ECCV Workshop on Recovering 6D Object Pose

From 3D descriptors to monocular 6D pose: what have we learned?

Federico Tombari
CAMP - TUM

Technische Universität München + Johns Hopkins Whiting School of Engineering

Dynamic occlusion

Low latency

High accuracy, low jitter

No expensive hardware

Features vs Template .. vs Learned

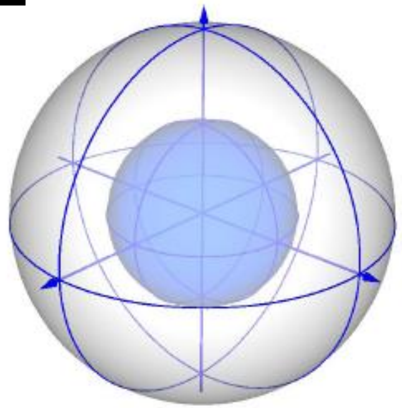
Features

Template matching

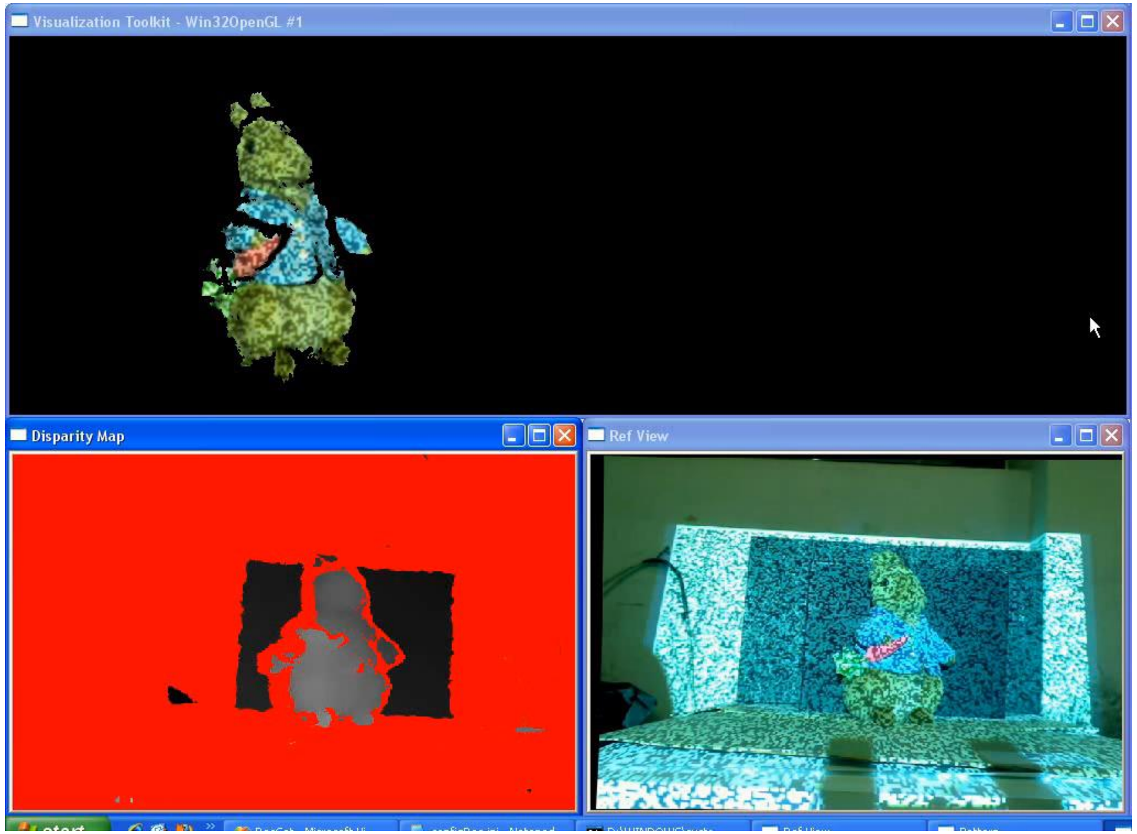
Learned

10 years ago

PR2



SHOT descriptor



Today

INPUT DATA

Mono

Consumer depth

Lidar / sparse

APPLICATIONS

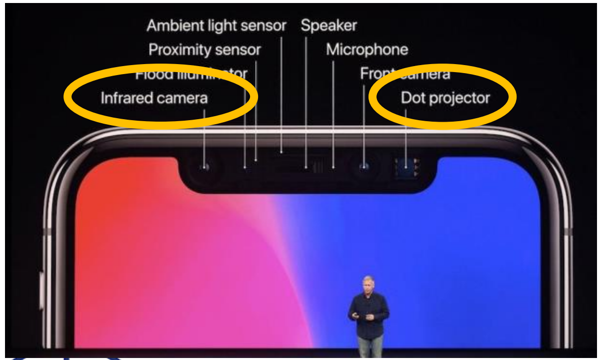
AR for Smartphones

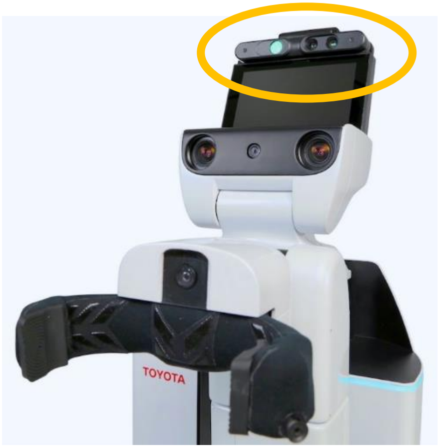
HMDs

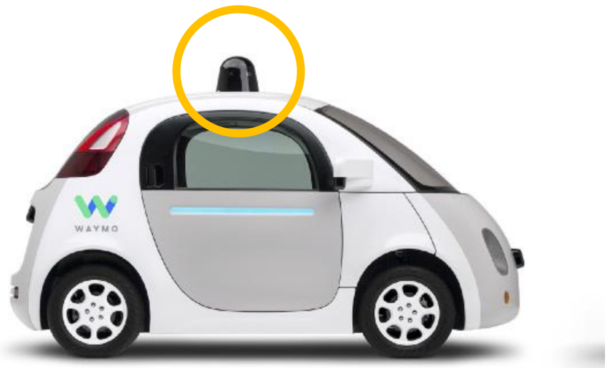
Personal Robotics

Industrial Robotics

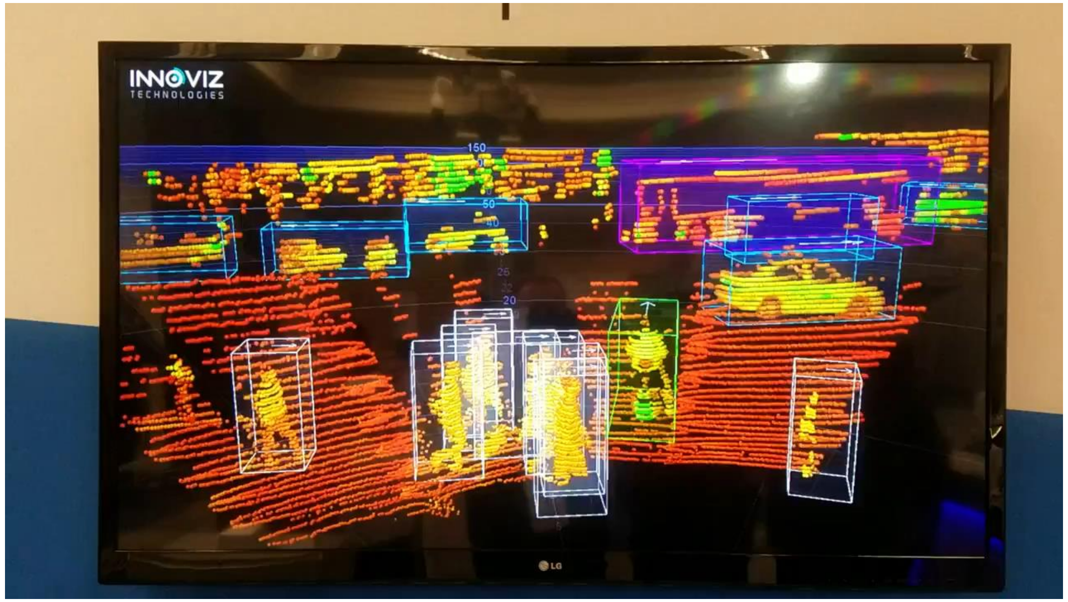
Autonomous Driving



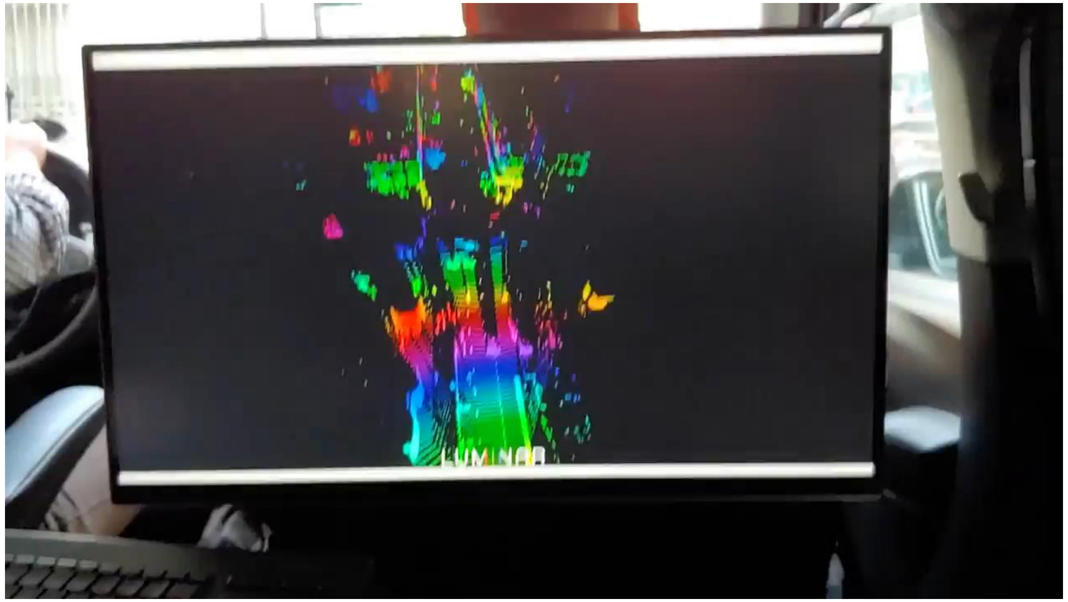




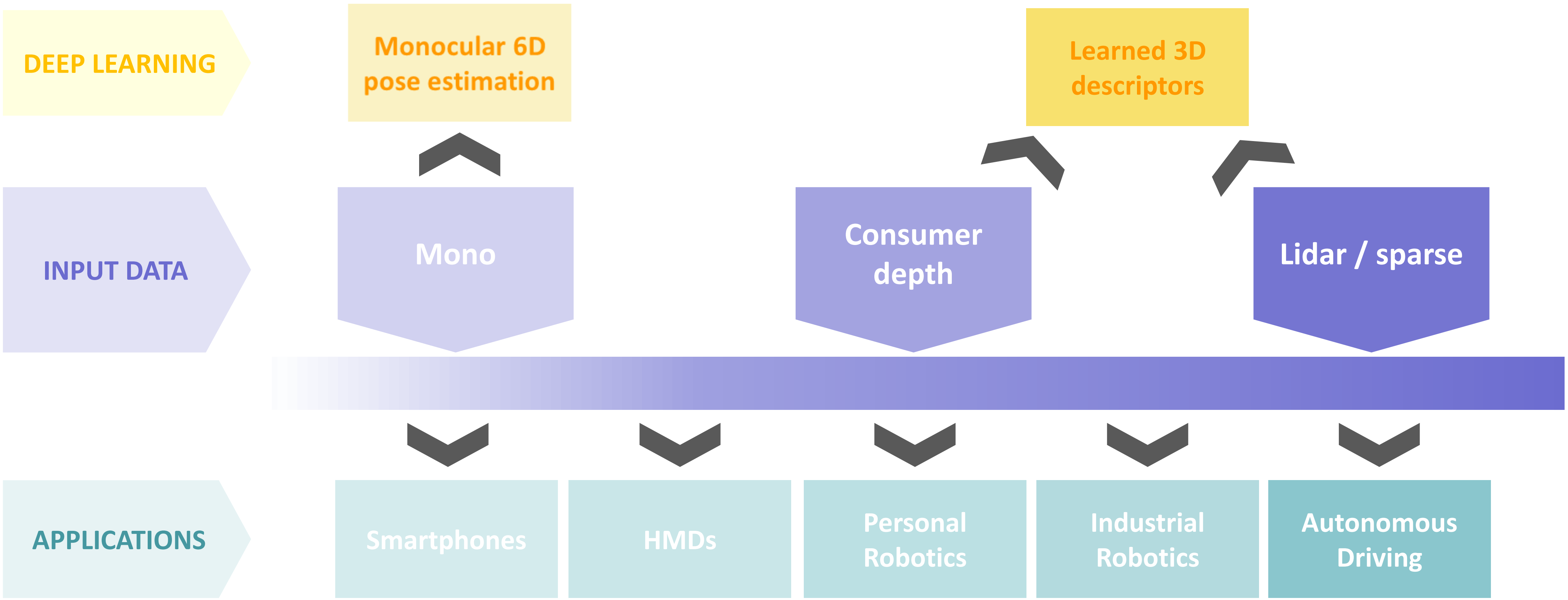
A new generation of cheaper, smaller, denser LIDARs?



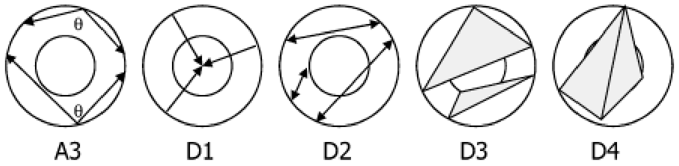
Innoviz



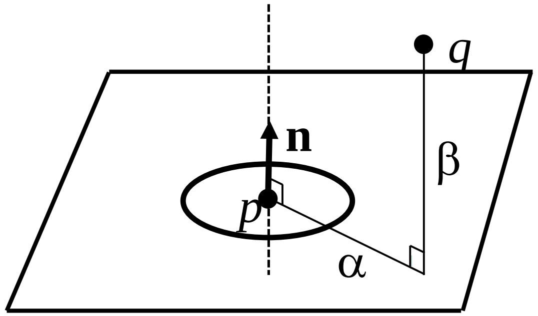
Luminar Technologies


DEEP LEARNING
Monocular 6D pose estimation
Learned 3D descriptors
INPUT DATA
Mono
Consumer depth
Lidar / sparse
APPLICATIONS
Smartphones
HMDs
Personal Robotics
Industrial Robotics
Autonomous Driving

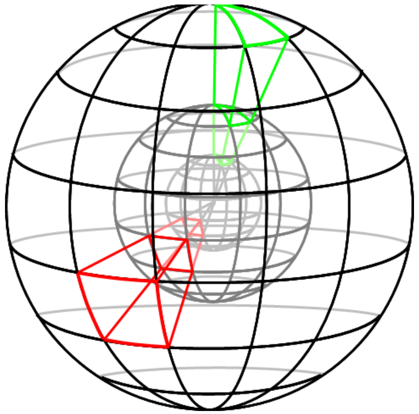
Hand-crafted Local 3D descriptors



Shape distributions [Osada02]



