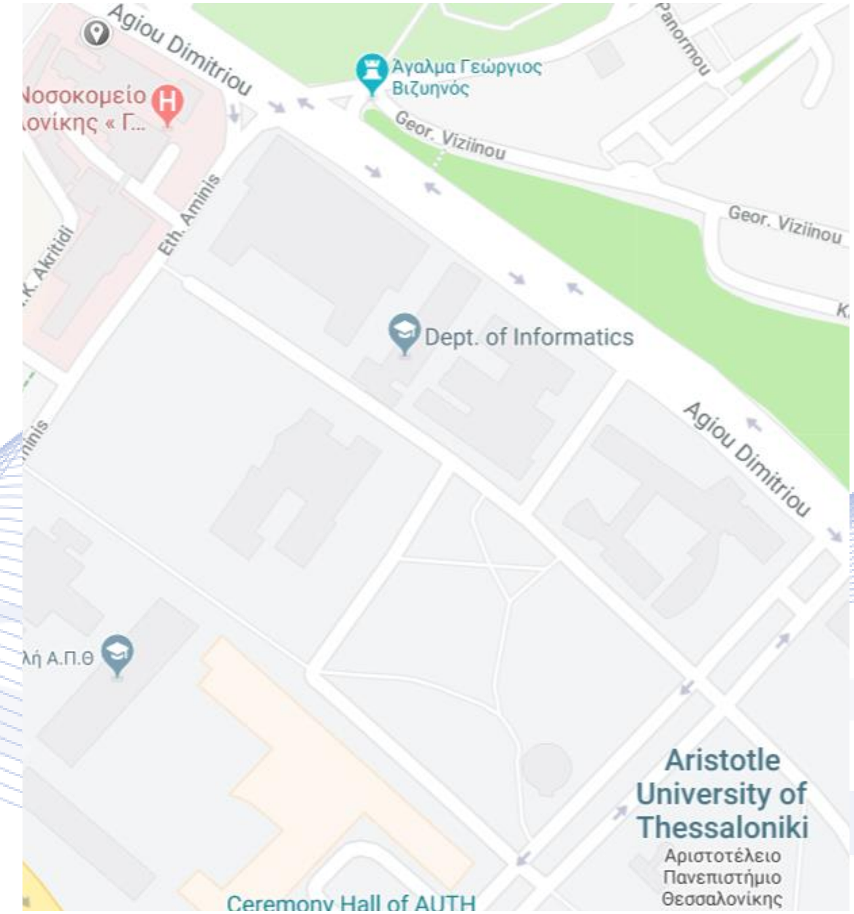
- Introduction
- Disparity/Depth Estimation with NNs
- Joint 3D Scene Geometry and Semantics Estimation
- Semantic 3D World Maps
- **Semantic 3D World Map Annotations**

Sources: 2D maps

- Google Maps.
- OpenStreetMaps.

- Semantic annotated information:
  - (roads, POIs, landing sites) in KML format in Google Maps.
  - roads in OSM (XML) in case of OpenStreetMaps.

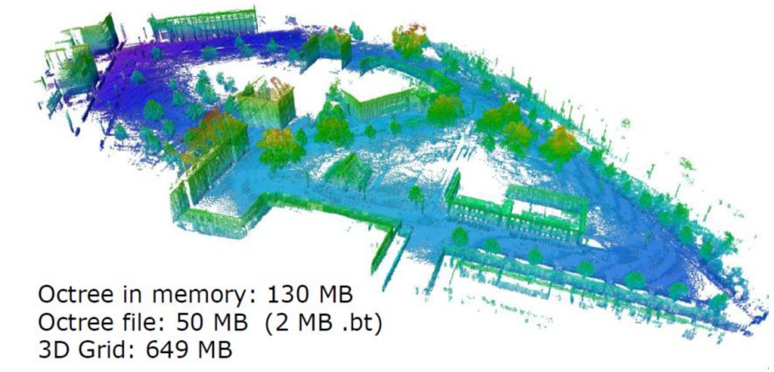
- Google Maps JavaScript API.
- OpenStreetMaps API.