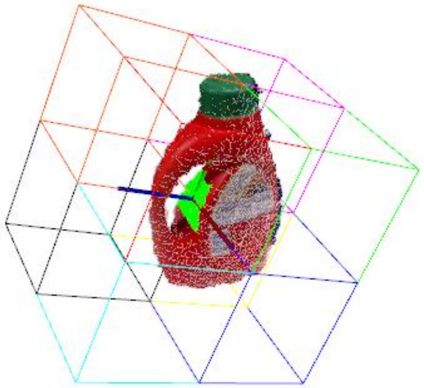
Spin Images [Johnson99]



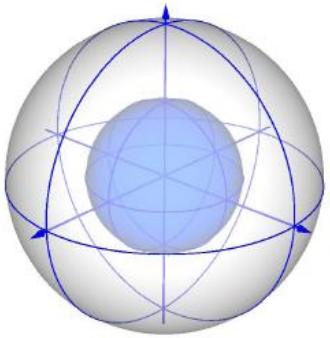
3D Shape Context [Frome04]

Global

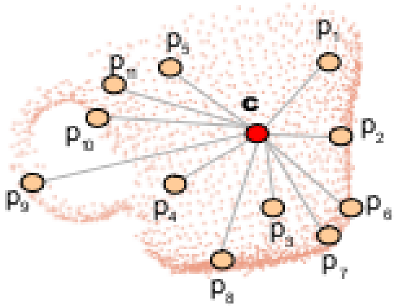
Local



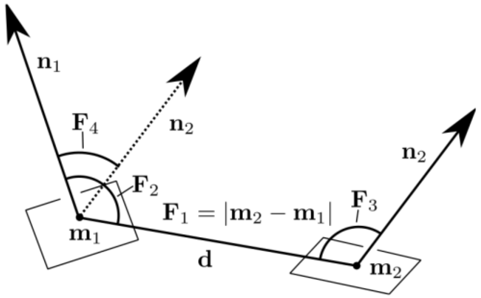
OUR-CVFH [Aldoma12]



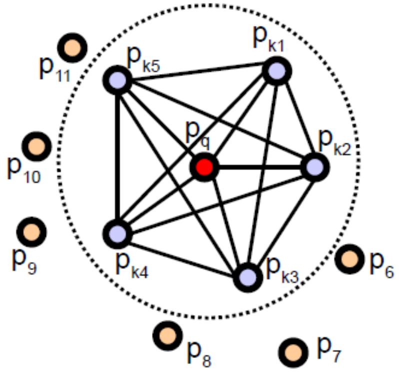
SHOT [Tombari10]



Viewpoint Feature Histogram [Rusu10]



Point Pair Features (PPF) [Drost10]

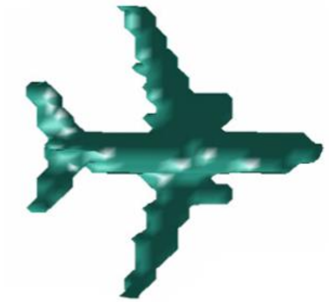


Fast Point Feature Histogram [Rusu09]

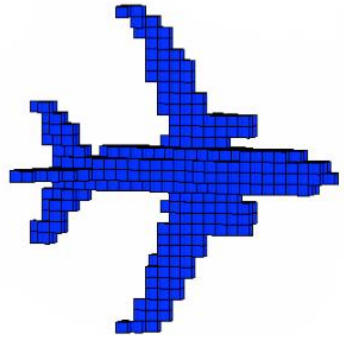
3D representations and deep learning

Point Clouds:
Unorganized, no topology

3D Mesh:
Unorganized, with topology



Voxel map:
Organized, no topology



Range (depth) map:
Organized, no topology

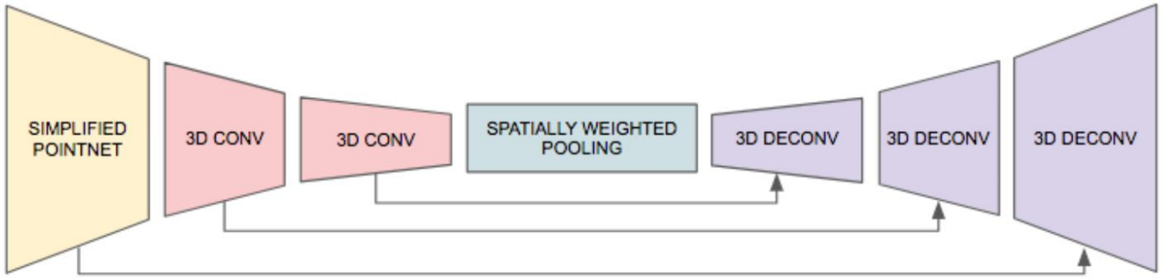
Unorganized 3D representations such as point clouds and meshes are not naturally suited to **convolutions**

Learned 3D descriptors – state of the art

	Input Data	Type	Rotation	Loss function
3DMatch [Zeng17]	Voxel	Local	Training	Contrastive
Compact Geometric Features [Khoury17]	Point clouds (via histograms)	Local	Hand-crafted LRF	Triplet
PPFNet [Deng18]	Point clouds	Local	Hand-crafted LRF	N-tuple
Pointnet [Qi17]	Point clouds	Global	T-net	Classification Segmentation
Pointnet++ [Qi17]	Point clouds	Global	T-net	Classification Segmentation
Dynamic Graph CNN [Wang18]	Point clouds	Global	T-Net	Classification Segmentation

Fully Convolutional Point Network

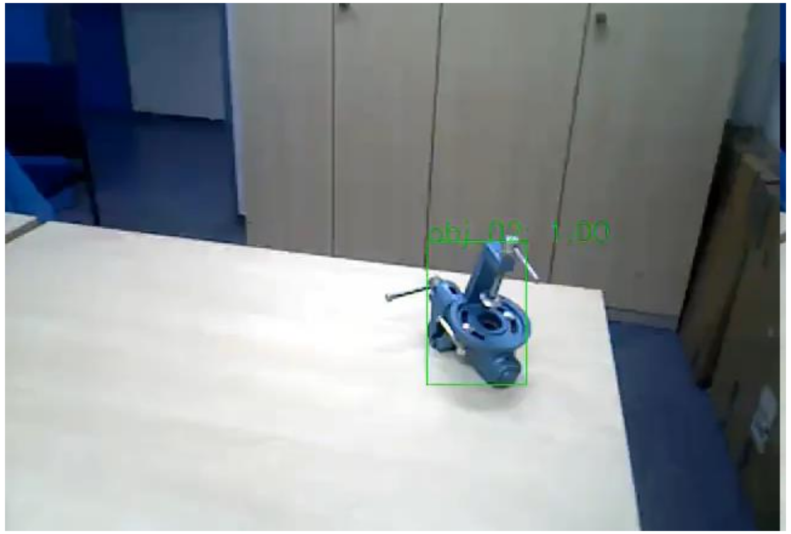
- Hybrid: Unorganized input, organized internal representation and output
- End-to-end, general-purpose, hierarchical learning on unordered 3D data
- Processing of large scale point clouds in one single pass



Point Count	Surface Area	Forward Pass	Memory
150k	$80m^2$	9.1s	9033 MB
36k	$36m^2$	2.9s	8515 MB
15k	$16m^2$	0.57s	6481 MB

Can we “learn” 6D pose without a 3D sensor?

2D vs. 3D object detection and pose estimation



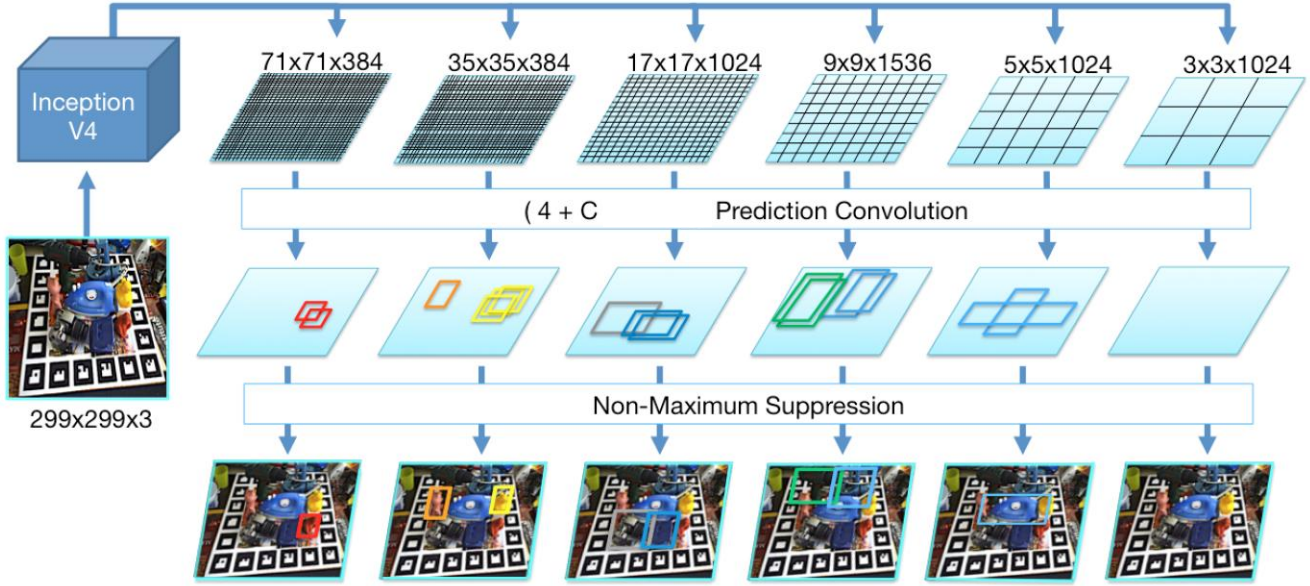
“2D” object detection

Monocular 6D object pose estimation – state of the art

Method	BACKBONE	Network Output	Pose Computation/Refinement
BB-8 [RAD2017]	VGG 16	8 corners of the projected 3D Bounding Box	PnP / VGG
[TEKIN2018]	YOLO V2	8 corners of the projected 3D Bounding Box + 3D centroid projection	PnP
POSECNN [XIANG2018]	VGG 16	Semantic Labeling + Regression of 6D pose	
DEEP 6D POSE [DO2018]	Mask R-CNN	Object Instance Segmentation + Regression of 6D pose	
SSD-6D [KEHL 2017]	SSD 300	Viewpoint and In-Plane rotation classification	Contour-based

- [Rad2017] Rad and Lepetit BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth, ICCV2017
- [Kehl2017] Kehl et al. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again, ICCV 2017
- [Tekin2018] Tekin et al. Real-Time Seamless Single Shot 6D Object Pose Prediction, CVPR 2018
- [Xiang2018] Xiang et al. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes, RSS 2018
- [Do2018] Do et al. Deep-6DPose: Recovering 6D Object Pose from a Single RGB Image, Arxive 2018

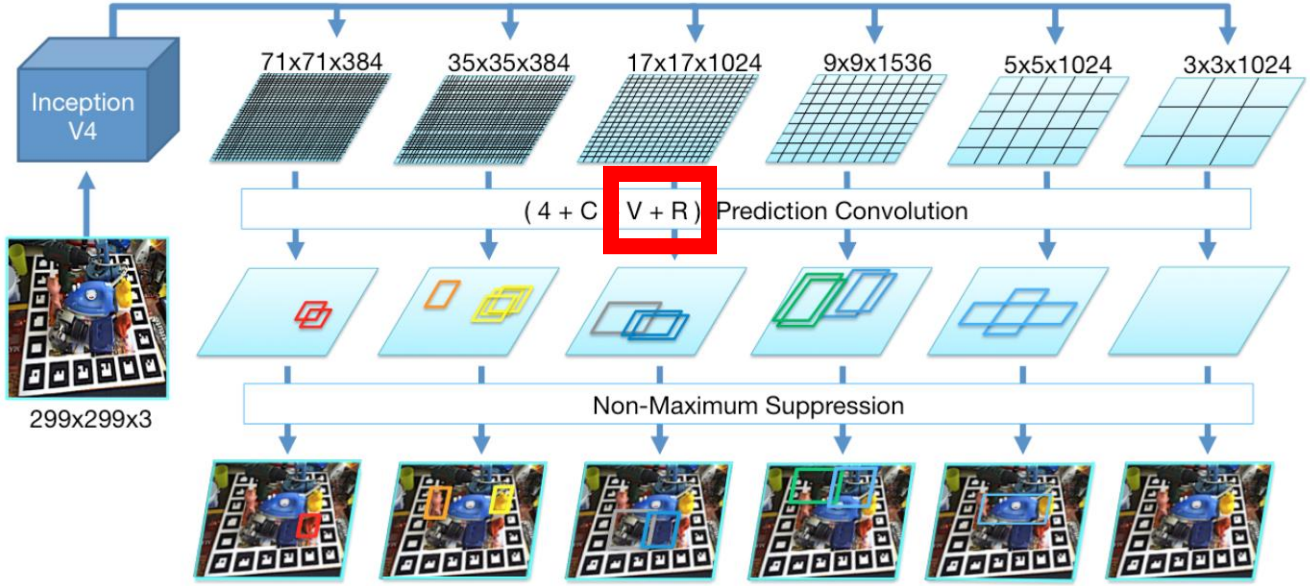
SSD-6D: monocular object detection and 6DoF pose estimation [Kehl17]



Bounding box location (4 values)
C: Probability for each class

Single Shot Detector (SSD) network from [Liu16]

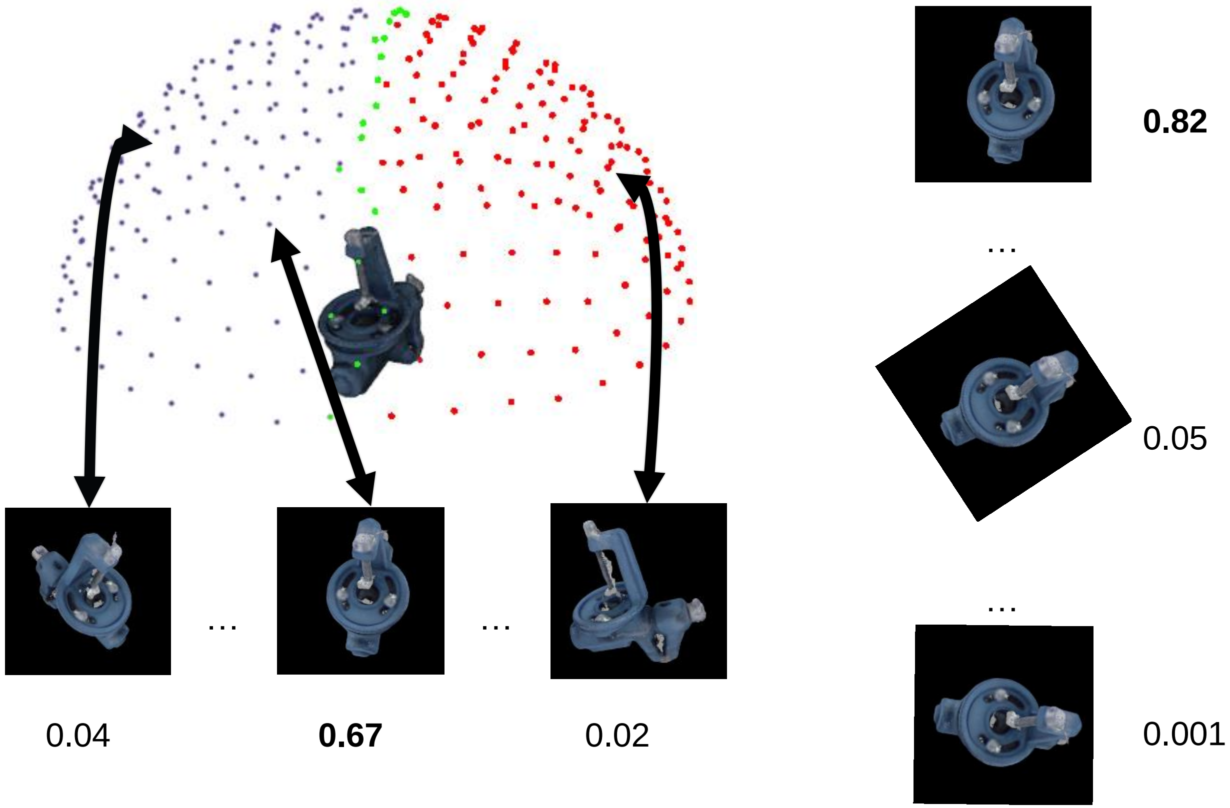
SSD-6D: monocular object detection and 6DoF pose estimation [Kehl17]



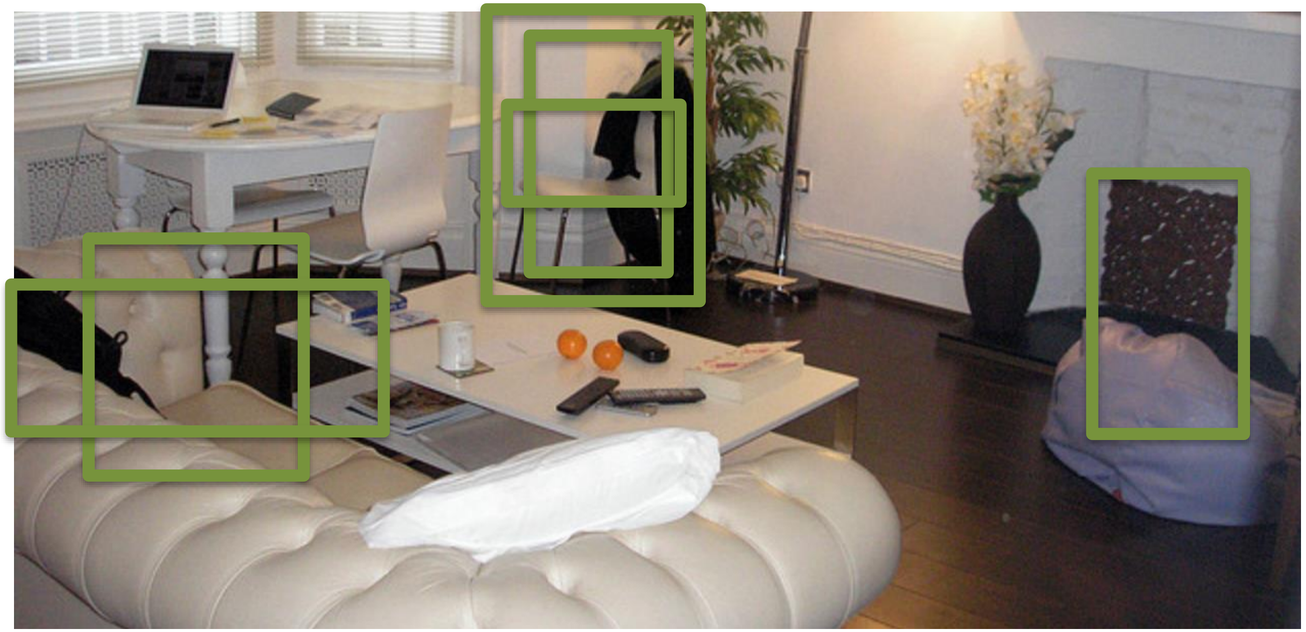
Bounding box location (4 values)
C: Probability for each class

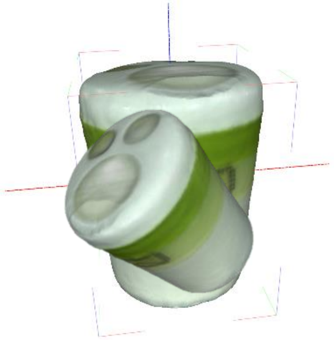
V: Probability for each viewpoint
R: Probability for each in-plane rotation

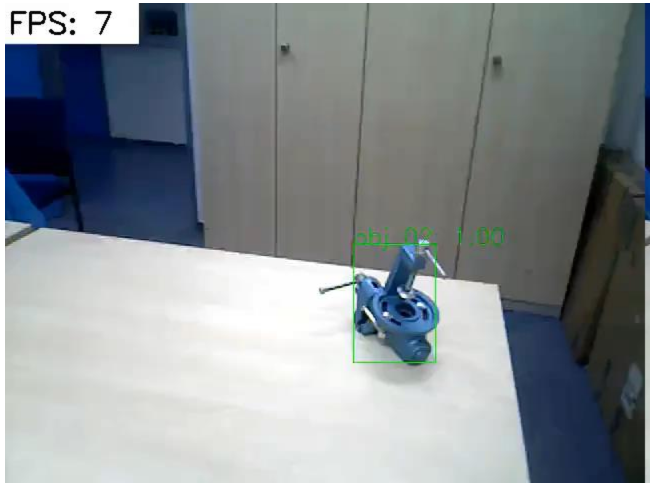
From pose regression to pose classification

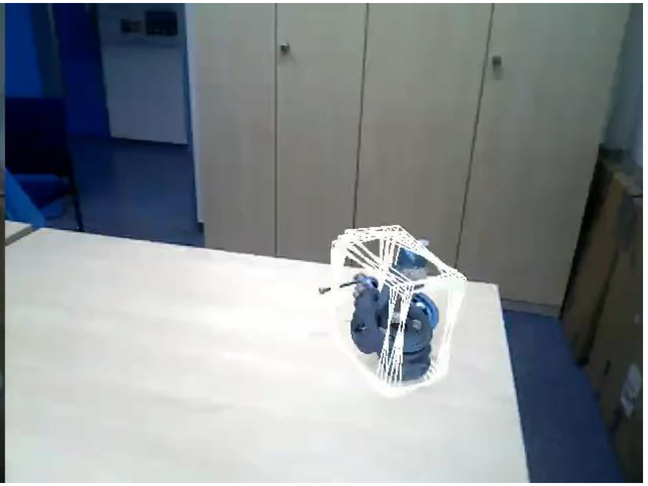


Training



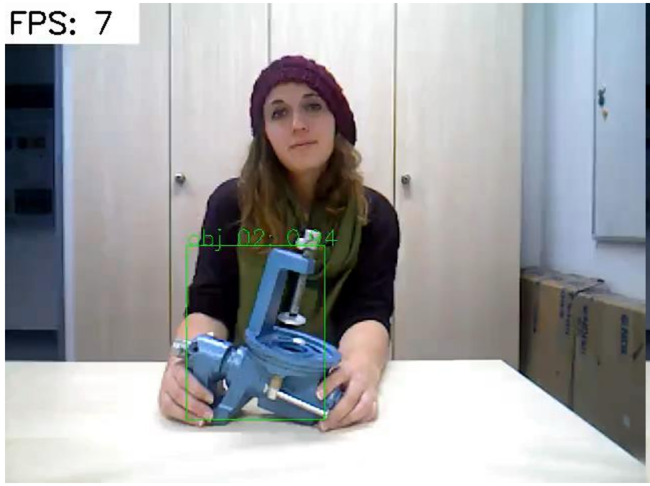


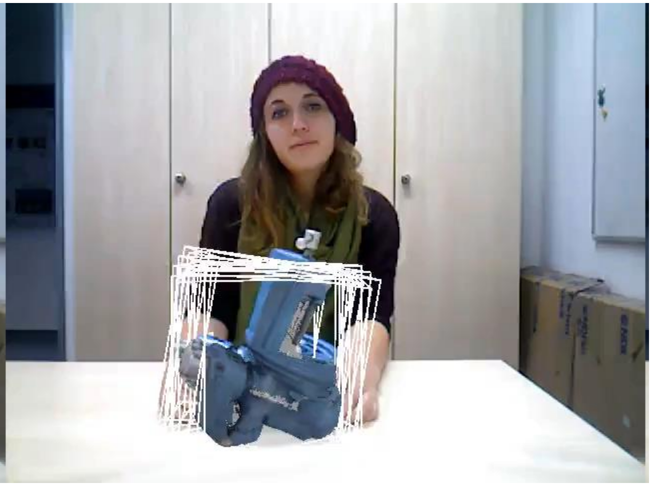




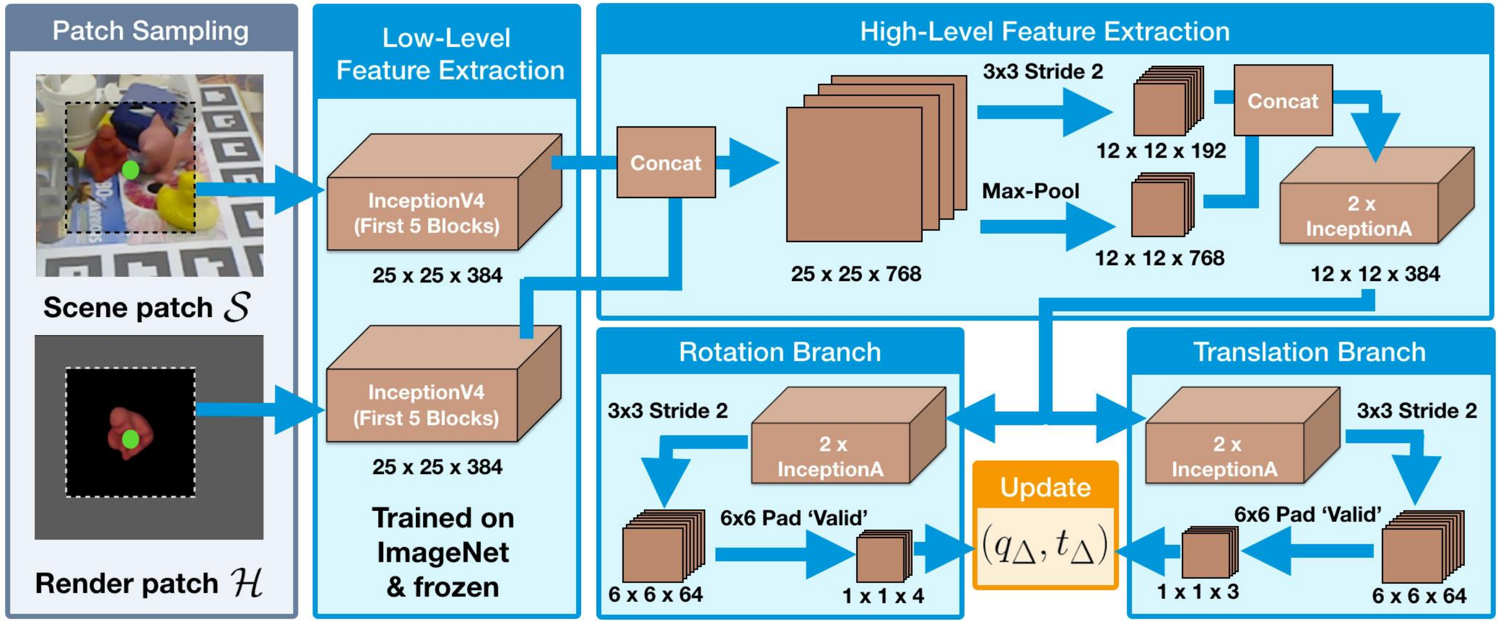
2D Detections

6D Pose estimation
(not refined)





Deep monocular 6D pose refinement



Deep-learned 6D pose refinement method that:

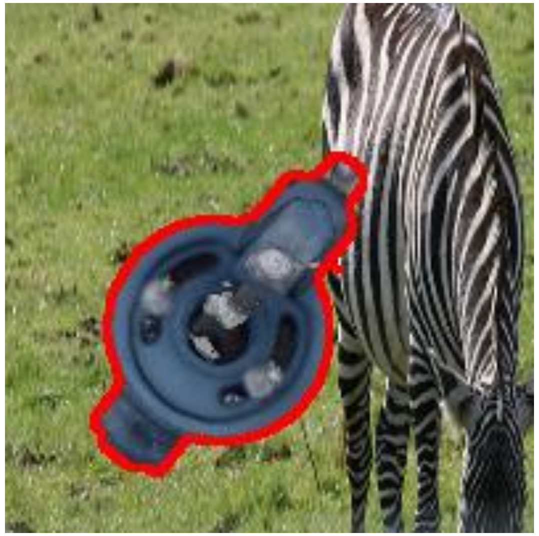
- uses RGB data only
- trained purely on synthetic data
- agnostic to geometrical symmetry and visual ambiguities

Provided a 3D CAD model, input scene image and 6D pose hypothesis, we

- render the model in a patch
- cut out a scene patch around the pose hypothesis
- feed both to a pre-trained feature extractor
- regress a rotational and translational pose update

F. Manhardt, W. Kehl, N. Navab, F. Tombari, "Deep Model-Based 6D Pose Refinement in RGB", ECCV 2018

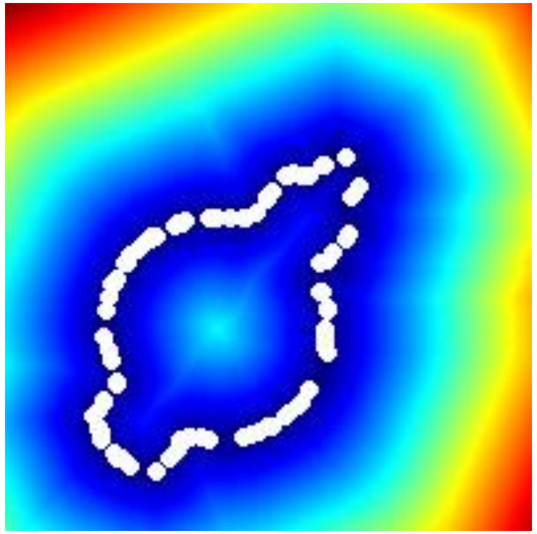
Proxy loss with distance transform



Synthetic Input Image

6D Pose Hypothesis

Pose Estimation at Initial State



Pose Estimation after convergence

$$\mathcal{L}(q_\Delta, t_\Delta, \mathcal{D}_\mathcal{S}, V_\mathcal{H}) := \sum_{v \in V_\mathcal{H}} \mathcal{D}_\mathcal{S}\left[\pi\left(q_\Delta \cdot v \cdot q_\Delta^{-1} + t_\Delta\right)\right]$$

Sum over all sampled points projected on the distance transform of the target.

$$\mathcal{L} := \mathcal{L}(q_\Delta, t_\Delta, \mathcal{D}_\mathcal{S}, V_\mathcal{H}) + \mathcal{L}(q_\Delta^{-1}, -t_\Delta, \mathcal{D}_\mathcal{H}, V_\mathcal{S})$$

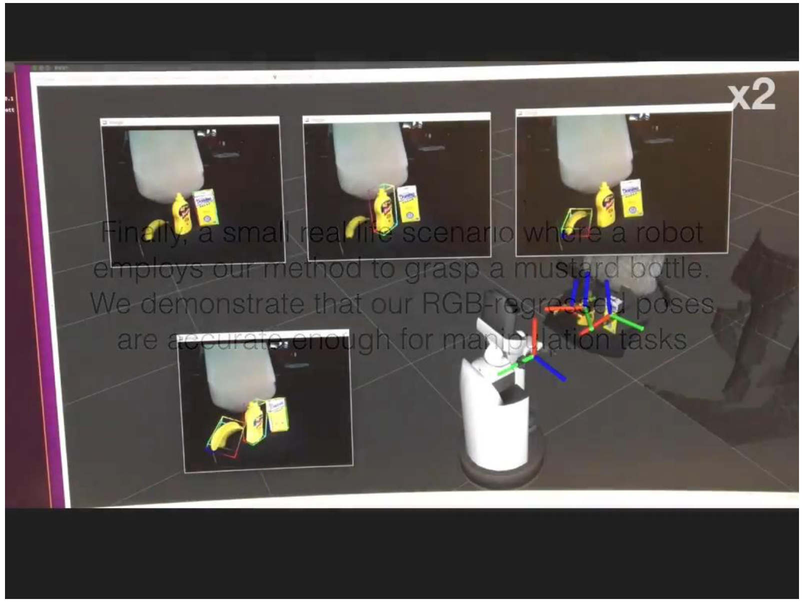
Extension of the loss to both directions, since sampled contour points do not originate from target contours.

Results – deep monocular 6D pose refinement

Tracking of unseen class instances

	Rot. Error [°]	Transl. Error [mm]
No Ref.	27.96	9.75, 9.33, 71.09
3D ICP	17.62	10.42, 10.56, **27.31**
Ours	**16.17**	**4.9**, **5.87**, 42.69

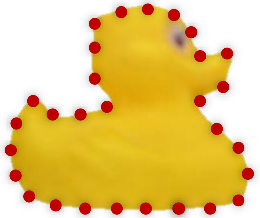
Pose Errors on LineMOD with Poses initialized from SSD-6D



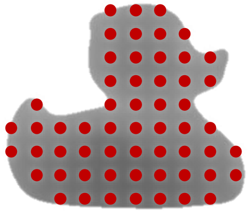
Robotic grasping application

Combining learners and optimizers – RGBD tracking

$$\epsilon_m = \mathbf{n}_m^\top \left[\mathbf{X}_m - \mathbf{T}(\delta\theta)\ \mathbf{T}_{t-1}\mathbf{X}_s\right]$$



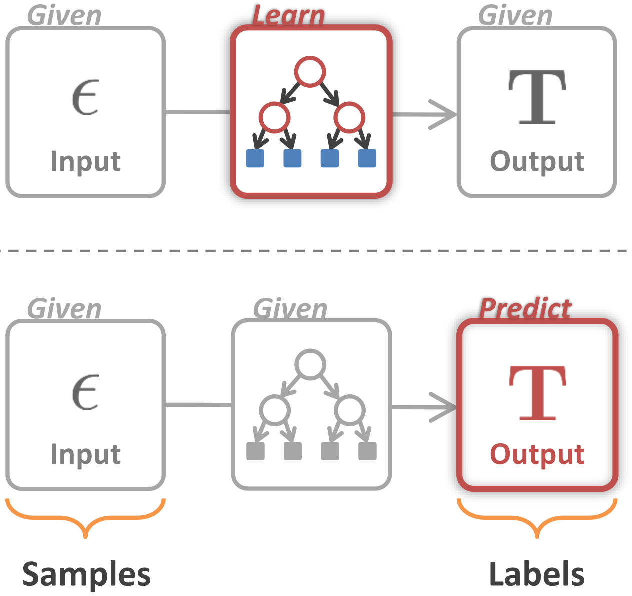
RGB Image
Points on the Edge



Depth Image
Points on the **Surface**

Learning
Use Forest to find the relation between input and output.

Prediction
Given the input, predict the output using the Forest.



Learning-based Method
+ More Robust
✕ Less Accurate

Energy-based Optimization
✕ Less Robust
+ More Accurate

D.J.Tan, N. Navab, F. Tombari, “Looking Beyond the Simple Scenarios: Combining Learners and Optimizers in 3D Temporal Tracking“, ISMAR 2017 (Best Demo Award)

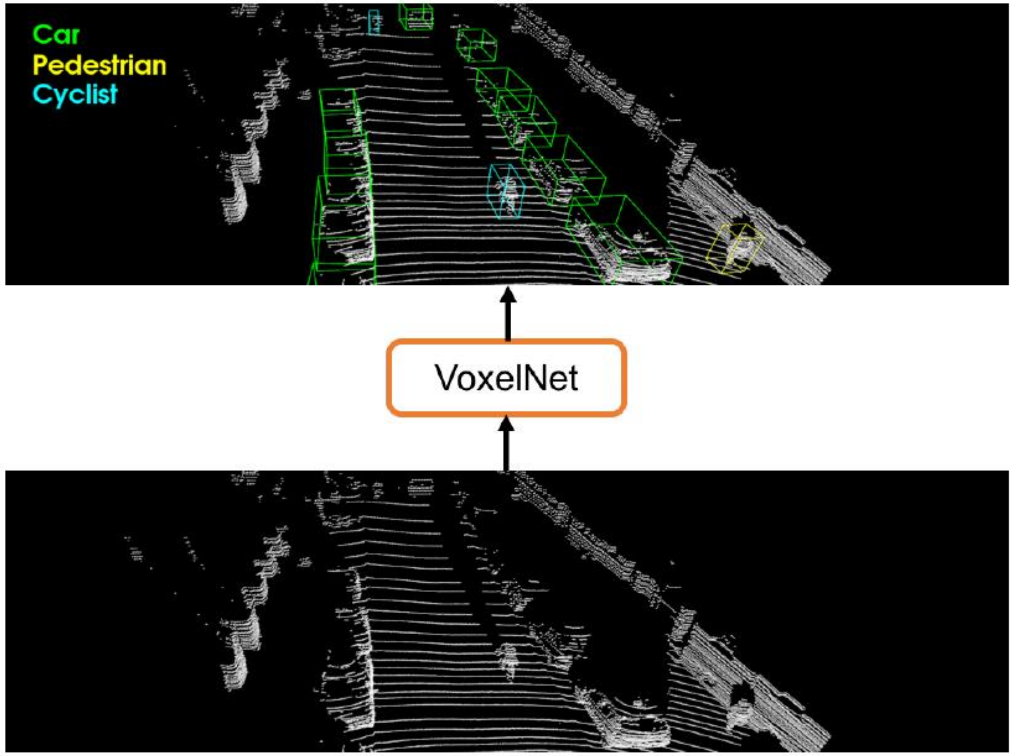
Comparison – monocular vs RGBD 6D pose tracking

Monocular pose refinement [Manhardt18]

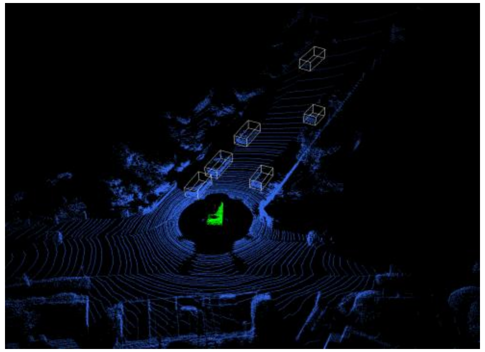
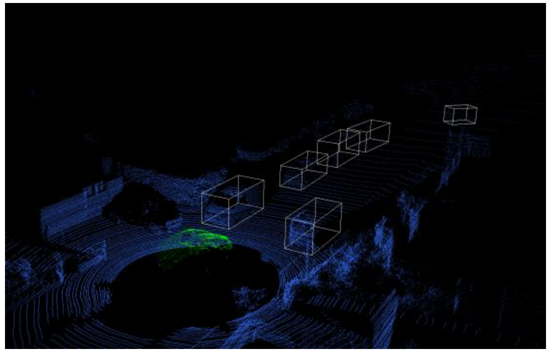
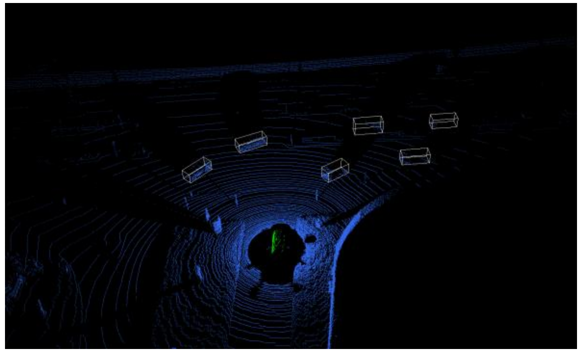
RGB-D pose refinement [Tan17]

Pose estimation for Autonomous Driving

- State of the art techniques mostly rely on LIDAR (or LIDAR+RGB)
- State of the art accuracy around 50% - 70%
- Current Contenders (Multimodal, Lidar only):
  - **VoxelNet:** Zhou and Touzel, 2017
  - **AVOD-FPN**: J. Ku, M. Mozifian, J. Lee, A. Harakeh and S. Waslander: Joint 3D Proposal Generation and Object Detection from View Aggregation. IROS 2018.
  - **F-PointNet:** C. Qi, W. Liu, C. Wu, H. Su and L. Guibas: Frustum PointNets for 3D Object Detection from RGB-D Data. arXiv 2017.
  - **MV3D:** X. Chen, H. Ma, J. Wan, B. Li and T. Xia: Multi-View 3D Object Detection Network for Autonomous Driving. CVPR 2017.

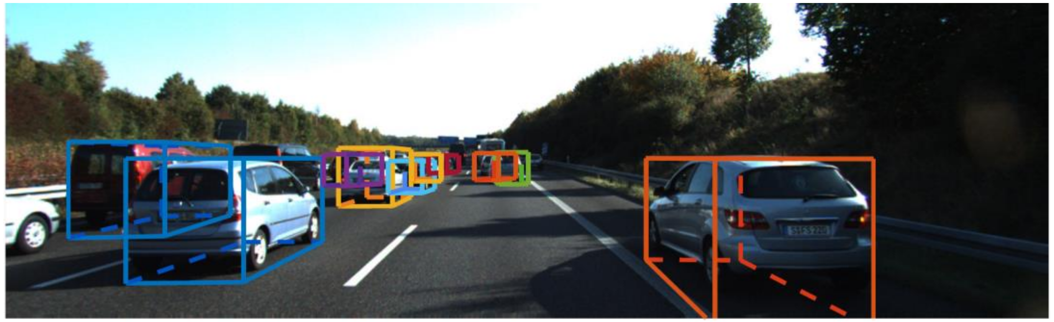


VoxelNet Detections (courtesy of Zhou and Touzel)

Monocular 6D pose also for AD?

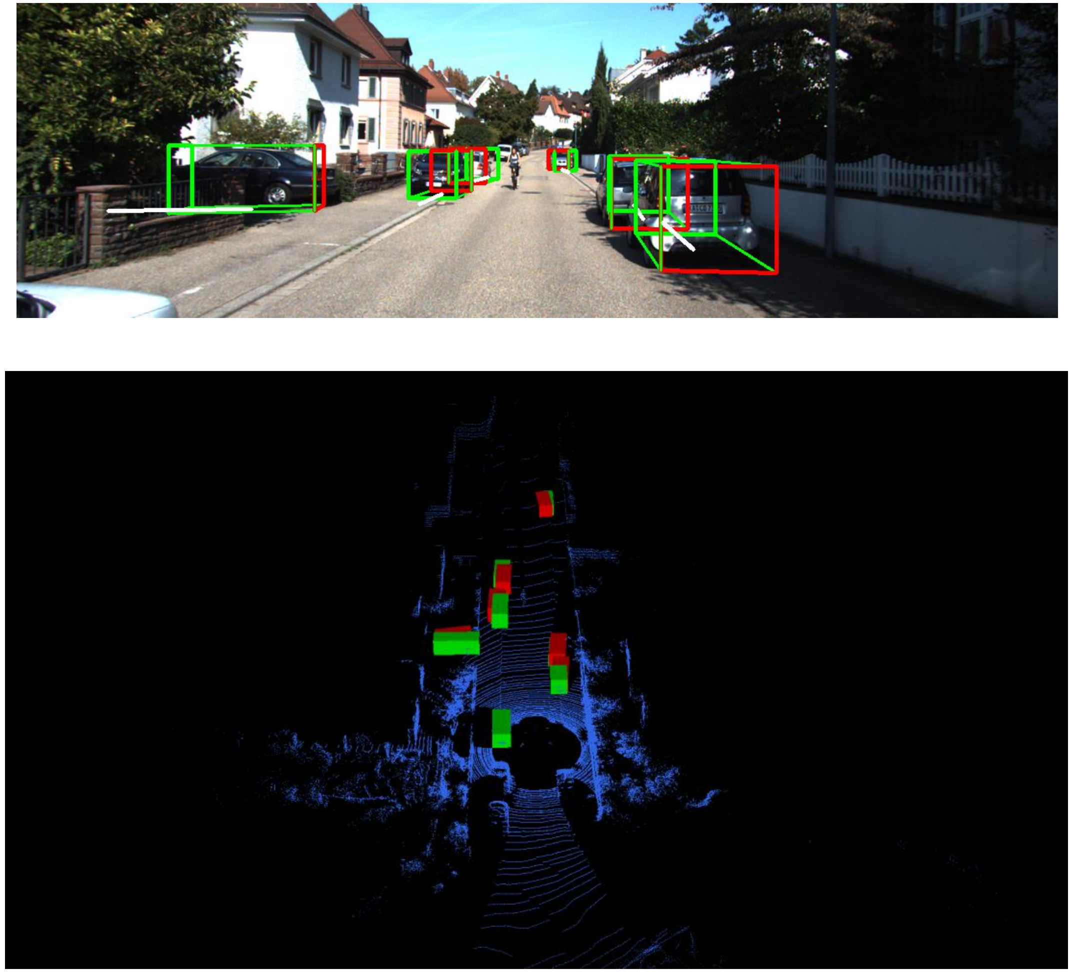
- Extend 2D detection to predict 3D bounding boxes/6D pose for AD classes (e.g. vehicles)
- Still very open problem (between 3 and 6% accuracy for KITTI 3D detection with IoU=0.7)
- Related work (Mono, Stereo):
  - **Mono3D:** X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3d object detection for autonomous driving. In CVPR, 2016
  - **3DOP:** X. Chen, K. Kundu, Y. Zhu, A. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals for accurate object class detection. In NIPS, 2015

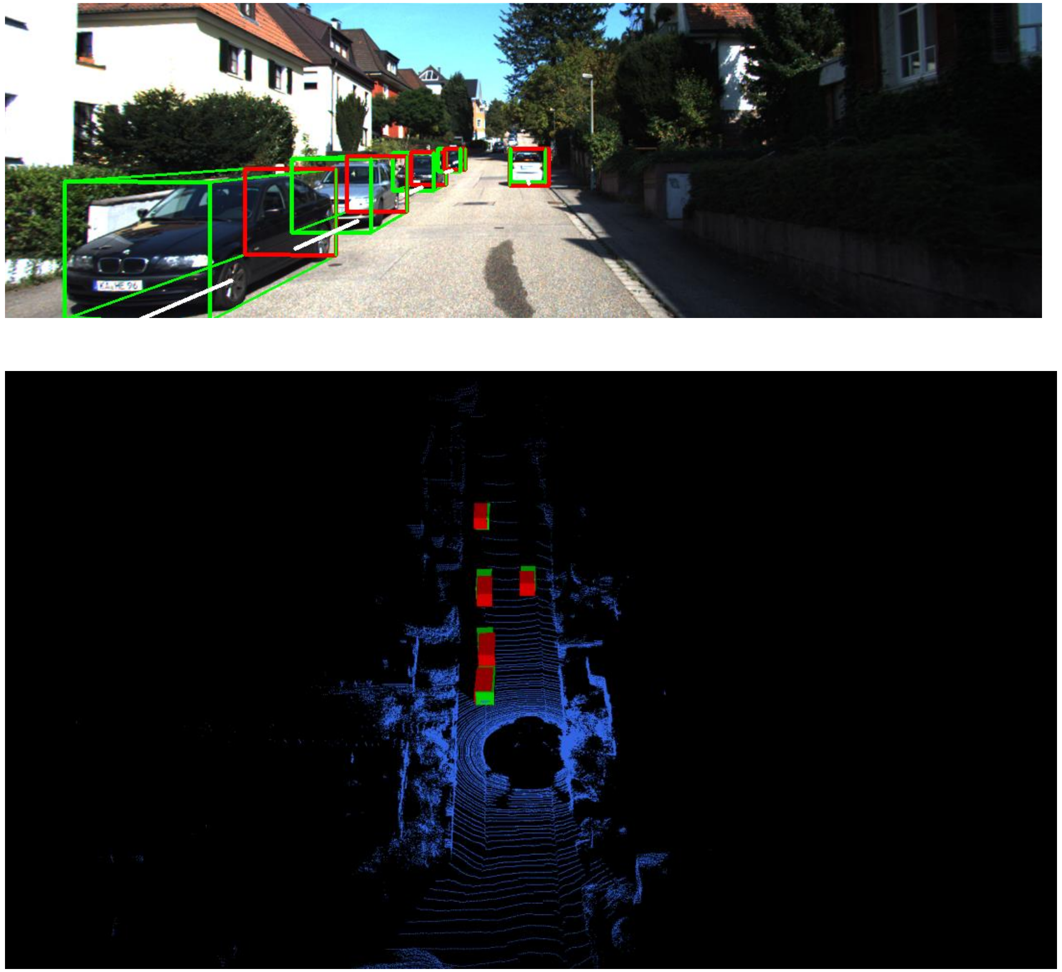


Mono3D Detections (courtesy of Chen et al.)

CAMP

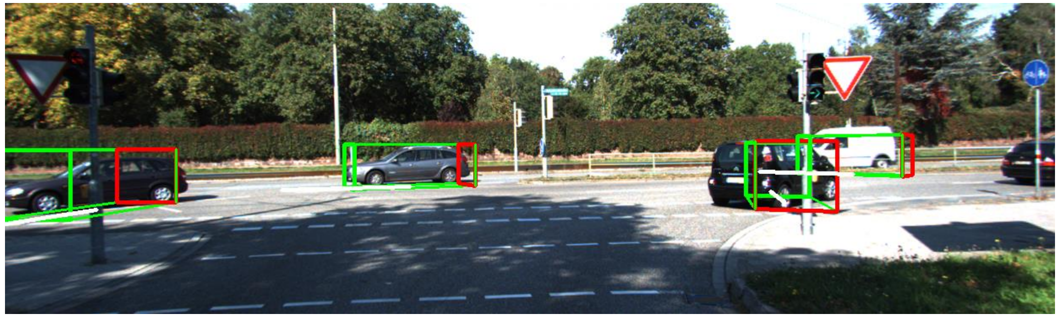
Qualitative Results

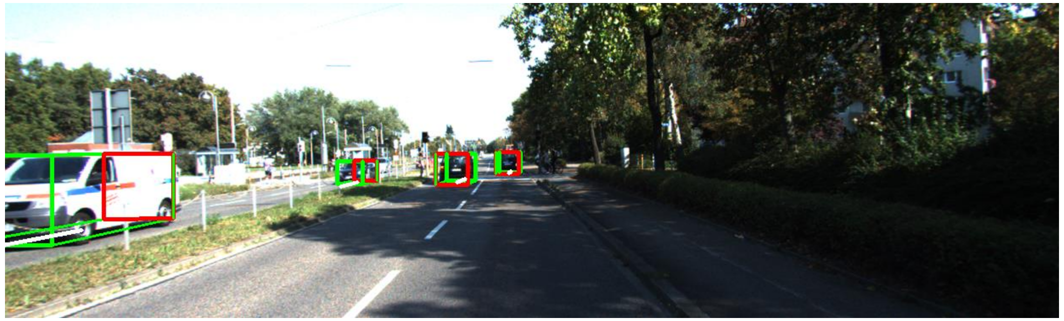


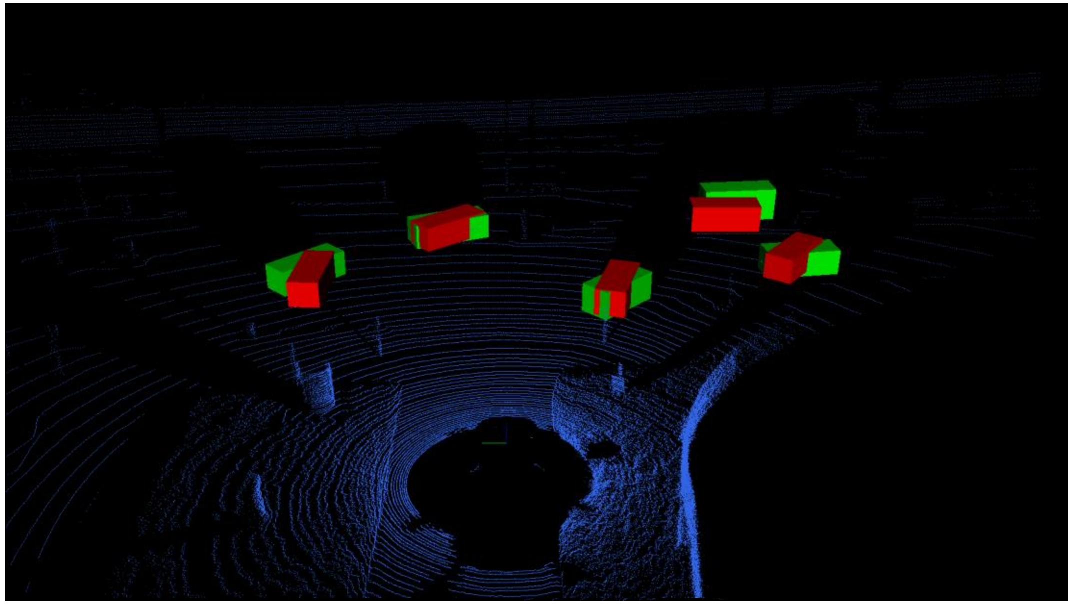


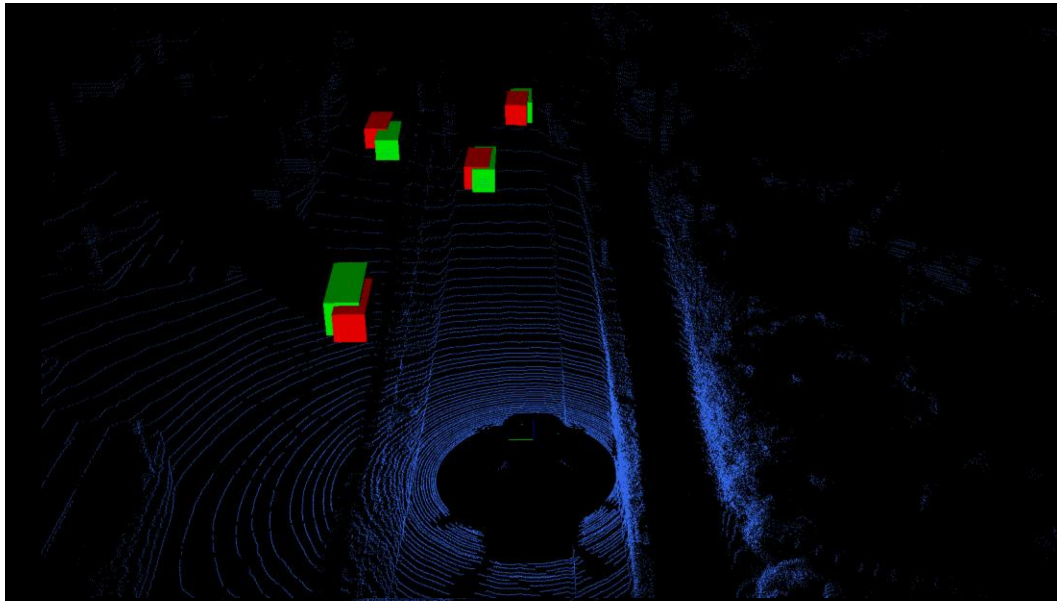
Lidar for visualization only
Green Boxes: Ground Truth
Red Boxes: Our predictions

Qualitative Results



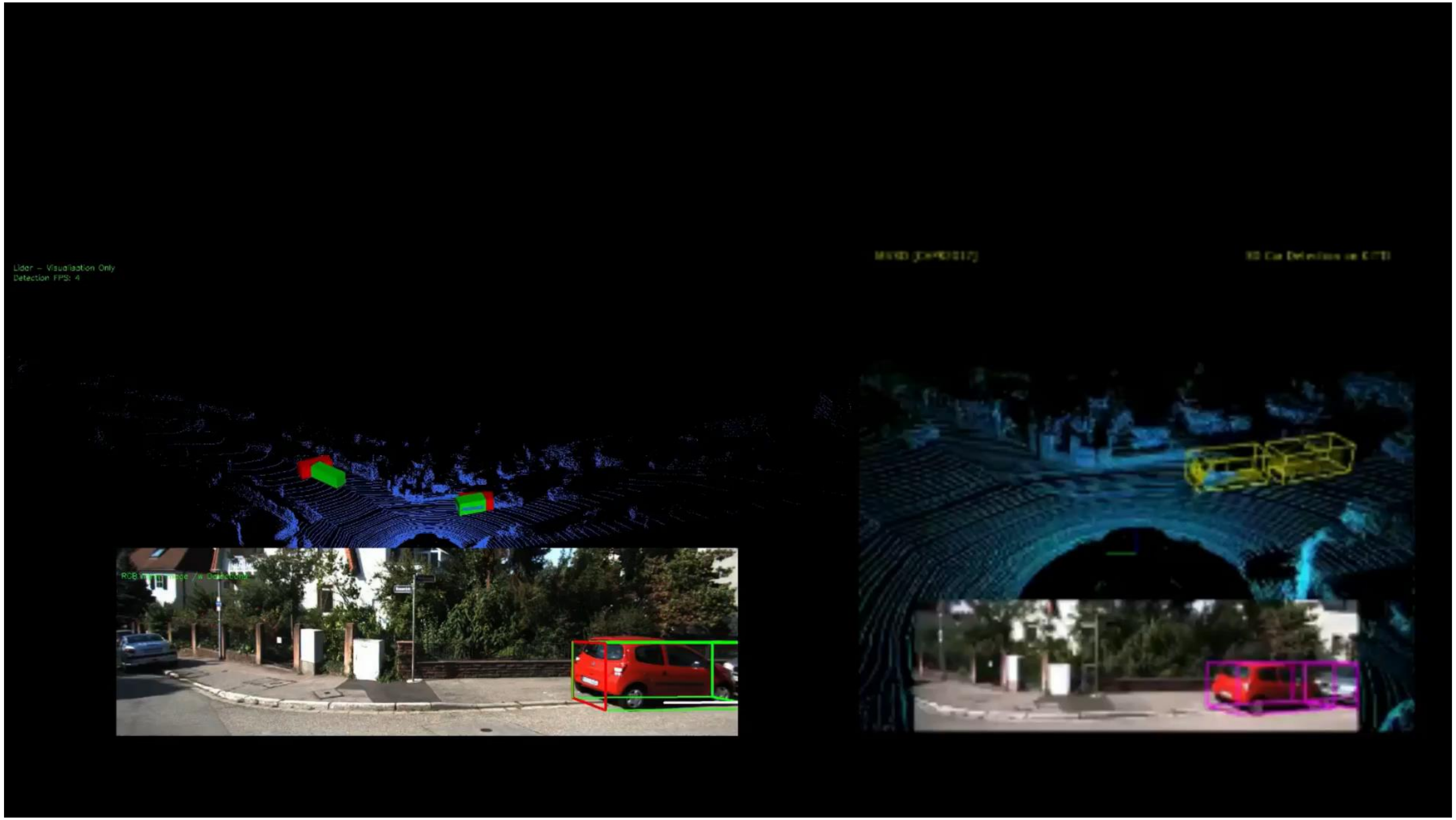






Lidar for visualization only
Green Boxes: Ground Truth
Red Boxes: Our predictions

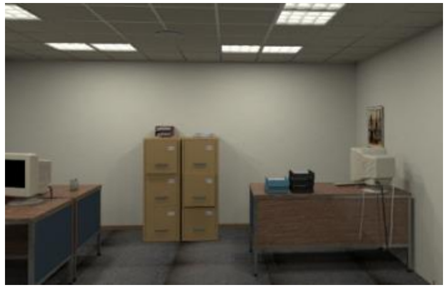
RGB vs. RGB+Lidar



Left: Ours (fully monocular)
Green Boxes: Ground Truth, Red Boxes: Predictions

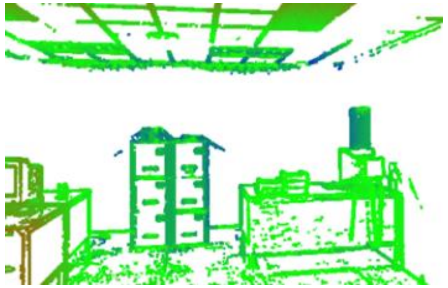
Right: MV3D [Chen17] (RGB+Lidar)

CNN-SLAM: monocular dense SLAM



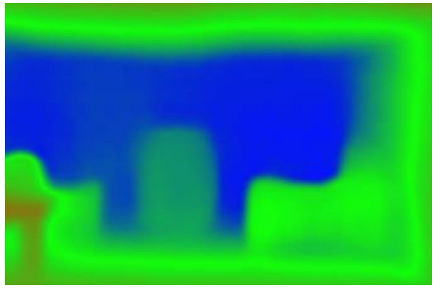
Monocular SLAM

Accurate on depth borders but sparse



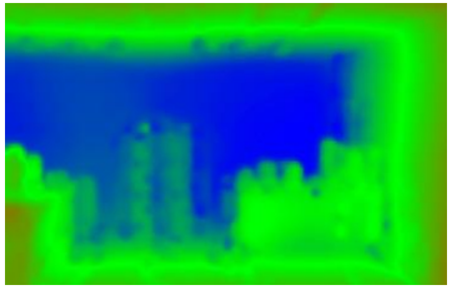
CNN Depth Prediction

Dense but imprecise along depth borders

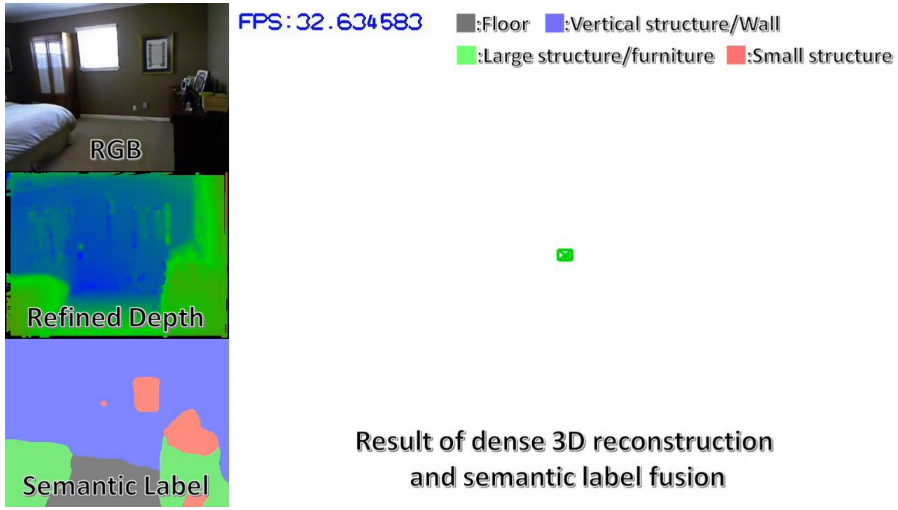


CNN-SLAM [Tateno17]

takes the best of both world by fusing monocular SLAM with depth prediction in real time



1. can learn the **absolute scale**
2. **dense maps**
3. can deal with **pure rotational motion**

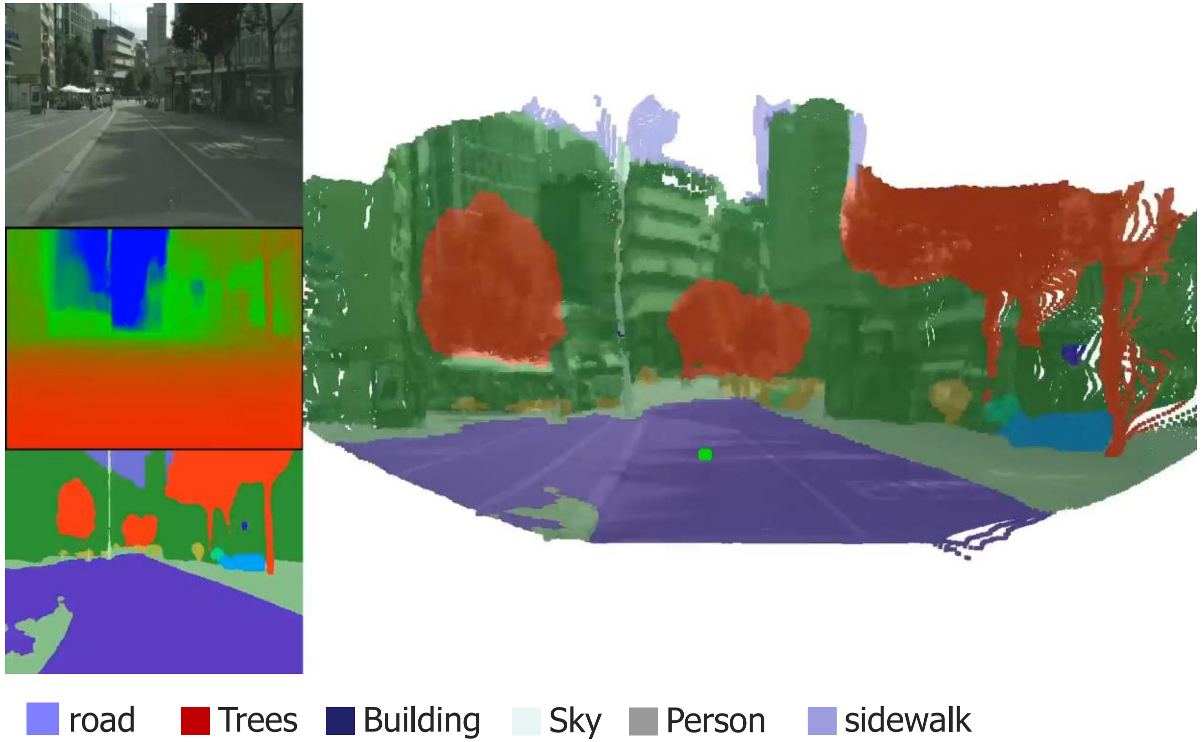


Result of dense 3D reconstruction and semantic label fusion

K. Tateno, F. Tombari, I. Laina, N. Navab, “CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction", CVPR 2017

CNN-SLAM for AD

KITTI dataset



Cityscapes dataset

What have we learned?

6D pose estimation can use deep learning to overcome the limitations of the sensing modality

3D learned descriptors generally report better performance in matching compared to hand-crafted

Monocular pose estimation can be carried out via deep learning (although not yet as accurately as with a depth sensor)

Open issues:

- Generalizability
- Geometric invariance
- Runtime/hardware limitations

Fusion of SLAM/real-time reconstruction with detection and pose estimation

New sensing technologies could be the next game changer

Main Credits (alphabetical)

- Wadim Kehl
- Ted Krubasik
- Fabian Manhardt
- Prof. Nassir Navab
- Dario Rethage
- Jürgen Sturm
- Dr. David J. Tan
- Keisuke Tateno
- Johanna Wald

Thanks to Google, Toyota and Pointu3D for supporting these research activities