Google Maps

Sources: 3D maps

Octree map (Octomap) of outdoor environment at 0.2 m resolution. Freiburg campus dataset [HOR2013].



- Formats:
  - 3D triangle mesh.
  - 3D Octomap.
- Octomap :
  - The Octomap is a fully 3D model representing the 3D environment, where the UAV navigates.
  - It provides a volumetric representation of space, namely of the occupied, free and unknown areas.
  - It is based on octrees and using probabilistic occupancy estimation.

Semantic Map Annotation types (navigation/logistics)

Type	Static/dynamic	Who	How	Geometric entity type
Regular takeoff and landing sites	Static	Supervisor	Manually	Point
No flight zones	Static	Supervisor	Manually or imported from a file, if available	Polygon (2D coordinates, longitude- latitude)
Potential emergency landing sites	Static	Supervisor	Manually	Polygon
Crowd gathering areas	Dynamic, during production	Visual Semantic annotator, Semantic map manager	Automatically	Polygon (2D coordinates, longitude- latitude)
Points of interest	Static		Manually	Point

Semantic information structure

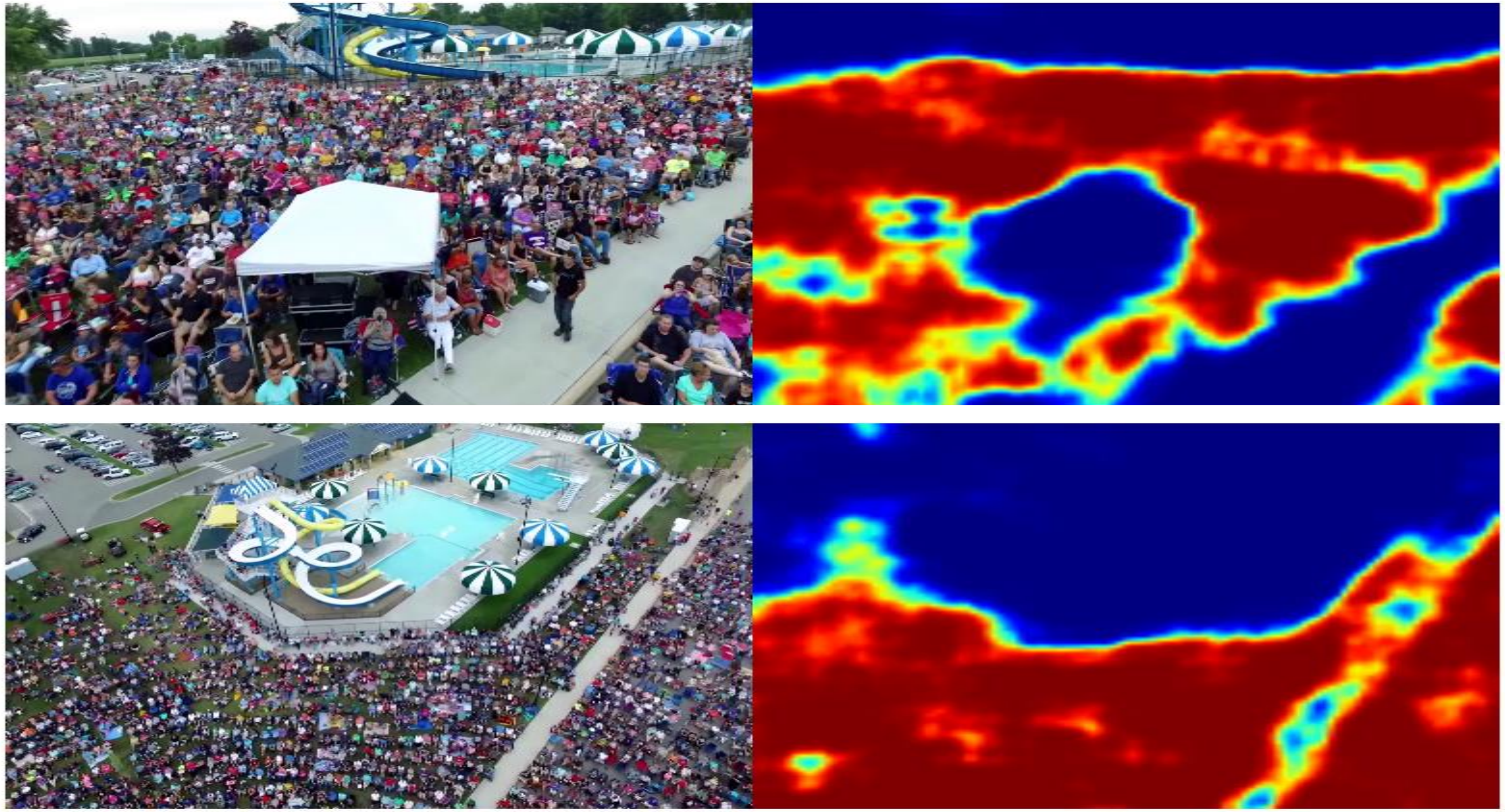
- Static semantic information:
  - Roads, POIs, no-flight zones, private areas.
- Dynamic semantic information:
  - Crowd locations.
- KML format.

Semantic Map Annotation types (static: navigation/logistics)

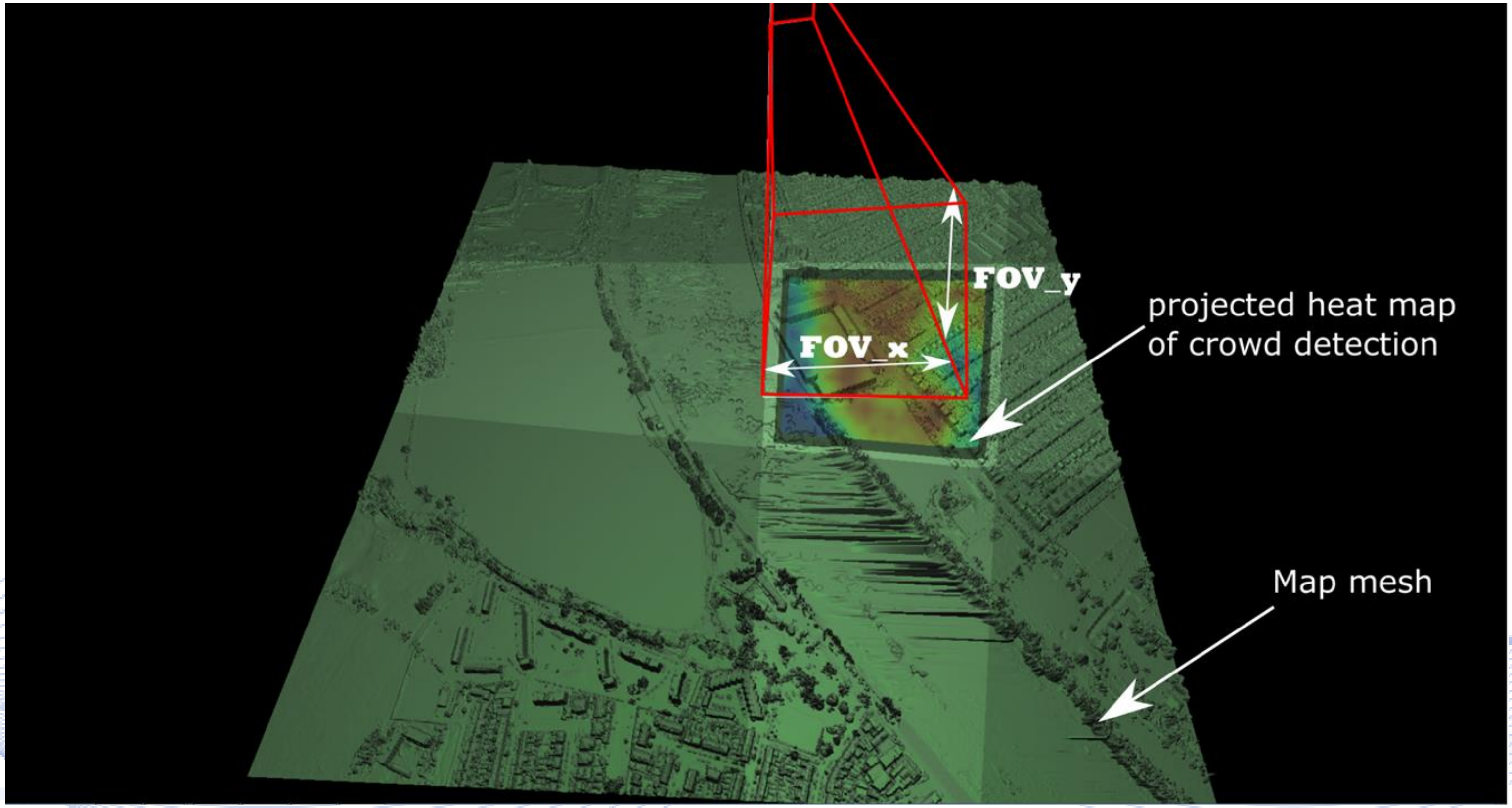
- Static annotations are stored in KML file available from a ROS service in ROS node Semantic Map Manager:

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>KML STRUCTURE</name>
   <Folder>
      <name>Annotations</name>
      <Placemark>
        <name>1
</name>
<address>1.1</address>
        <description> Landing Site/Regular Takeoff Site (re-charging/ relay stations)</description>
        <Point>
          <coordinates>
            22.9662323,40.6832416,0
          </coordinates>
        </Point>
      </Placemark>
….
</kml>
```

Projection of crowd location onto the 3D map

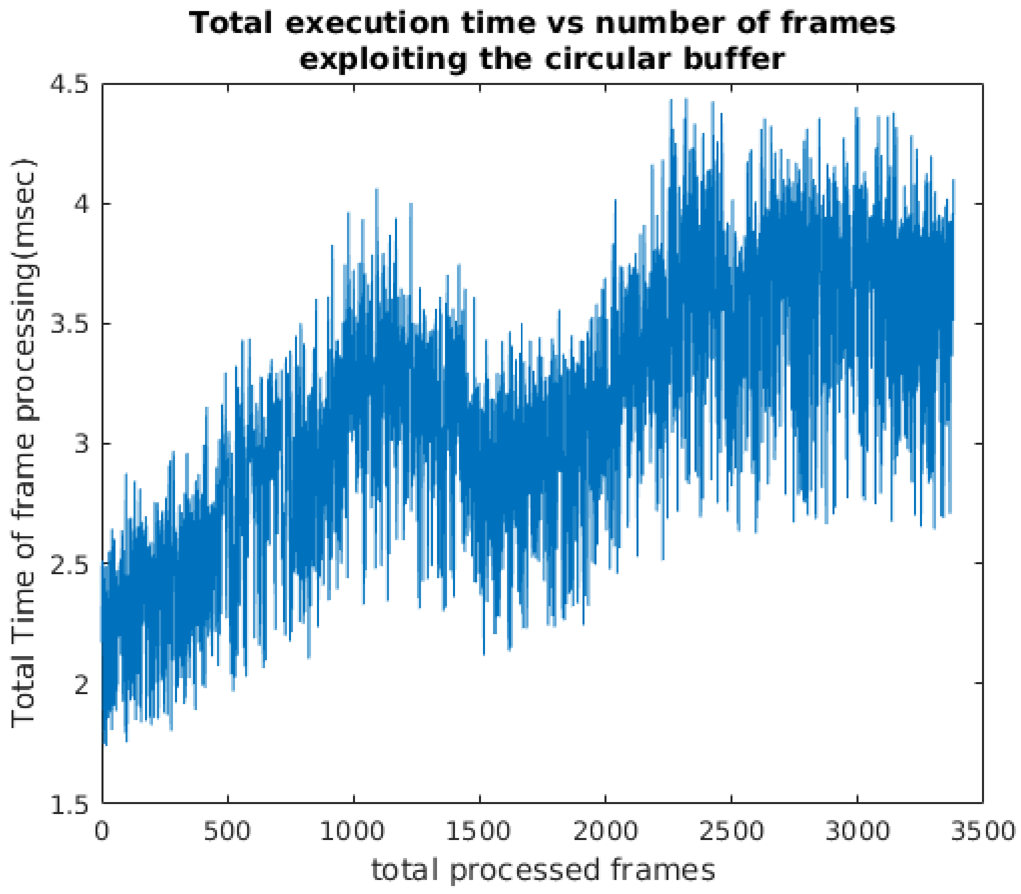


Semantic 3D Mesh Map Annotation



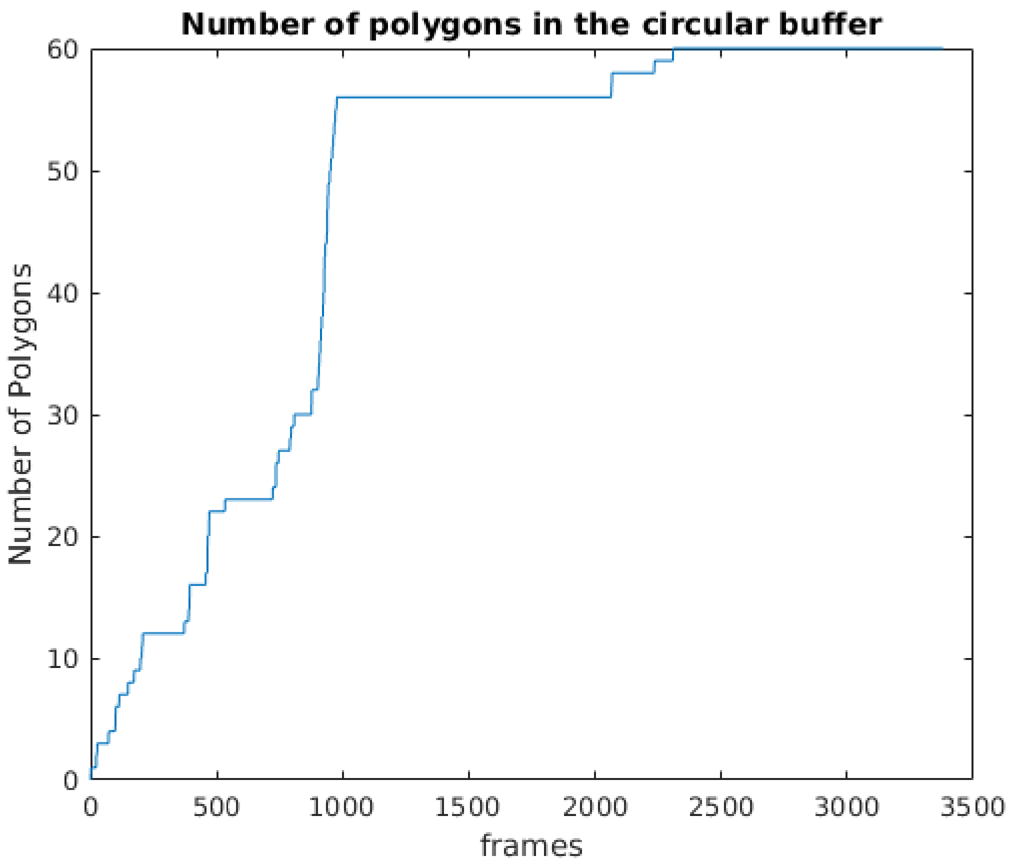
Scalability of semantic map manager

- Total Processing Time of SMM nodes
  - as the circular buffer is being filled in the first 2500 frames the total duration of time processing is increased and
  - when it is filled, the processing time is being stable with a mean value around 3.5msec.



Scalability of semantic map manager

- Storage of the respective 2D polygons
  - Number of polygons in the circular buffer capacity equals to 60 polygons



References

[KAK2019] Kakaletsis, E., Tzelepi, M., Kaplanoglou, P. I., Symeonidis, C., Nikolaidis, N., Tefas, A., & Pitas, I. (2019, January). Semantic map annotation through UAV video analysis using deep learning models in ROS. In International Conference on Multimedia Modeling (pp. 328-340). Springer, Cham.

[HOR2013] Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. Autonomous robots, 34(3), 189-206

Q & A

Thank you very much for your attention!

More material/lectures in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr