# 3D Object Detection for Autonomous Driving

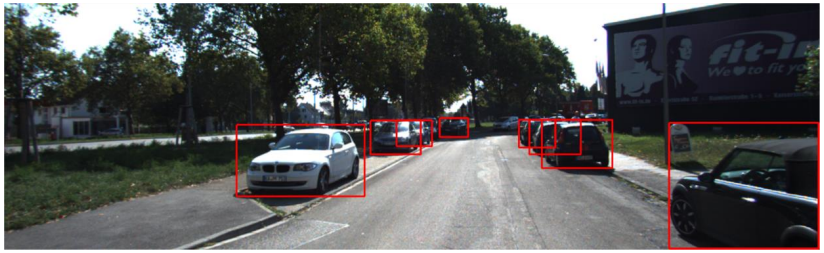
Xiaozhi Chen
Tsinghua University

Joint work with Kaustav Kunku, Yukun Zhu, Ziyu Zhang, Andrew Berneshawi, Huimin Ma, Sanja Fidler and Raquel Urtasun

# Goal: 3D Object Detection

Input Image

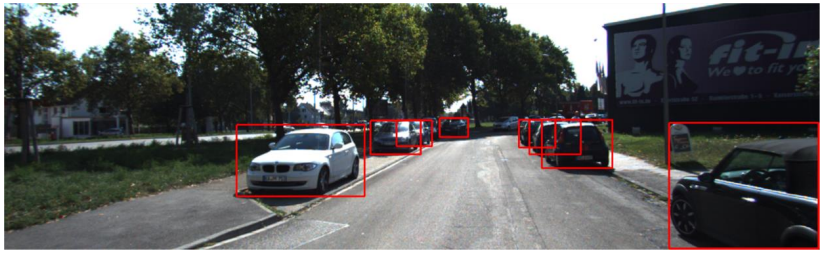
Where are the cars in the image?



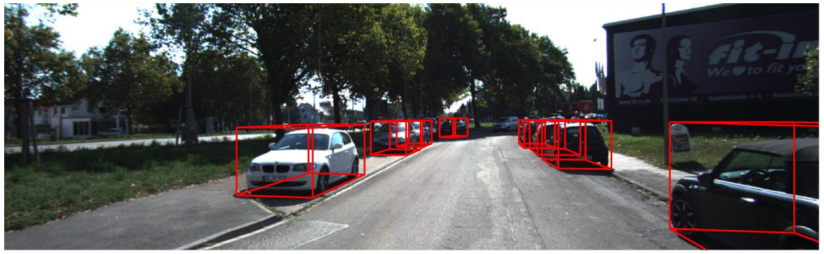
# Goal: 3D Object Detection

Input Image

Where are the cars in the image?

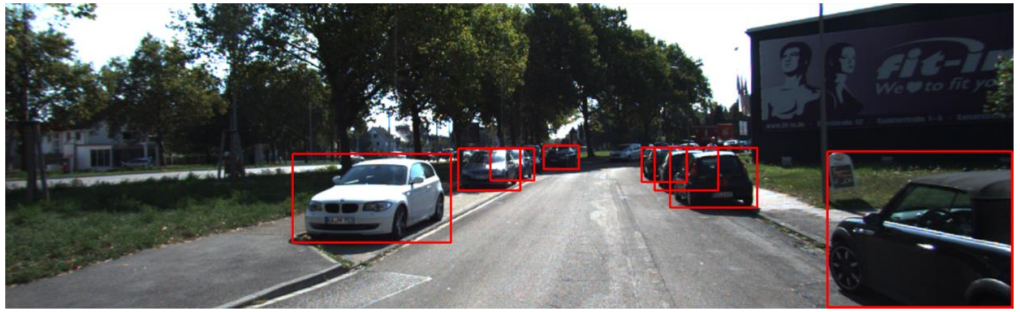


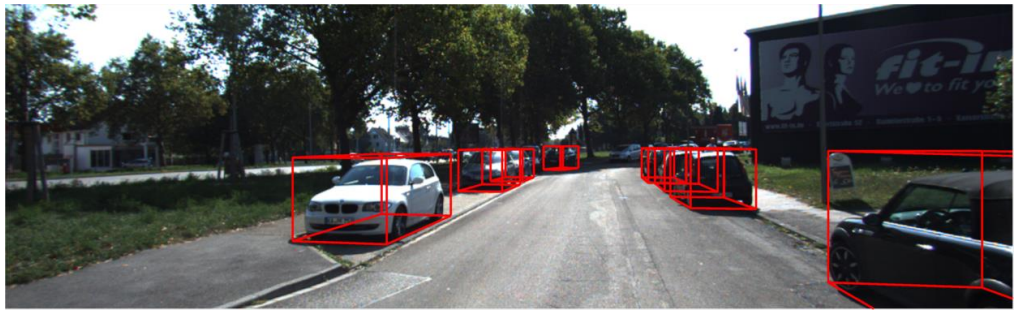
How far are the cars from the driver?



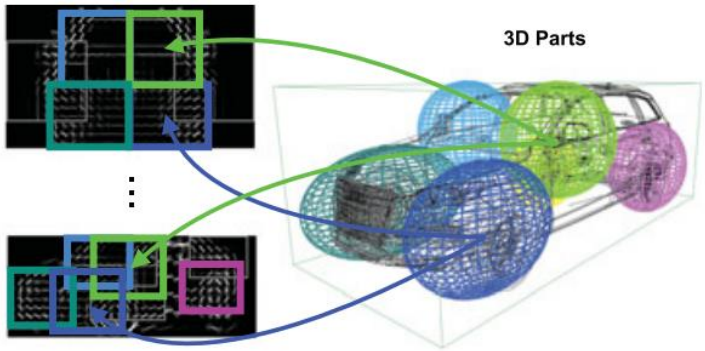
# Goal: 3D Object Detection

- ✓ 2D boxes
- ✓ 3D poses
- ✓ 3D location
- ✓ 3D boxes

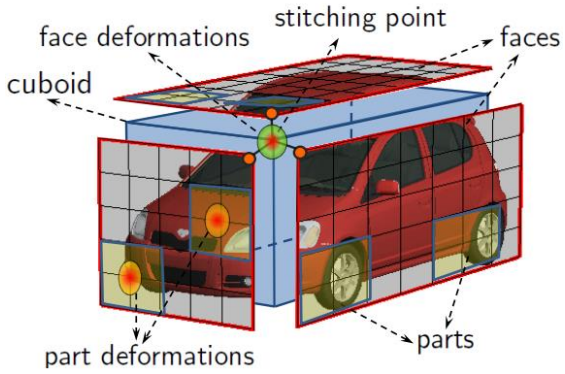




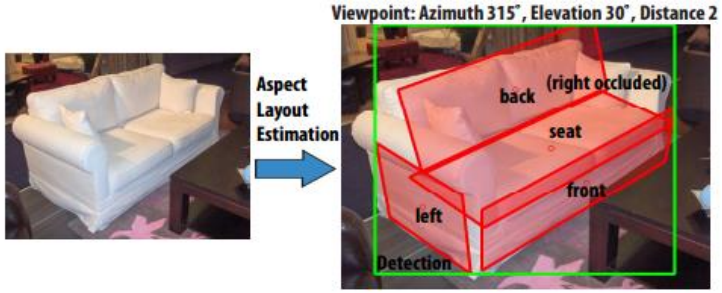
# Related Work: 3D Pose Estimation



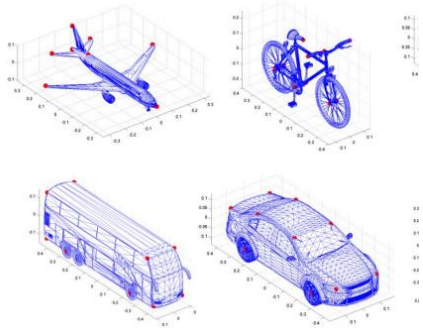

3D$^2$PM, Pepik et al. CVPR'12



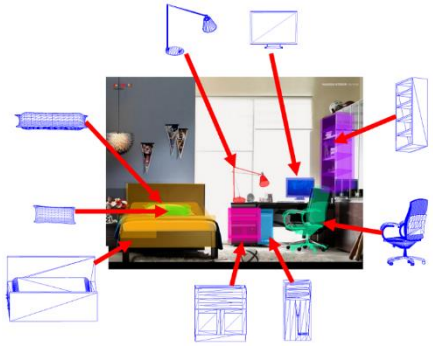

Fidler et al. NIPS'12




ALM, Xiang et al. CVPR'12



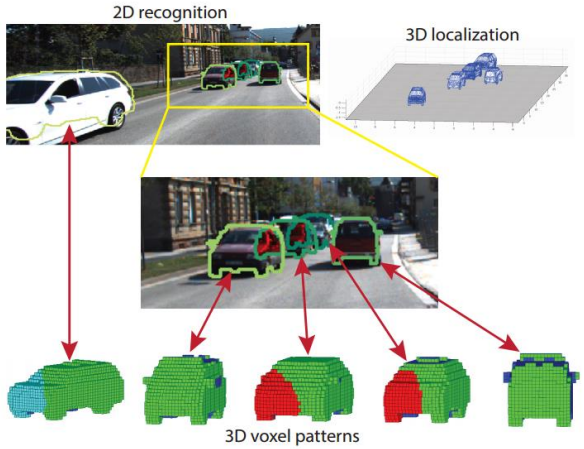
PASCAL3D+
Xiang et al. WACV'14



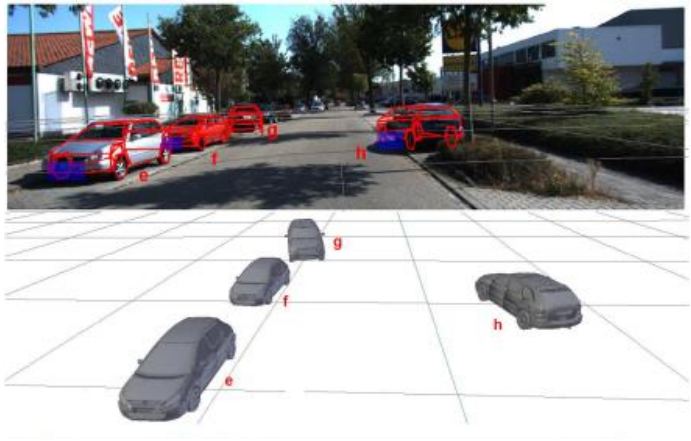
ObjectNet3D
Xiang et al. ECCV'16

- Thomas et al. CVPR'06
- Hoiem et al. CVPR'07
- Yan et al. ICCV'07
- Glasner et al. ICCV'11
- Hejrati et al. NIPS'12
- Etc.

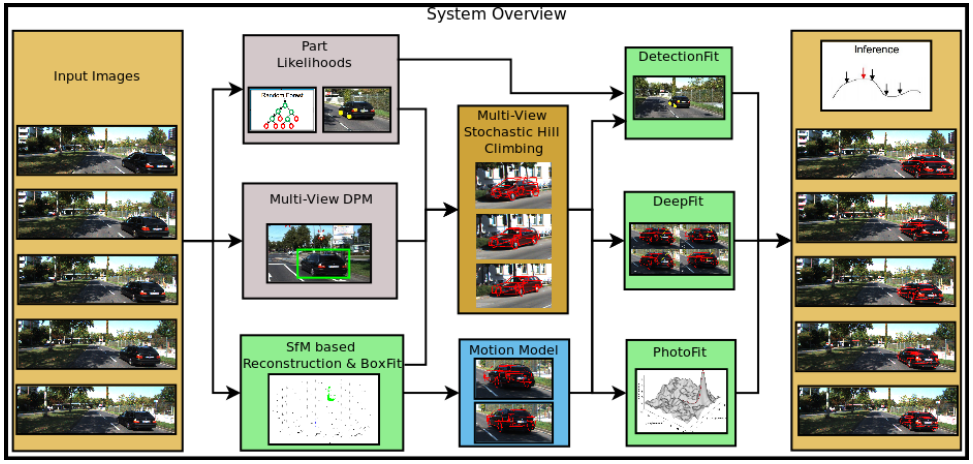
# Related Work: 3D Object Localization




Xiang et al. CVPR'15, arXiv'16

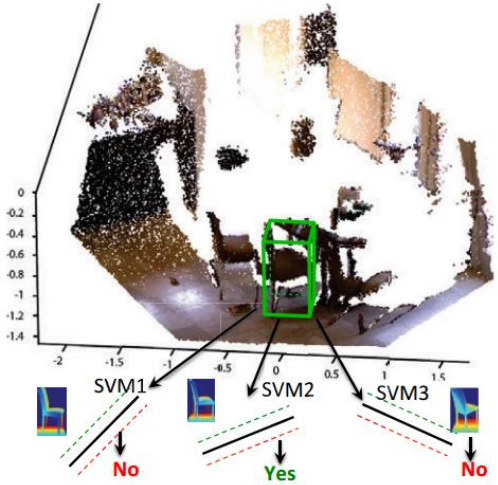


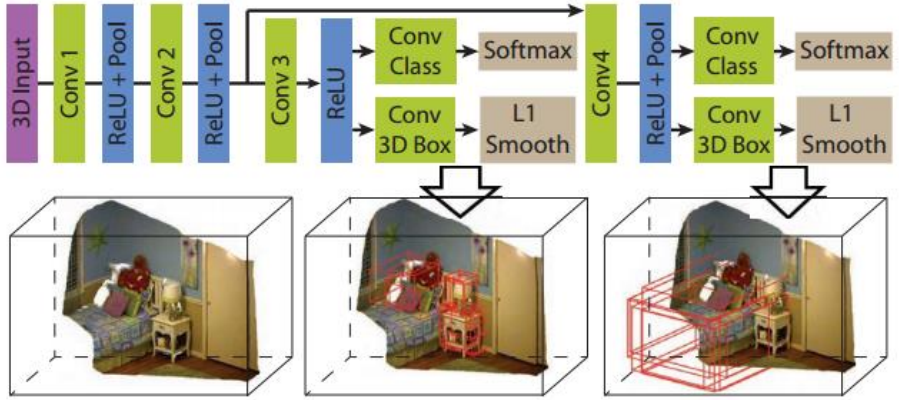

Zia et al. CVPR'14, IJCV'15



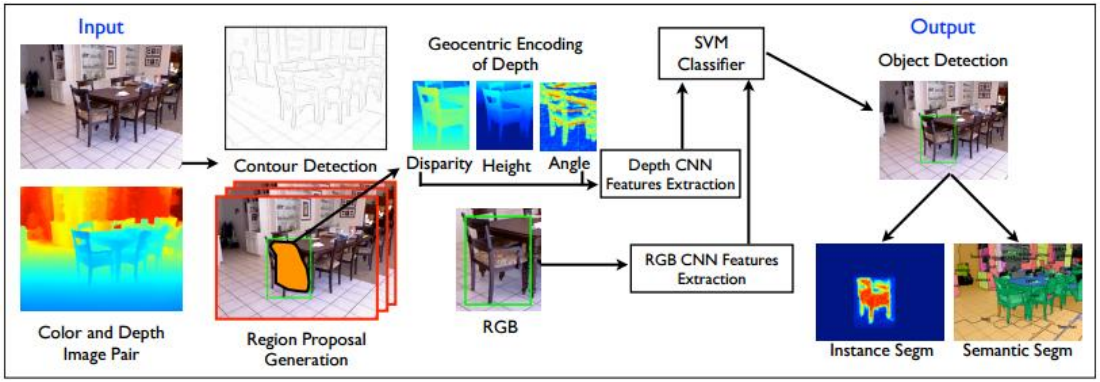

Chhaya et al. ICRA'16

# Related Work: 3D Object Detection (Indoor)



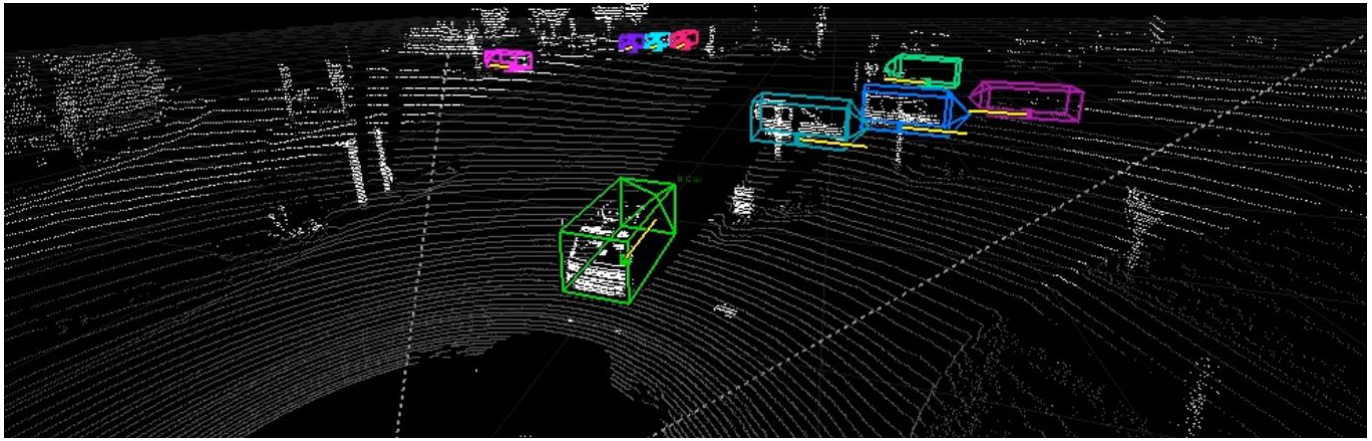




(Deep) Sliding Shape
Song & Xiao. ECCV'14, CVPR'16



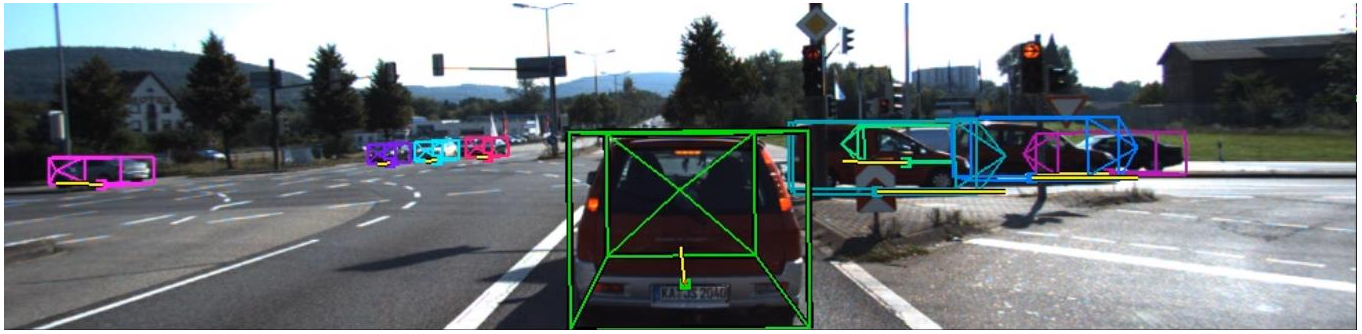

Depth R-CNN
Gupta et al. ECCV'14, CVPR'15

# What's the Best Sensor for Self-driving Cars?



## LIDAR

e.g., Google, Baidu

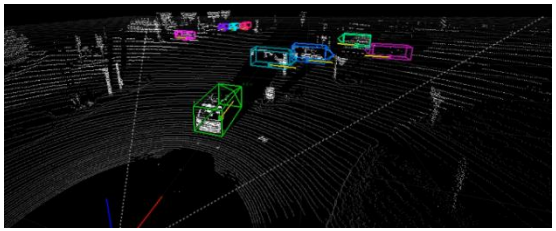


## Camera

e.g., Mobileye, Tesla

# Outline

LIDAR

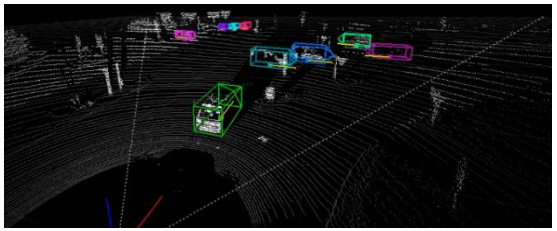


Stereo

Monocular

# Outline

LIDAR



Stereo

Monocular

**1** **3D Object Detection using Stereo Images**

NIPS'15

**2** **Monocular 3D Object Detection**

CVPR'16

# 3D Object Detection using Stereo Images

- Xiaozhi Chen*, Kaustav Kunku*, Yukun Zhu, Andrew Berneshawi, Huimin Ma, Sanja Fidler, Raquel Urtasun. ***3D Object Proposals for Accurate Object Class Detection***. NIPS 2015.

# Typical Object Detection Pipeline

Input Image

$$\begin{bmatrix} \mathbf{x} \end{bmatrix}$$

Feature Extraction

$$f_c(\mathbf{x})$$

Classification

## Candidate Box Selection

- Sliding Window
  - Exhaustive search across the entire image at multiple scales
- Object Proposal
  - Reduces the search space to focus on few regions, requires high recall

## Feature Extraction

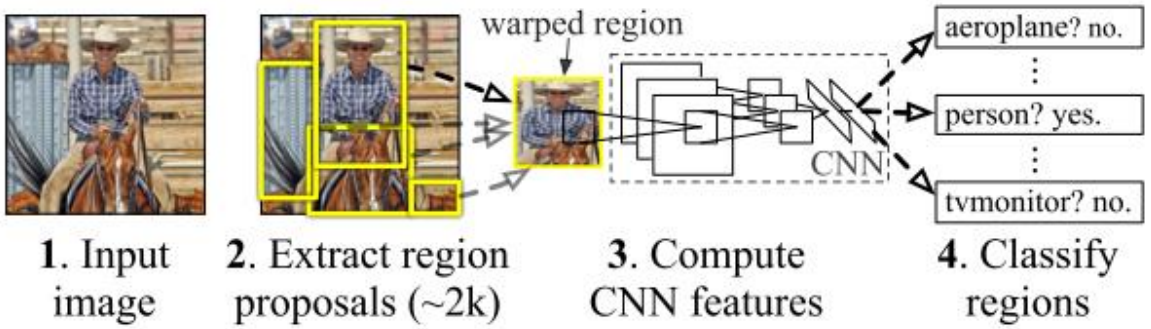
- HOG, CNN, etc.

## Classification

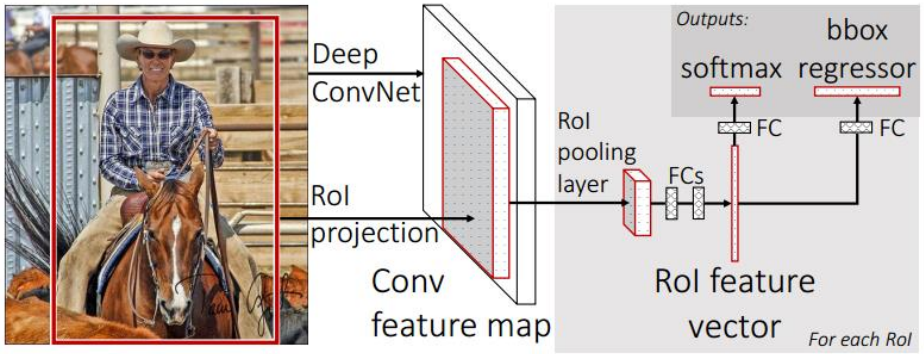
- Linear classifiers

# Typical Object Detection Pipeline

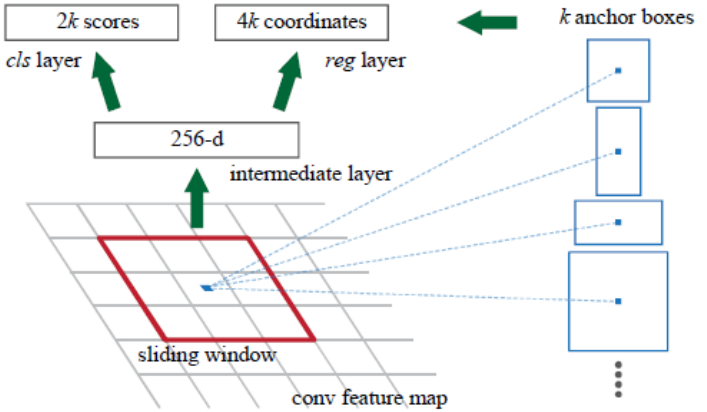
## R-CNN [CVPR'14]




## Fast R-CNN [ICCV'15]



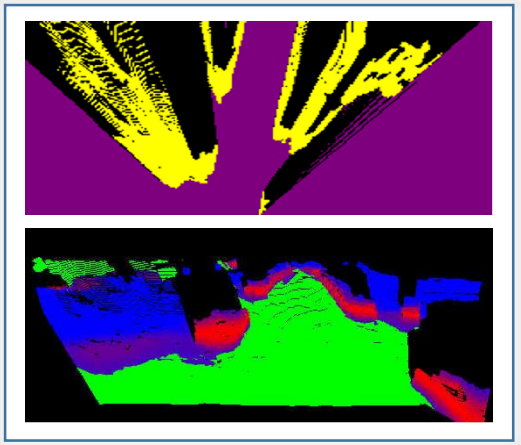

## Faster R-CNN [NIPS'15]



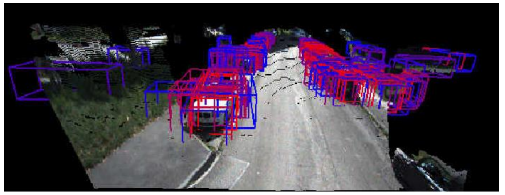
# 3DOP: Overview

## 3D Proposal Generation

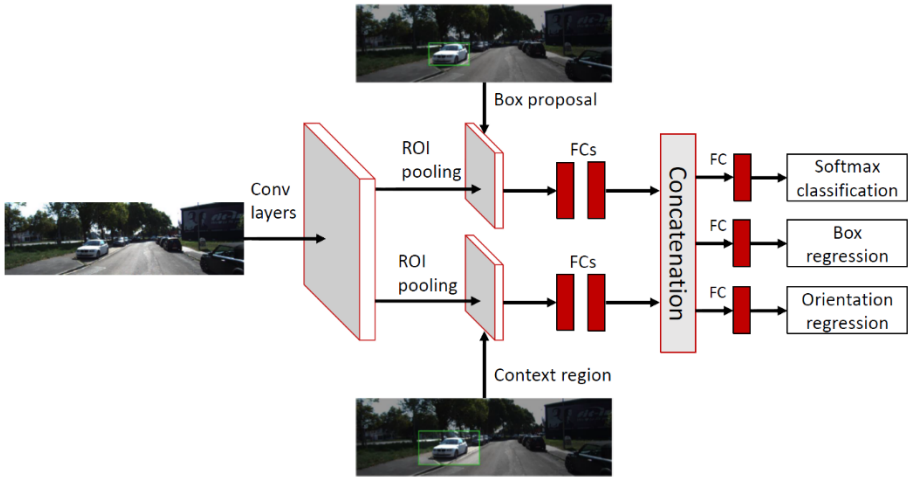
Stereo images



3D proposals



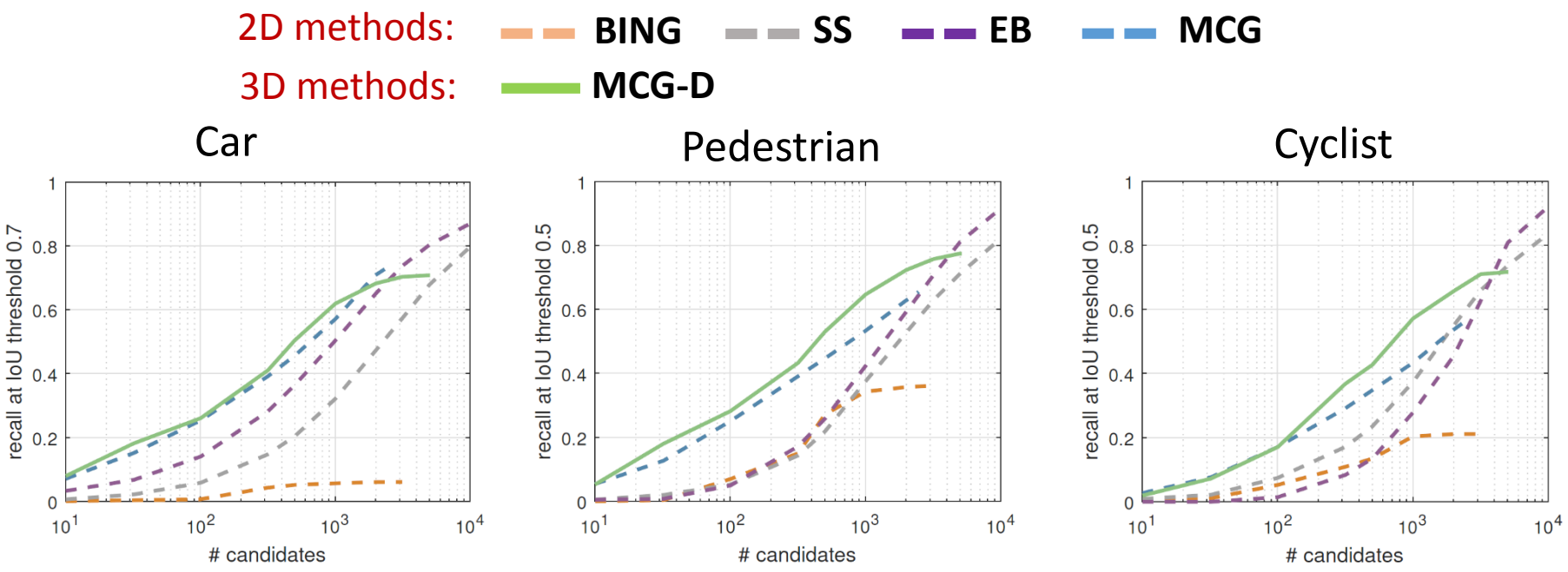
## CNN Scoring




# KITTI: Autonomous Driving Dataset

- **KITTI** (Geiger et al., CVPR'12)
  - **Categories**: Car, Pedestrian, Cyclist
  - **Data:** LIDAR point cloud, stereo images
  - **Annotations**: 2D/3D bounding boxes, occlusion/truncation labels

# 2D Proposals Recall on KITTI



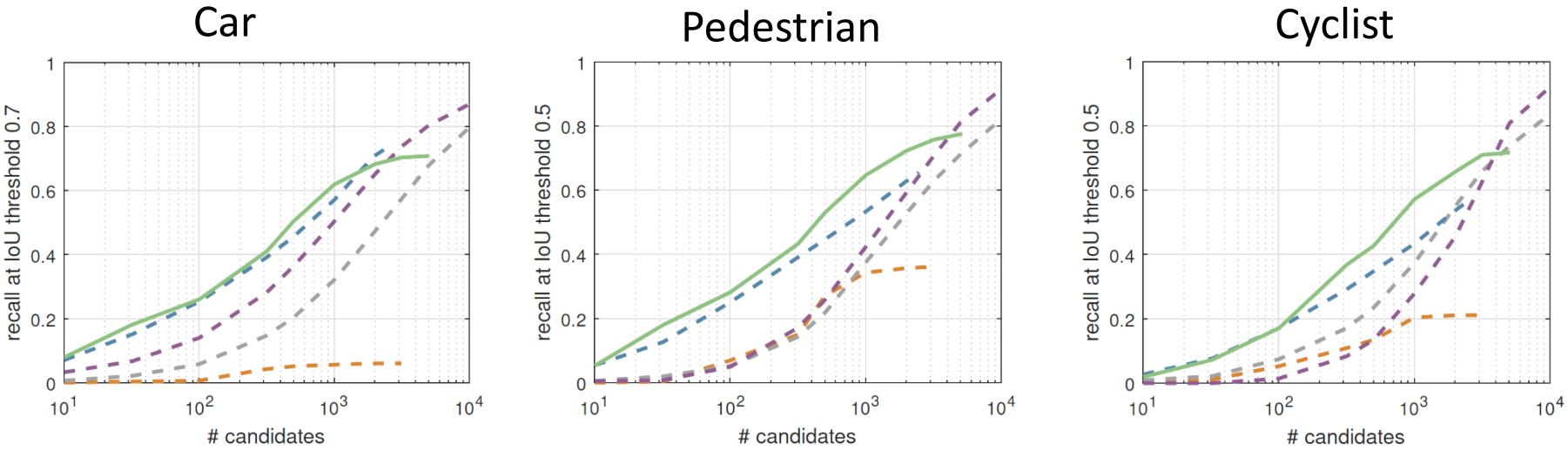

- **PASCAL**: recall (1K Prop.) > **95%**
- **KITTI**: recall (1K Prop.) < **75%!!!**

- **[BING]** BING: Binarized normed gradients for objectness estimation at 300fps. CVPR'14. Cheng et al.
- **[SS]** Segmentation as selective search for object recognition. ICCV'11. Sande et al.
- **[EB]** Edge boxes: locating object proposals from edges. ECCV'14. Zitnick et al.
- **[MCG]** Multiscale combinatorial grouping. CVPR'14. Pablo et al.
- **[MCG-D]** Learning rich features from RGB-D images for object detection and segmentation. ECCV'14. Gupta et al.

# 2D Proposals Recall on KITTI

2D methods: BING SS EB MCG

3D methods: MCG-D




- **PASCAL**: recall (1K Prop.) > **95%**
- **KITTI**: recall (1K Prop.) < **75%!!!**

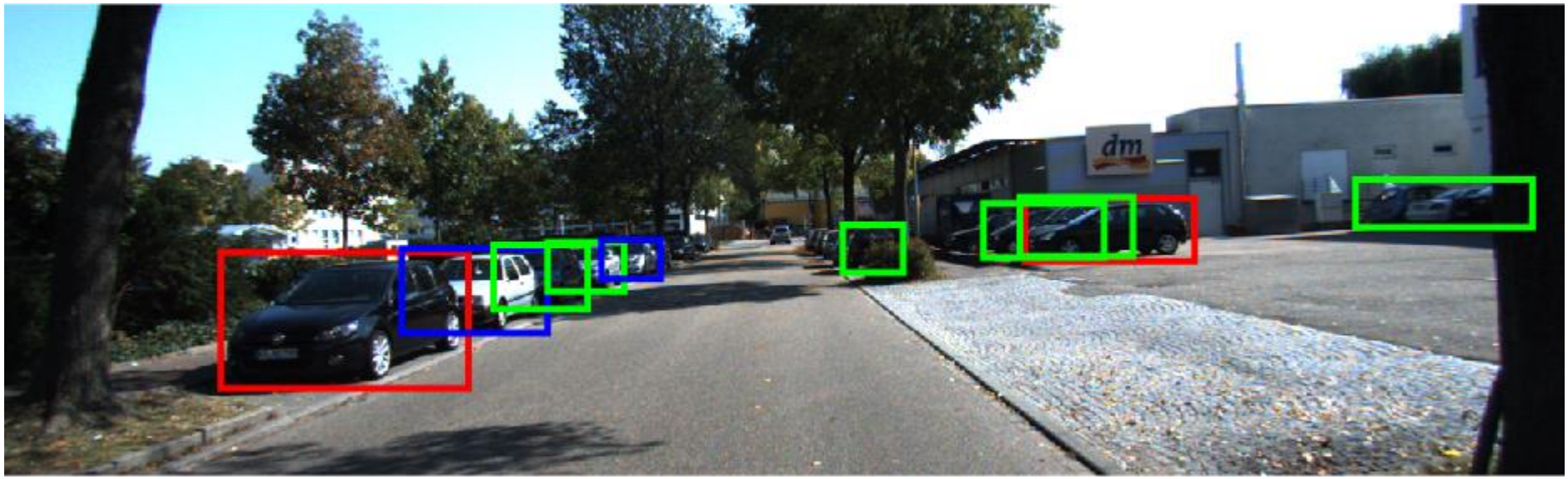
Why?

- **[BING]** BING: Binarized normed gradients for objectness estimation at 300fps. CVPR'14. Cheng et al.
- **[SS]** Segmentation as selective search for object recognition. ICCV'11. Sande et al.
- **[EB]** Edge boxes: locating object proposals from edges. ECCV'14. Zitnick et al.
- **[MCG]** Multiscale combinatorial grouping. CVPR'14. Pablo et al.
- **[MCG-D]** Learning rich features from RGB-D images for object detection and segmentation. ECCV'14. Gupta et al.

# Challenges on KITTI

- **Strict localization metric**
  - 0.7 IoU overlap threshold for Cars
- **Clutter scene**
- **Heavy occlusion**
- **Small objects, high resolution (370x1240)**

Easy Moderate Hard

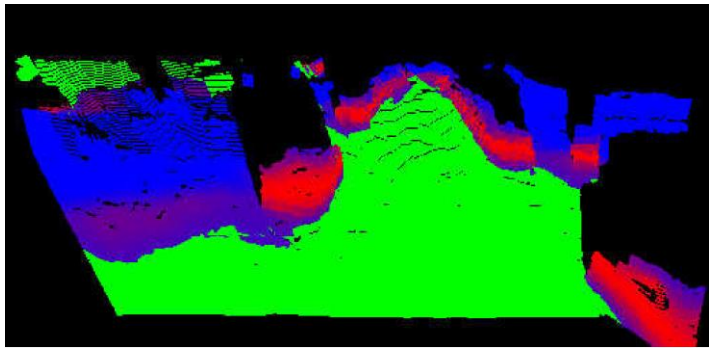


# 3DOP: Feature Computation

Left image

Right image

Bird's eye view



Height prior

Yellow: Occupancy
Purple: Free space

Green: Ground plane
Red → Blue: Increasing height prior

# Parameterization

- $\mathbf{x}$: Point cloud of input stereo image pair

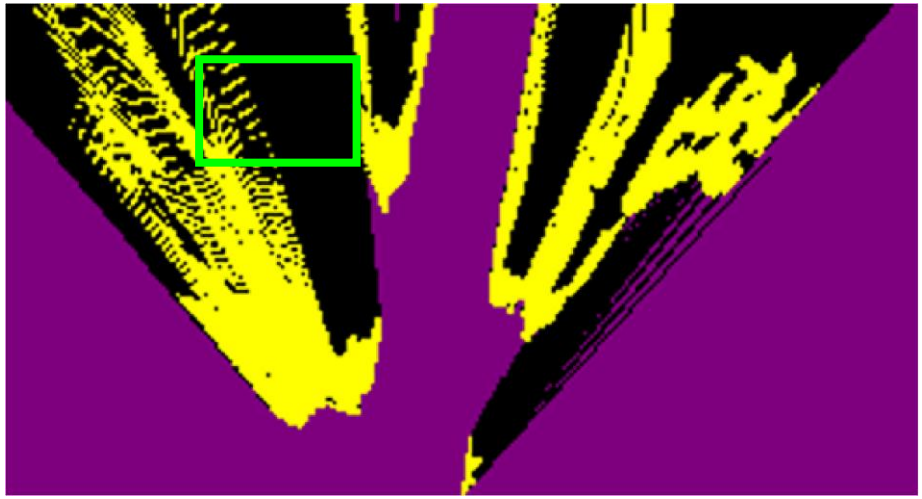
# Parameterization

- $\mathbf{x}$: Point cloud of input stereo image pair
- $\mathbf{y} = (x, y, z, \theta, c, t)$: 3D bounding box candidate

# Parameterization

- $\mathbf{x}$: Point cloud of input stereo image pair
- $\mathbf{y} = (x, y, z, \theta, c, t)$: 3D bounding box candidate

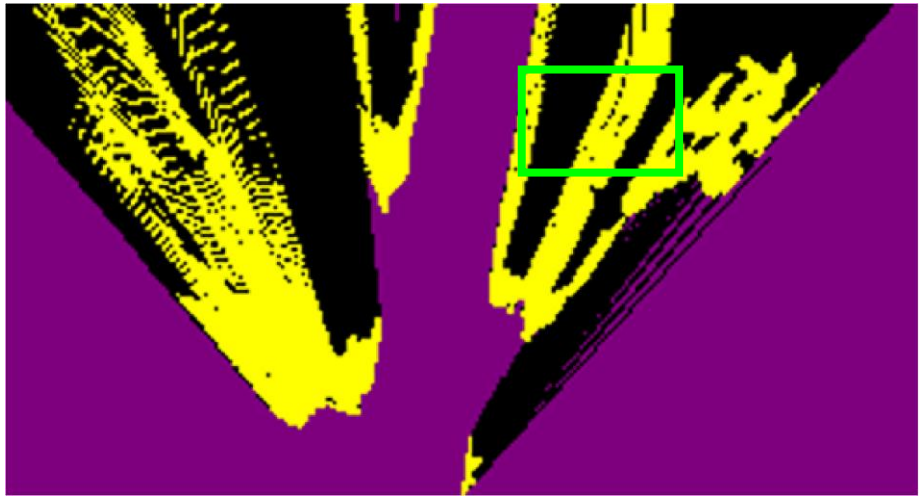
$(x, y, z)$: center of 3D box



# Parameterization

- $\mathbf{x}$: Point cloud of input stereo image pair
- $\mathbf{y} = (x, y, z, \theta, c, t)$: 3D bounding box candidate

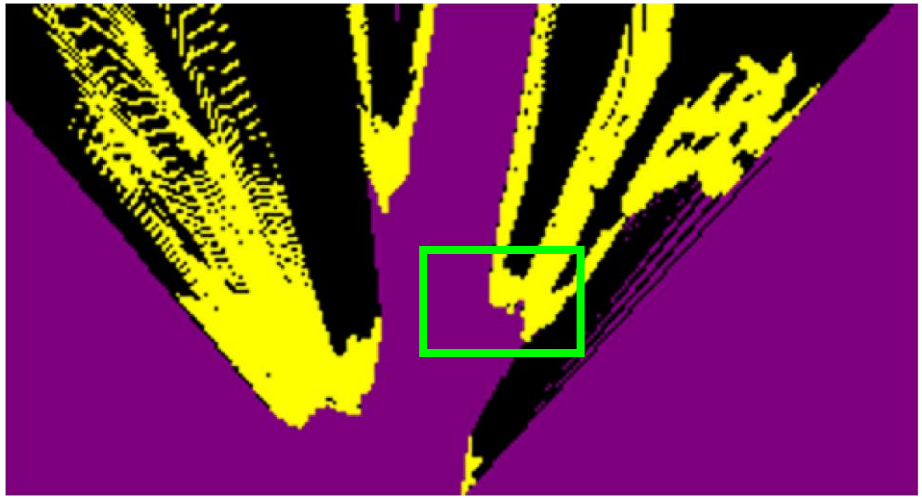
$(x, y, z)$: center of 3D box



# Parameterization

- $\mathbf{x}$: Point cloud of input stereo image pair
- $\mathbf{y} = (x, y, z, \theta, c, t)$: 3D bounding box candidate

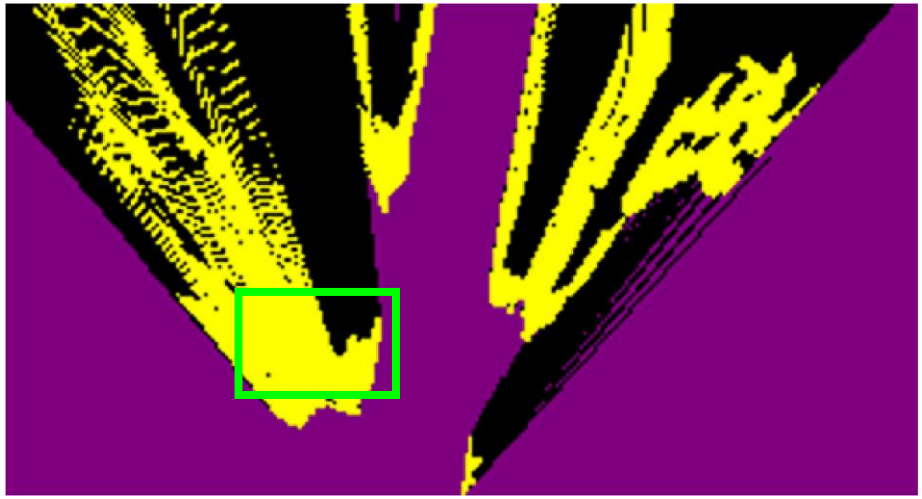
$(x, y, z)$: center of 3D box



# Parameterization

- $\mathbf{x}$: Point cloud of input stereo image pair
- $\mathbf{y} = (x, y, z, \theta, c, t)$: 3D bounding box candidate

$(x, y, z)$: center of 3D box

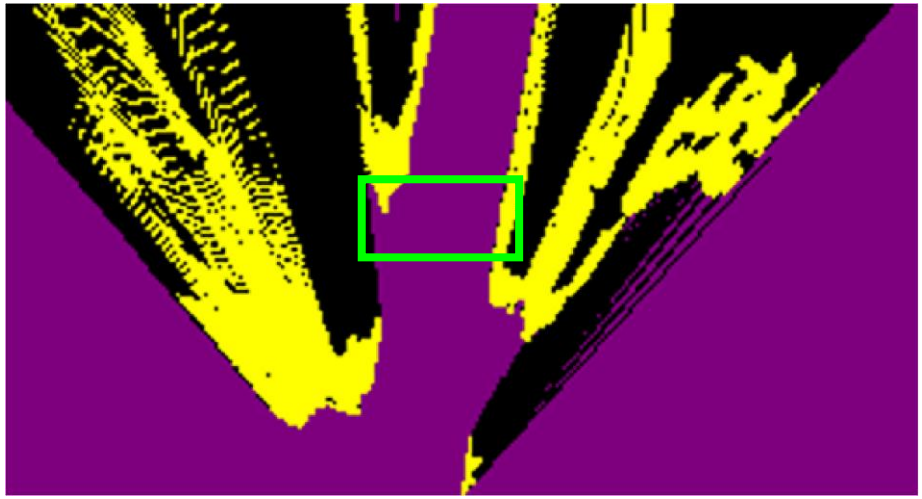


# Parameterization

- $\mathbf{x}$: Point cloud of input stereo image pair
- $\mathbf{y} = (x, y, z, \theta, c, t)$: 3D bounding box candidate

$(x, y, z)$: center of 3D box

$\theta$: azimuth angle

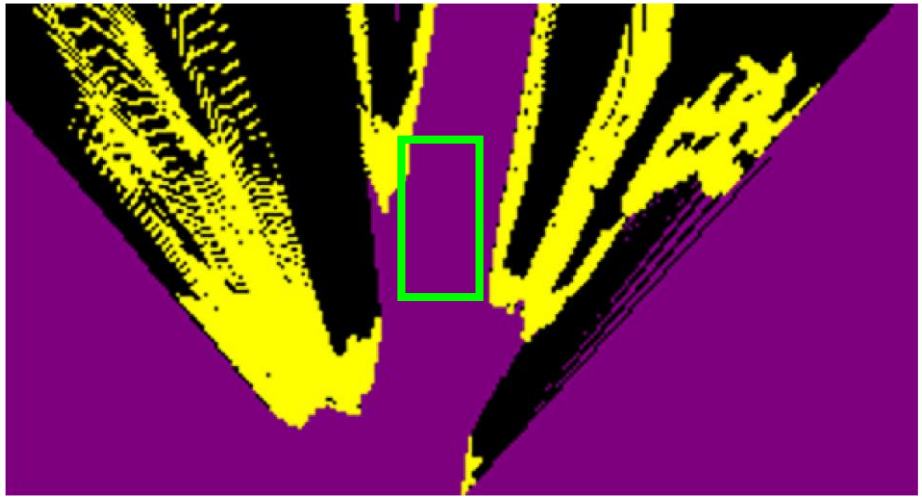


# Parameterization

- $\mathbf{x}$: Point cloud of input stereo image pair
- $\mathbf{y} = (x, y, z, \theta, c, t)$: 3D bounding box candidate

$(x, y, z)$: center of 3D box

$\theta$: azimuth angle



# Parameterization

- $\mathbf{x}$: Point cloud of input stereo image pair
- $\mathbf{y} = (x, y, z, \theta, c, t)$: 3D bounding box candidate

$(x, y, z)$: center of 3D box

$\theta$: azimuth angle

$c$: object category ∈{Car, Pedestrian, Cyclist}

# Parameterization

- $\mathbf{x}$: Point cloud of input stereo image pair
- $\mathbf{y} = (x, y, z, \theta, c, t)$: 3D bounding box candidate

$(x, y, z)$: center of 3D box

$\theta$: azimuth angle

$c$: object category $\in$ {Car, Pedestrian, Cyclist}

$t \in \{1, \dots, T_c\}$: category-specific template

# Parameterization

- $\mathbf{x}$: Point cloud of input stereo image pair
- $\mathbf{y} = (x, y, z, \theta, c, t)$: 3D bounding box candidate

$(x, y, z)$: center of 3D box

$\theta$: azimuth angle

$c$: object category ∈{Car, Pedestrian, Cyclist}

$t \in \{1, \dots, T_c\}$: category-specific template

$$E(\mathbf{x}, \mathbf{y}) = E_{pc}(\mathbf{x}, \mathbf{y}) + E_{fs}(\mathbf{x}, \mathbf{y}) + E_{ht}(\mathbf{x}, \mathbf{y}) + E_{ht-contr}(\mathbf{x}, \mathbf{y})$$

# Energy Terms

- $\mathbf{x}$: Point cloud of input stereo image pair
- $\mathbf{y} = (x, y, z, \theta, c, t)$: 3D bounding box candidate

$$E(\mathbf{x}, \mathbf{y}) = E_{pc}(\mathbf{x}, \mathbf{y}) + E_{fs}(\mathbf{x}, \mathbf{y}) + E_{ht}(\mathbf{x}, \mathbf{y}) + E_{ht-contr}(\mathbf{x}, \mathbf{y})$$

Point cloud occupancy

# Energy Terms

- $\mathbf{x}$: Point cloud of input stereo image pair
- $\mathbf{y} = (x, y, z, \theta, c, t)$: 3D bounding box candidate

$$E(\mathbf{x}, \mathbf{y}) = E_{pc}(\mathbf{x}, \mathbf{y}) + E_{fs}(\mathbf{x}, \mathbf{y}) + E_{ht}(\mathbf{x}, \mathbf{y}) + E_{ht-contr}(\mathbf{x}, \mathbf{y})$$

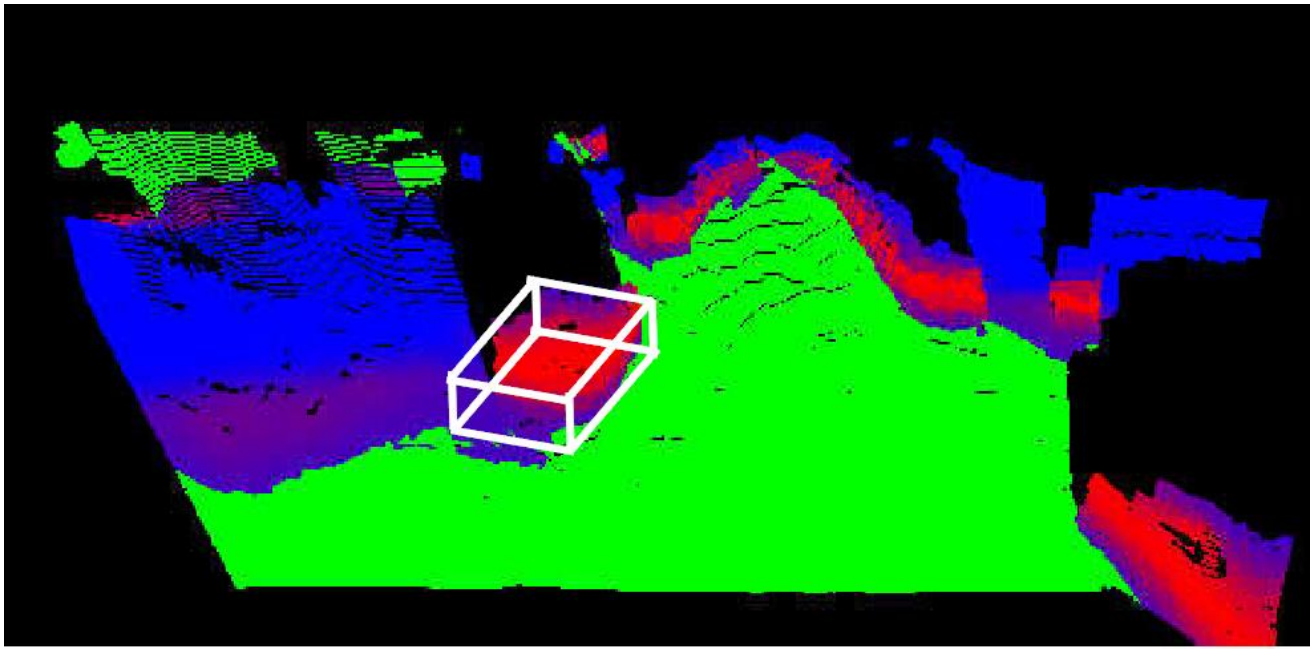
Free space

# Energy Terms

- $\mathbf{x}$: Point cloud of input stereo image pair
- $\mathbf{y} = (x, y, z, \theta, c, t)$: 3D bounding box candidate

$$E(\mathbf{x}, \mathbf{y}) = E_{pc}(\mathbf{x}, \mathbf{y}) + E_{fs}(\mathbf{x}, \mathbf{y}) + E_{ht}(\mathbf{x}, \mathbf{y}) + E_{ht-contr}(\mathbf{x}, \mathbf{y})$$

Height prior

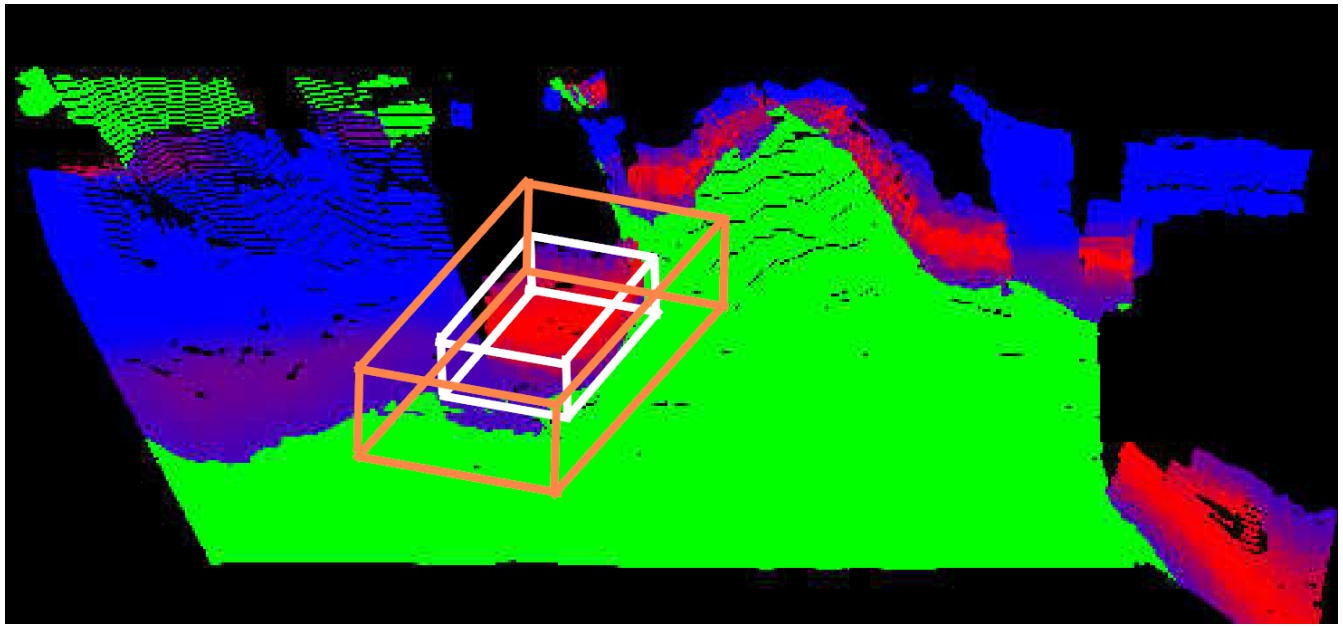


# Energy Terms

- $\mathbf{x}$: Point cloud of input stereo image pair
- $\mathbf{y} = (x, y, z, \theta, c, t)$: 3D bounding box candidate

$$E(\mathbf{x}, \mathbf{y}) = E_{pc}(\mathbf{x}, \mathbf{y}) + E_{fs}(\mathbf{x}, \mathbf{y}) + E_{ht}(\mathbf{x}, \mathbf{y}) + E_{ht-contr}(\mathbf{x}, \mathbf{y})$$

Height contrast



# Inference

$$\mathbf{y}^* = \underset{\mathbf{y}}{\operatorname{argmin}}\, E_{pc}(\mathbf{x},\mathbf{y}) + E_{fs}(\mathbf{x},\mathbf{y}) + E_{ht}(\mathbf{x},\mathbf{y}) + E_{ht-contr}(\mathbf{x},\mathbf{y})$$

- **Voxelization**
  - Voxel Dim. = 0.2m
- **Candidate sampling**
  - Sample cuboids closed the road plane
- **Feature computation**
  - 3D integral images
- **Proposals ranking**
  - Sort all candidates according to $E(\mathbf{x},\mathbf{y})$, NMS

Inference time: ~1.2s in a single thread

# Inference

## Speed Comparison

| Method | Time (sec.) |
|---|---|
| BING [CVPR'14] | 0.01 |
| Selective Search [ICCV'11] | 15 |
| EdgeBoxes [ECCV'14] | 1.5 |
| MCG [CVPR'14] | 100 |
| MCG-D [ECCV'14] | 160 |
| **Ours** | **1.2** |

# Learning

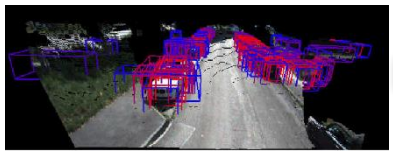
## Structured SVM:

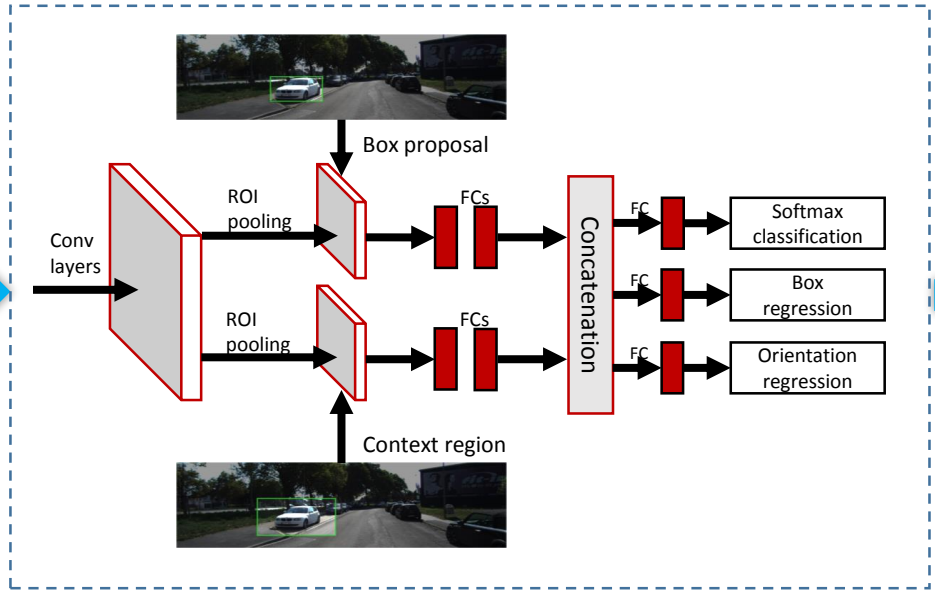
$$\min_{\mathbf{w}\in\mathbb{R}^D} \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i$$

$$\text{s.t.:} \quad \mathbf{w}^T(\phi(\mathbf{x}^{(i)},\mathbf{y}) - \phi(\mathbf{x}^{(i)},\mathbf{y}^{(i)})) \geq \Delta(\mathbf{y}^{(i)},\mathbf{y}) - \xi_i, \quad \forall \mathbf{y} \setminus \mathbf{y}^{(i)}$$

$$\Delta(\mathbf{y}^{(i)},\mathbf{y}) = 1 - \text{3D IoU}$$

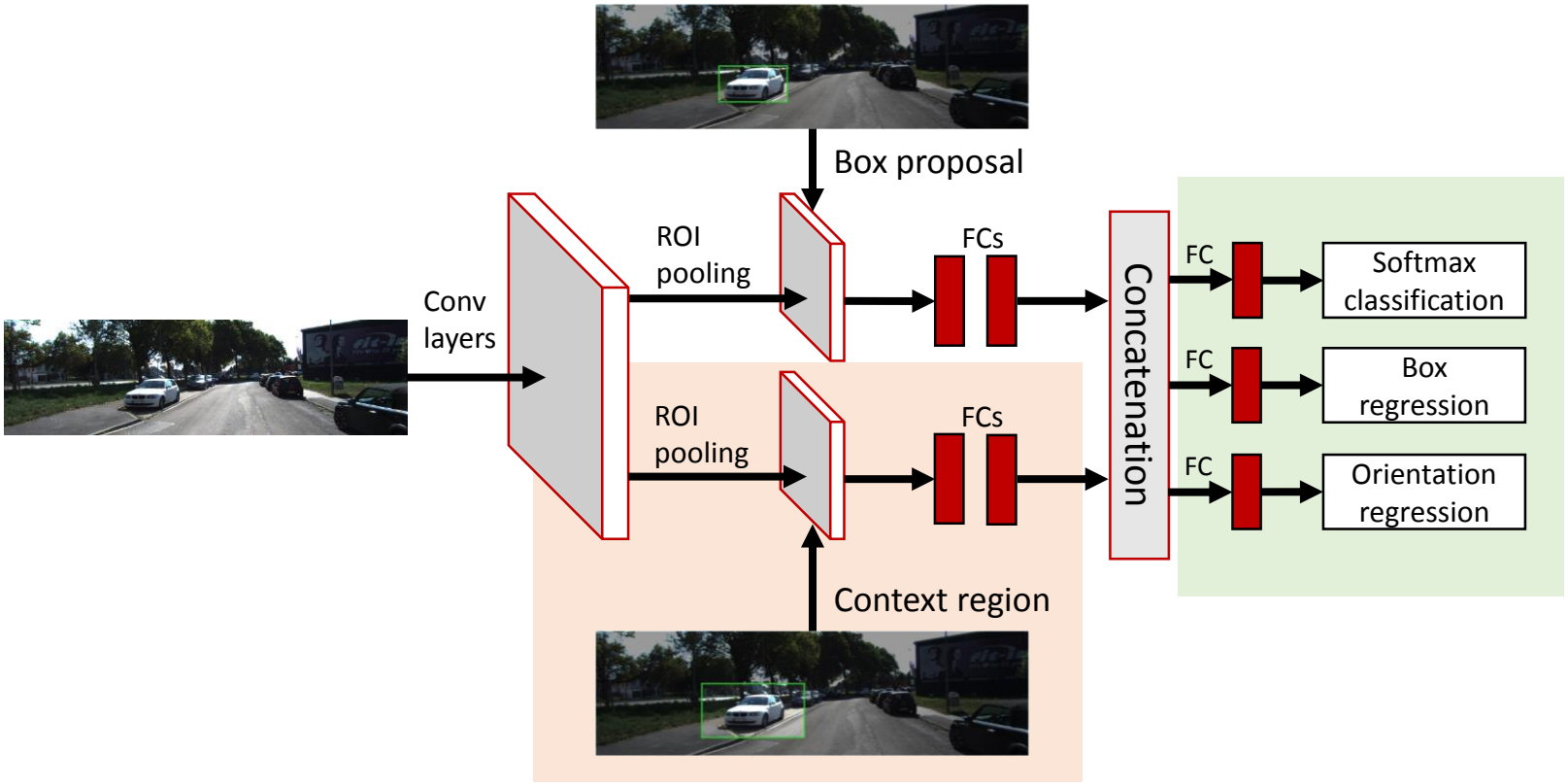
# 3D Object Detection Network



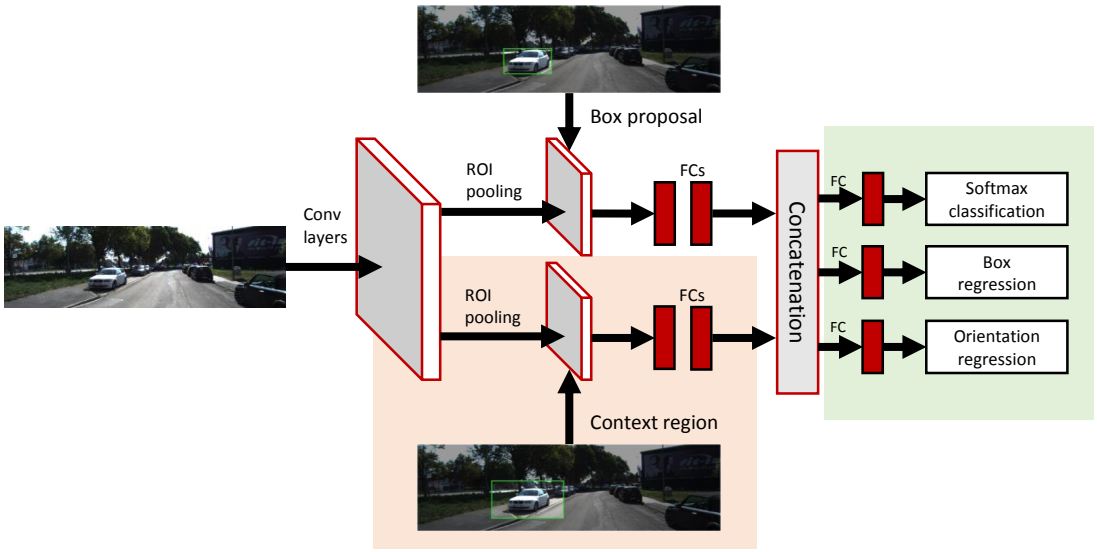



# 3D Object Detection Network

- Incorporating context information
- Joint object detection and orientation estimation



# 3D Object Detection Network




- Regression targets:

$$\mathbf{t}_{\text{2D}} = (t_x, t_y, t_w, t_h)$$
$$\mathbf{t}_{\text{3D}} = (t_X, t_Y, t_Z, t_L, t_W, t_H)$$
$$\mathbf{t}_{\text{ort}} = t_\theta$$

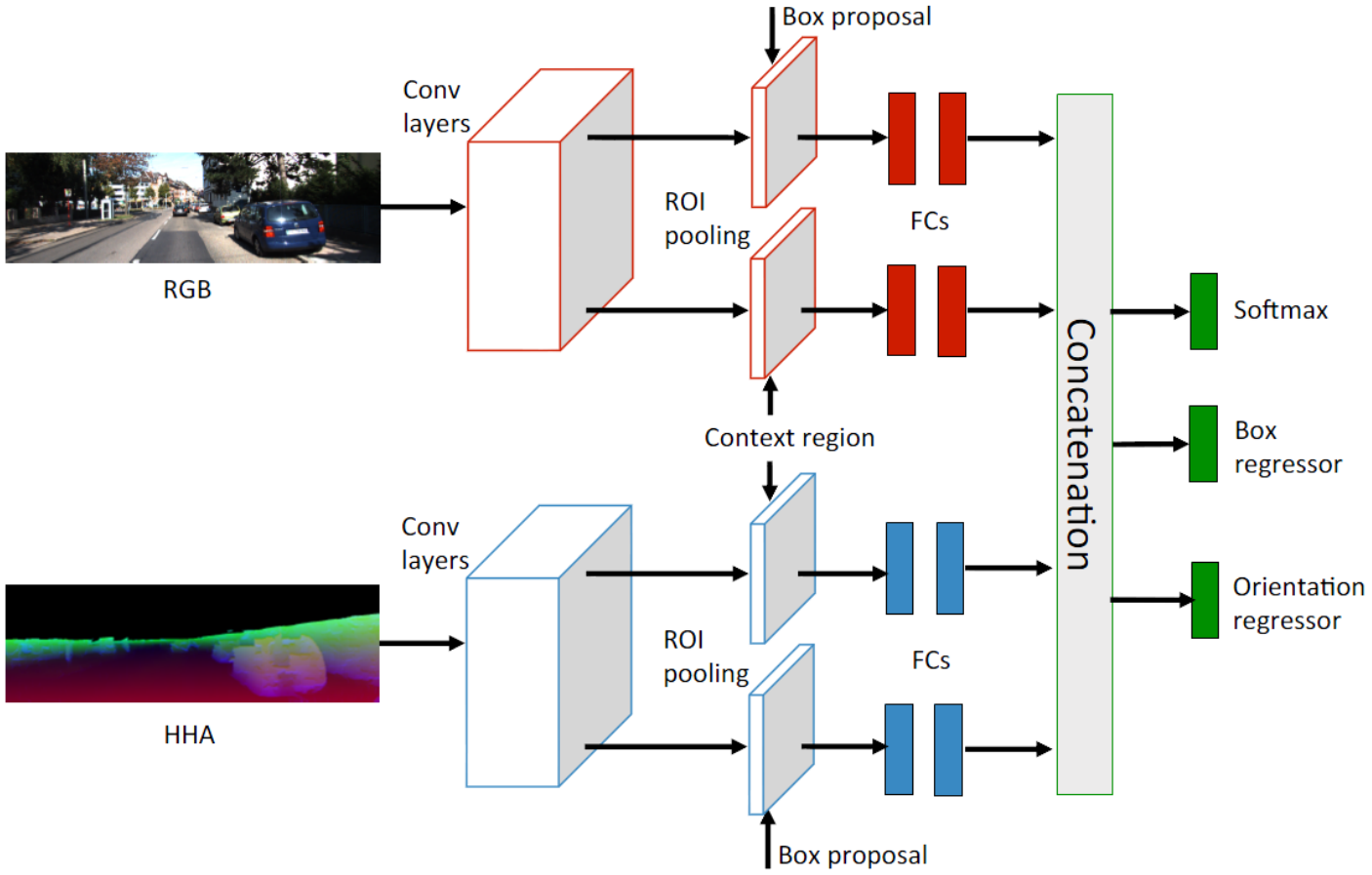
- Multi-task loss:

$$L = L_{\text{classification}} + L_{\text{box}} + L_{\text{orientation}}$$

Softmax loss  Smooth $l_1$ loss

# 3D Object Detection Network

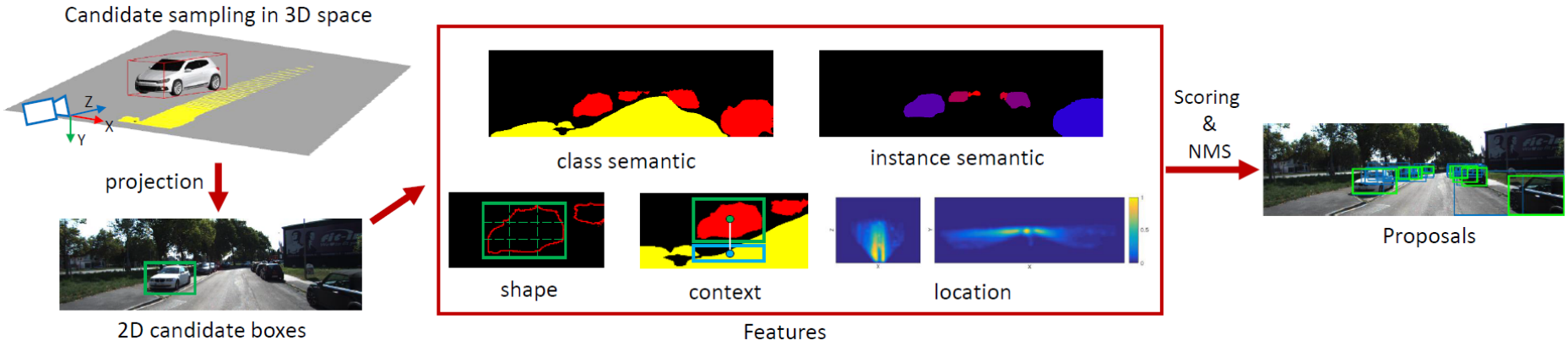
- Incorporating context information
- Joint object detection and orientation estimation
- Multi-stream feature learning



# Monocular 3D Object Detection (Mono3D)

- Xiaozhi Chen, Kaustav Kunku, Ziyu Zhang, Huimin Ma, Sanja Fidler, Raquel Urtasun. ***Monocular 3D Object Detection for Autonomous Driving***. CVPR 2016.

# Mono3D: Overview




## ☐ Stereo

- 3D Sampling
- Road Estimation from 3D
- Point Cloud Features
- Exhaustive Search
- Structured SVM

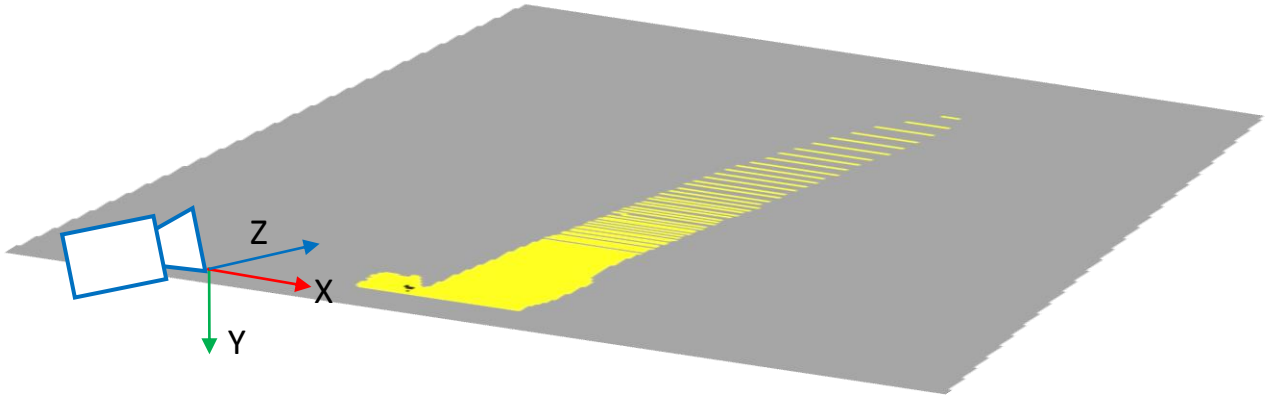
## ☐ Monocular

- 3D Sampling
- Road Estimation from 2D
- Semantic Features
- Exhaustive Search
- Structured SVM

# 3D Candidates Sampling

Road semantic

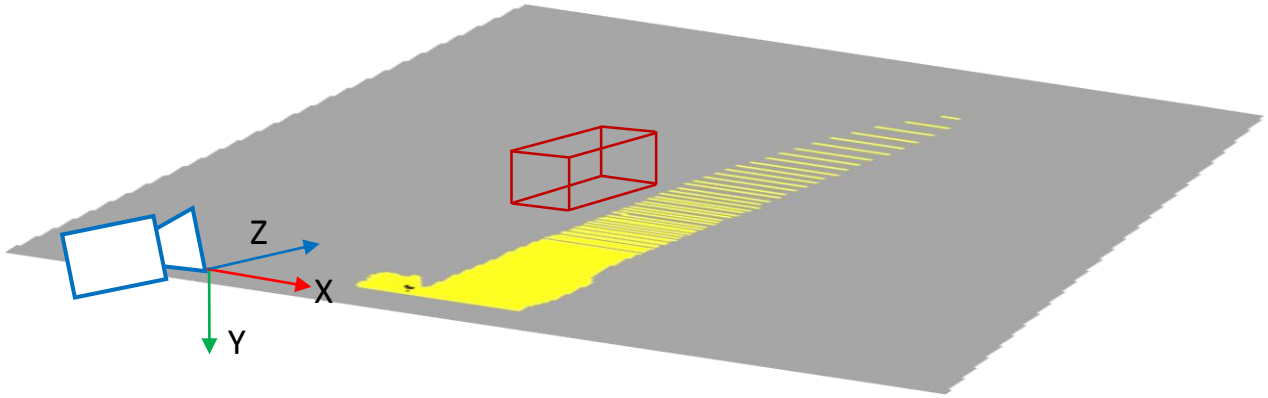
Back-projection
(Ground Prior)



# 3D Candidates Sampling

Road semantic

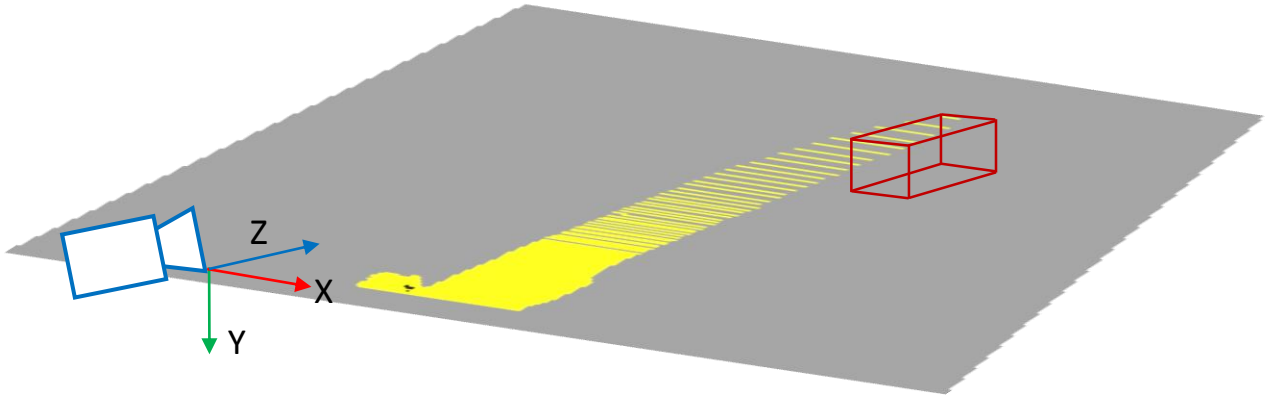
Back-projection
(Ground Prior)



# 3D Candidates Sampling

Road semantic

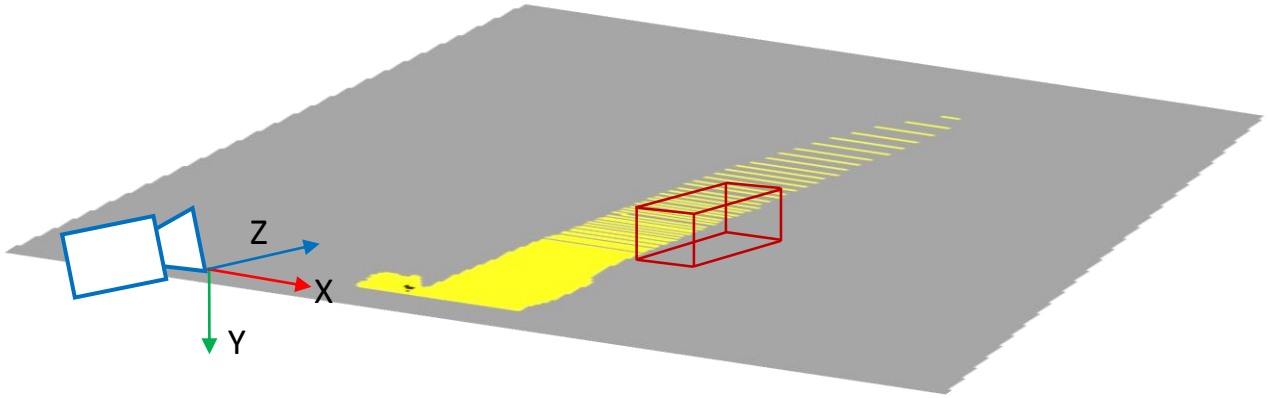
Back-projection
(Ground Prior)



# 3D Candidates Sampling

Road semantic

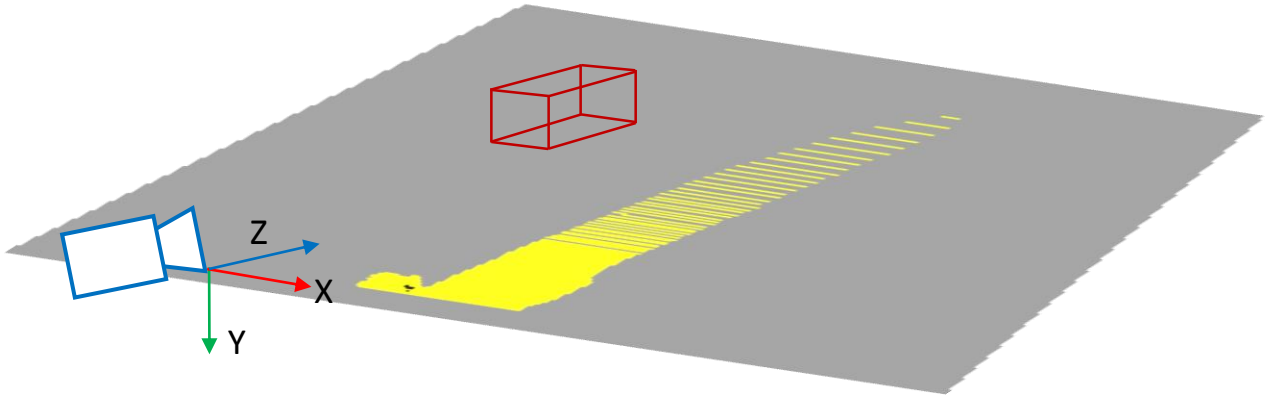
Back-projection
(Ground Prior)



# 3D Candidates Sampling

Road semantic

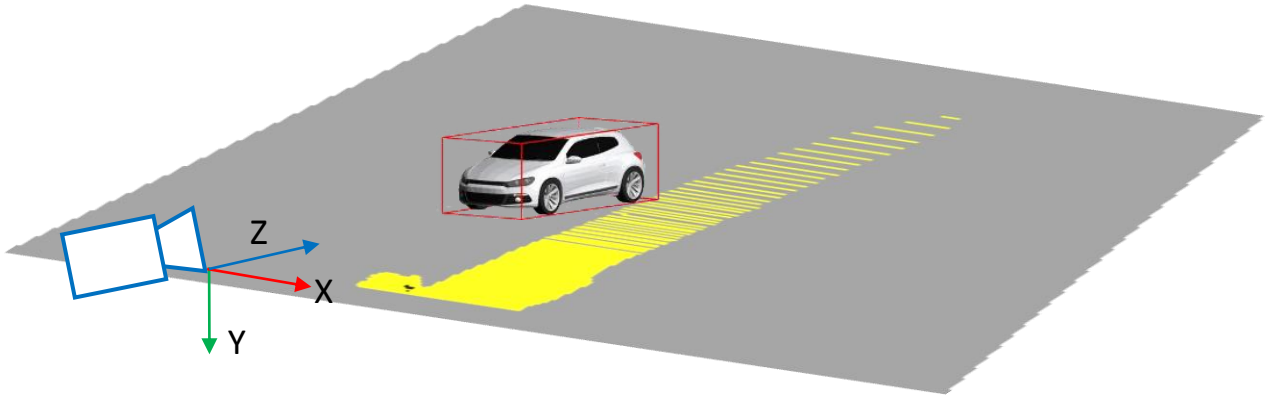
Back-projection
(Ground Prior)



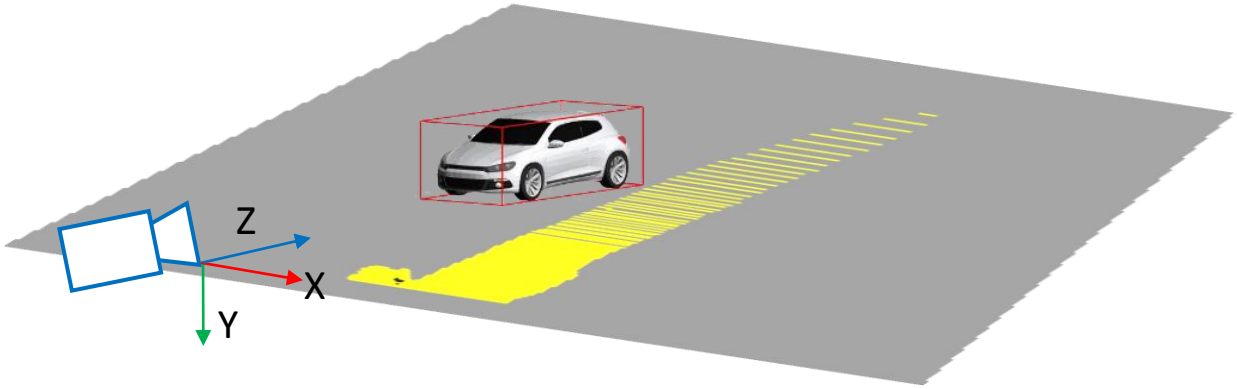
# 3D Candidates Sampling

Road semantic

Back-projection
(Ground Prior)



# 3D Candidates Sampling

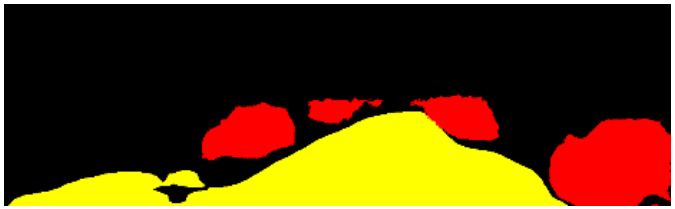



projection

2D candidate boxes

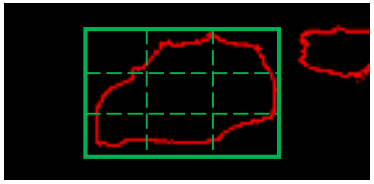
# Feature Computation

2D candidate boxes

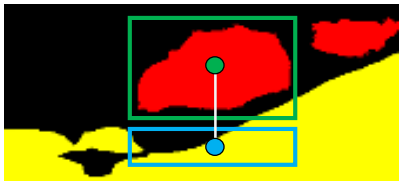


class semantic

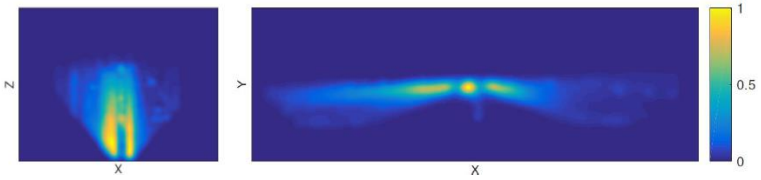
instance semantic



shape



context

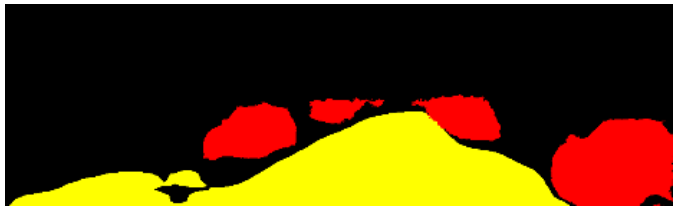



location

Features

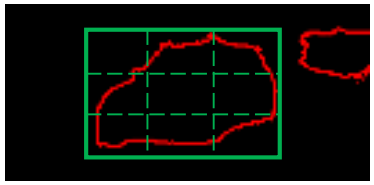
[1] S. Zheng, et al. Conditional random fields as recurrent neural networks. ICCV'15
[2] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. arXiv, 2015.
[3] Z. Zhang, et al. Monocular Object Instance Segmentation and Depth Ordering with CNNs. ICCV'15.

# Feature Computation

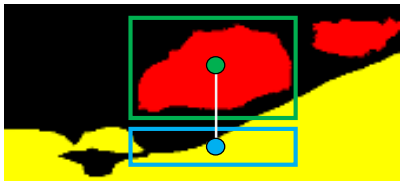


class semantic

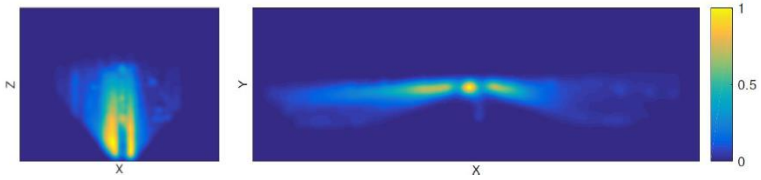
instance semantic



shape



context



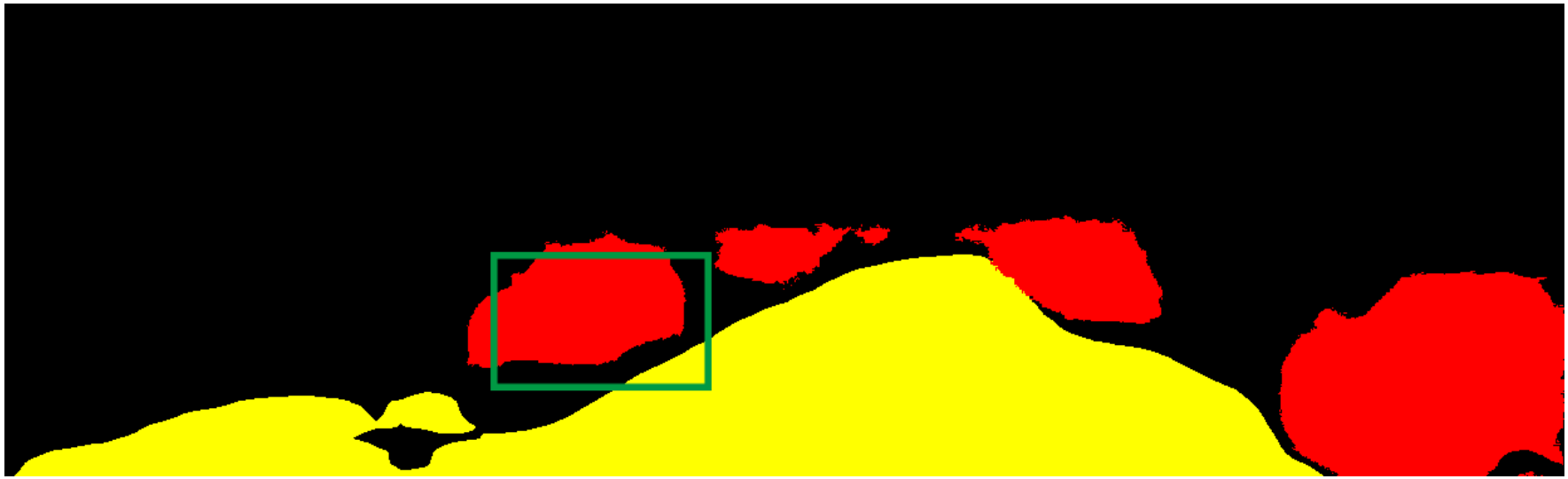

location

$$E(\mathbf{x},\mathbf{y}) = E_{sem}(\mathbf{x},\mathbf{y}) + E_{inst}(\mathbf{x},\mathbf{y}) + E_{shape}(\mathbf{x},\mathbf{y}) + E_{context}(\mathbf{x},\mathbf{y}) + E_{loc}(\mathbf{x},\mathbf{y})$$

# Energy Terms

$$E(\mathbf{x},\mathbf{y}) = E_{sem}(\mathbf{x},\mathbf{y}) + E_{inst}(\mathbf{x},\mathbf{y}) + E_{shape}(\mathbf{x},\mathbf{y}) + E_{context}(\mathbf{x},\mathbf{y}) + E_{loc}(\mathbf{x},\mathbf{y})$$

Class Semantics $E_{sem}(\mathbf{x},\mathbf{y}) = E_{seg}(\mathbf{x},\mathbf{y}) + E_{non-seg}(\mathbf{x},\mathbf{y})$

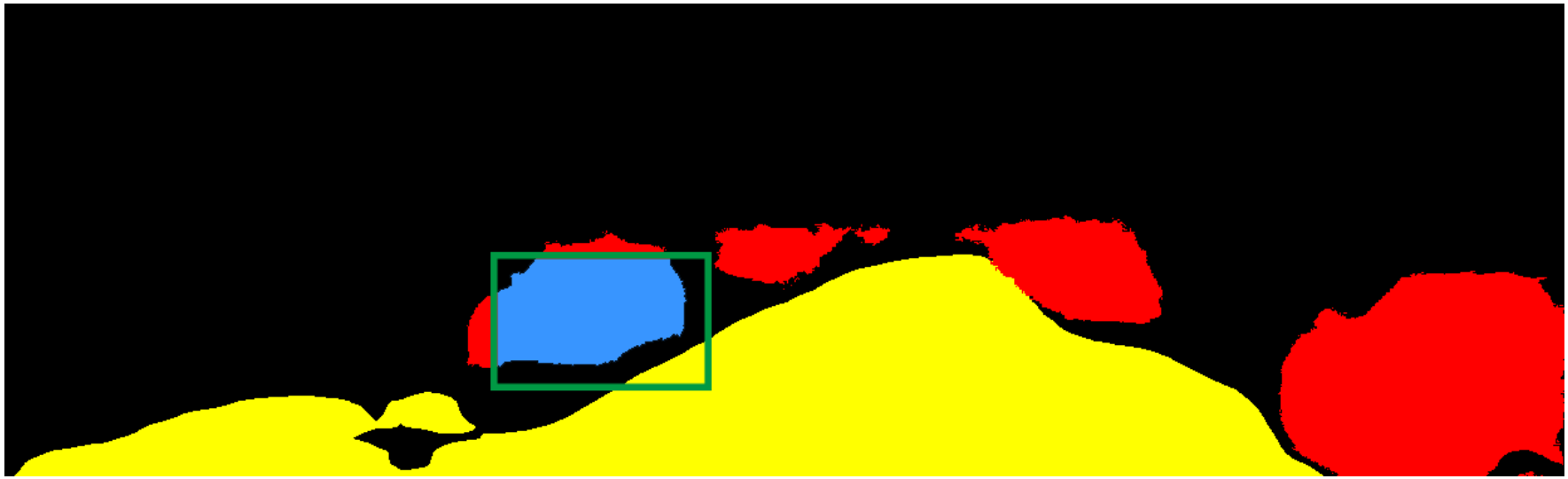


# Energy Terms

$$E(\mathbf{x},\mathbf{y}) = E_{sem}(\mathbf{x},\mathbf{y}) + E_{inst}(\mathbf{x},\mathbf{y}) + E_{shape}(\mathbf{x},\mathbf{y}) + E_{context}(\mathbf{x},\mathbf{y}) + E_{loc}(\mathbf{x},\mathbf{y})$$

Class Semantics $E_{sem}(\mathbf{x},\mathbf{y}) = E_{seg}(\mathbf{x},\mathbf{y}) + E_{non-seg}(\mathbf{x},\mathbf{y})$

e.g., car

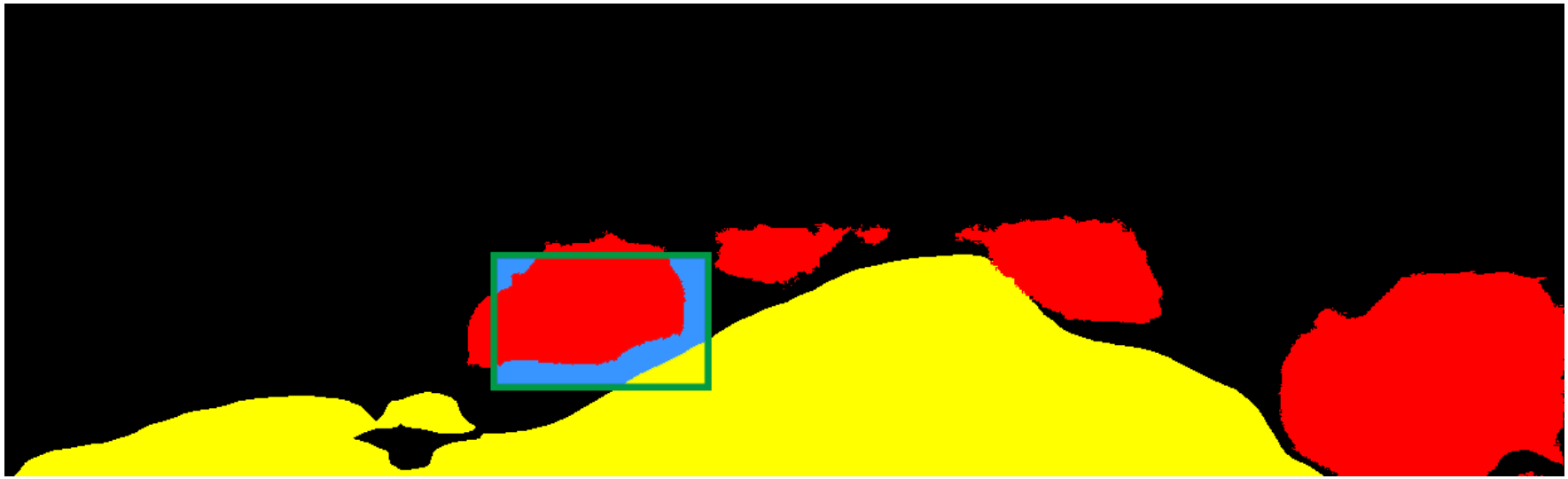


# Energy Terms

$$E(\mathbf{x},\mathbf{y}) = E_{sem}(\mathbf{x},\mathbf{y}) + E_{inst}(\mathbf{x},\mathbf{y}) + E_{shape}(\mathbf{x},\mathbf{y}) + E_{context}(\mathbf{x},\mathbf{y}) + E_{loc}(\mathbf{x},\mathbf{y})$$

Class Semantics $E_{sem}(\mathbf{x},\mathbf{y}) = E_{seg}(\mathbf{x},\mathbf{y}) + E_{non-seg}(\mathbf{x},\mathbf{y})$

e.g., car background

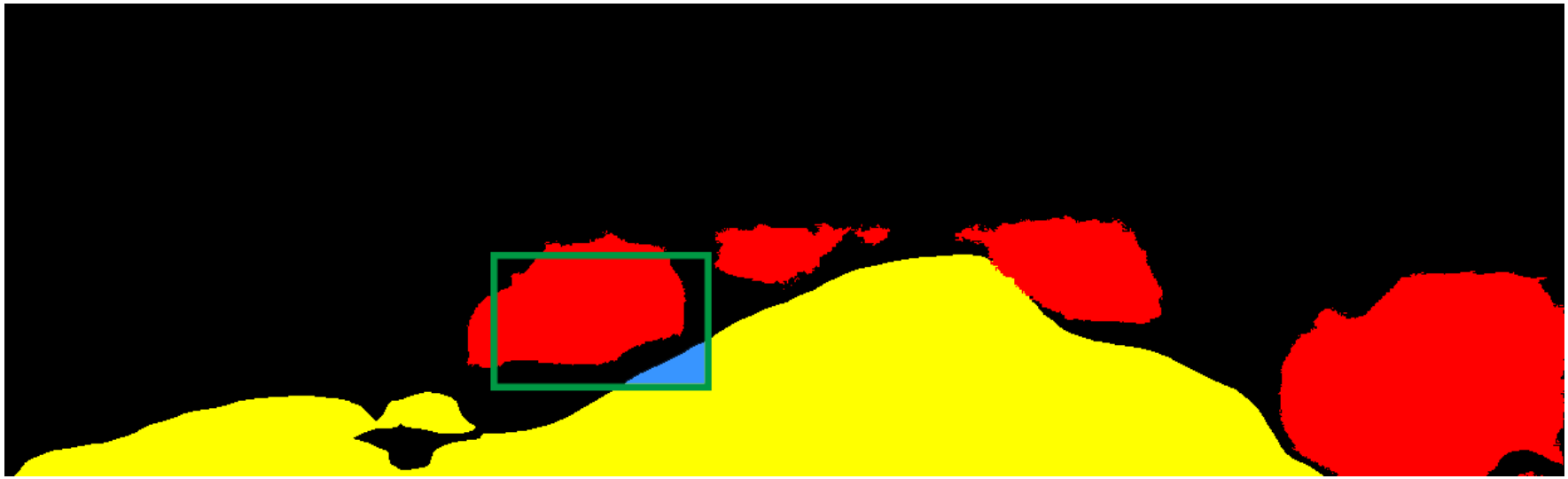


# Energy Terms

$$E(\mathbf{x},\mathbf{y}) = E_{sem}(\mathbf{x},\mathbf{y}) + E_{inst}(\mathbf{x},\mathbf{y}) + E_{shape}(\mathbf{x},\mathbf{y}) + E_{context}(\mathbf{x},\mathbf{y}) + E_{loc}(\mathbf{x},\mathbf{y})$$

Class Semantics $E_{sem}(\mathbf{x},\mathbf{y}) = E_{seg}(\mathbf{x},\mathbf{y}) + E_{non-seg}(\mathbf{x},\mathbf{y})$

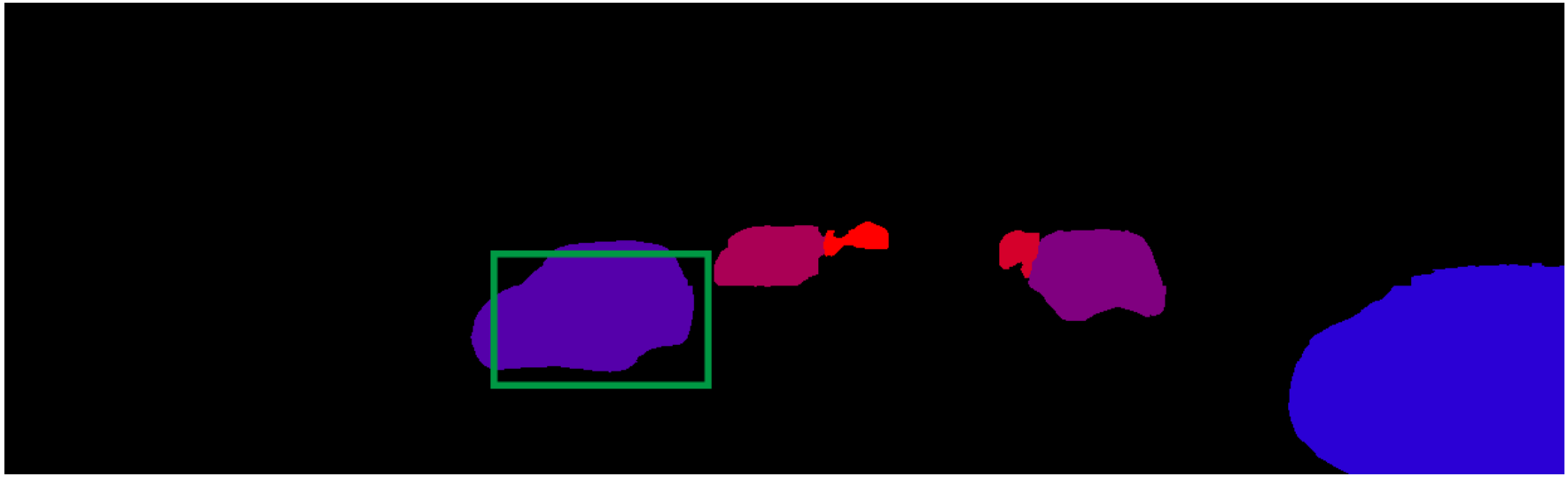
e.g., car background, road



# Energy Terms

$$E(\mathbf{x},\mathbf{y}) = E_{sem}(\mathbf{x},\mathbf{y}) + E_{inst}(\mathbf{x},\mathbf{y}) + E_{shape}(\mathbf{x},\mathbf{y}) + E_{context}(\mathbf{x},\mathbf{y}) + E_{loc}(\mathbf{x},\mathbf{y})$$

Instance Semantics $E_{inst}(\mathbf{x},\mathbf{y}) = E_{seg-in}(\mathbf{x},\mathbf{y}) + E_{bg-in}(\mathbf{x},\mathbf{y})$

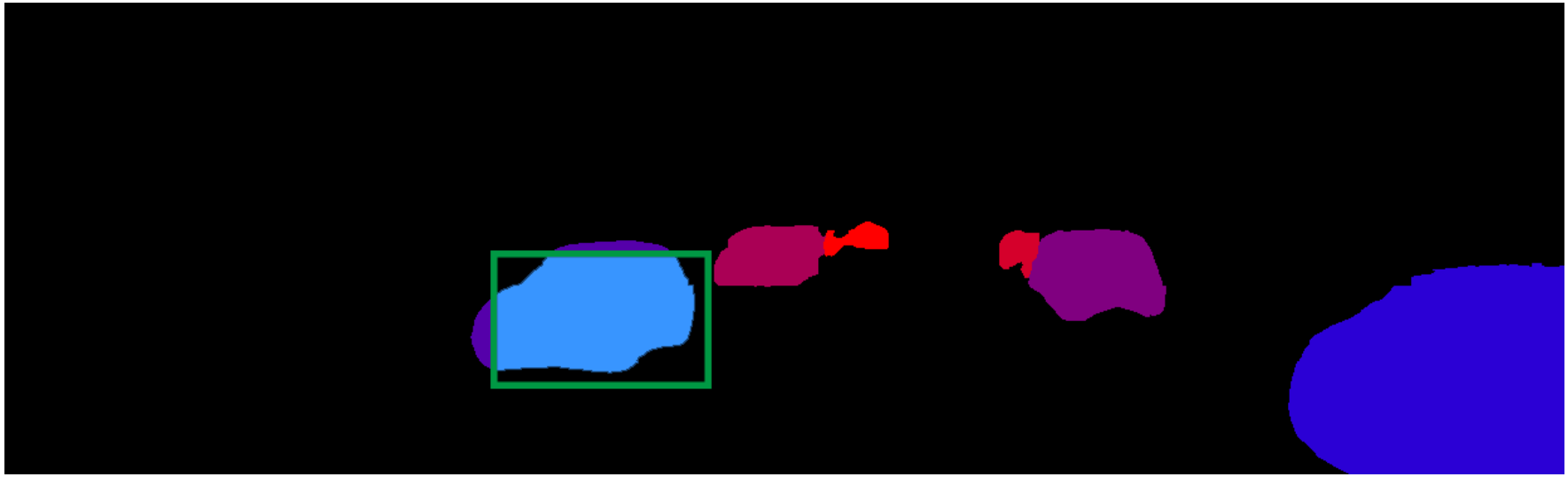


# Energy Terms

$$E(\mathbf{x},\mathbf{y}) = E_{sem}(\mathbf{x},\mathbf{y}) + E_{inst}(\mathbf{x},\mathbf{y}) + E_{shape}(\mathbf{x},\mathbf{y}) + E_{context}(\mathbf{x},\mathbf{y}) + E_{loc}(\mathbf{x},\mathbf{y})$$

Instance Semantics $E_{inst}(\mathbf{x},\mathbf{y}) = E_{seg-in}(\mathbf{x},\mathbf{y}) + E_{bg-in}(\mathbf{x},\mathbf{y})$

e.g., car instance

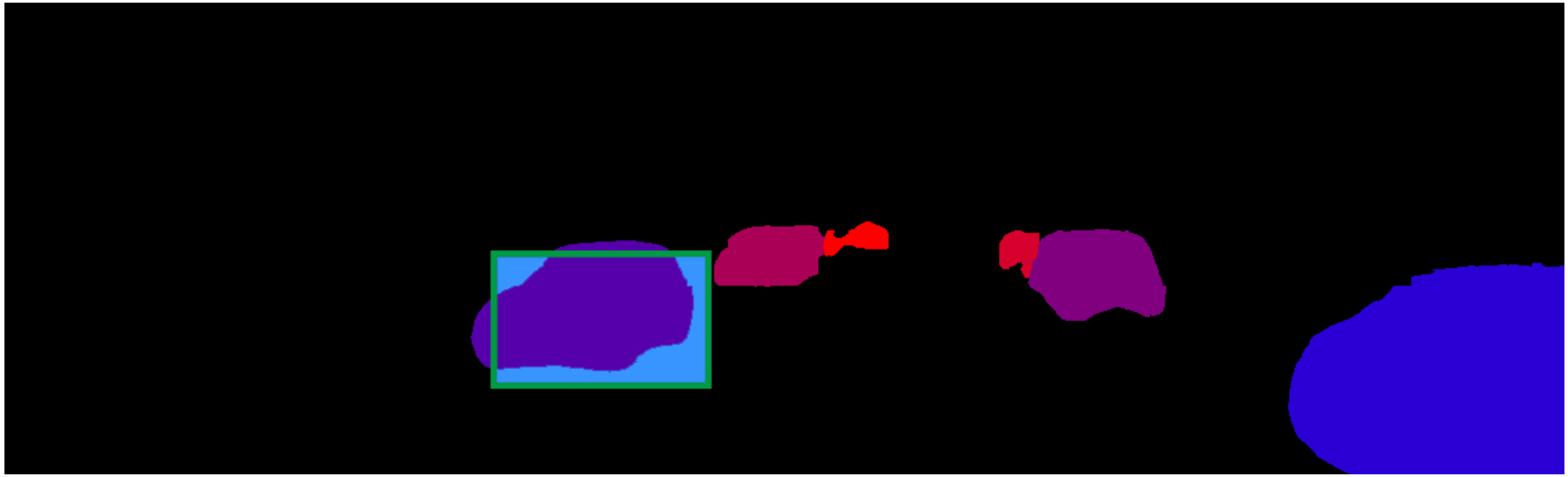


# Energy Terms

$$E(\mathbf{x},\mathbf{y}) = E_{sem}(\mathbf{x},\mathbf{y}) + E_{inst}(\mathbf{x},\mathbf{y}) + E_{shape}(\mathbf{x},\mathbf{y}) + E_{context}(\mathbf{x},\mathbf{y}) + E_{loc}(\mathbf{x},\mathbf{y})$$

Instance Semantics $E_{inst}(\mathbf{x},\mathbf{y}) = E_{seg-in}(\mathbf{x},\mathbf{y}) + E_{bg-in}(\mathbf{x},\mathbf{y})$

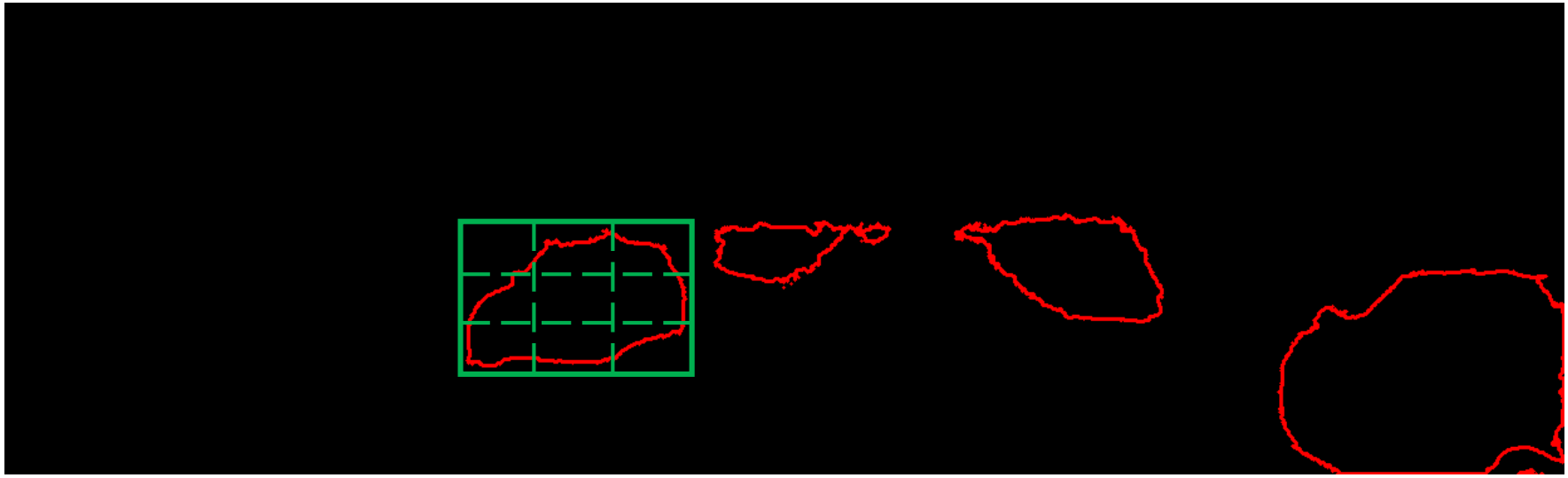
e.g., car instance  background



# Energy Terms

$$E(\mathbf{x},\mathbf{y}) = E_{sem}(\mathbf{x},\mathbf{y}) + E_{inst}(\mathbf{x},\mathbf{y}) + E_{shape}(\mathbf{x},\mathbf{y}) + E_{context}(\mathbf{x},\mathbf{y}) + E_{loc}(\mathbf{x},\mathbf{y})$$

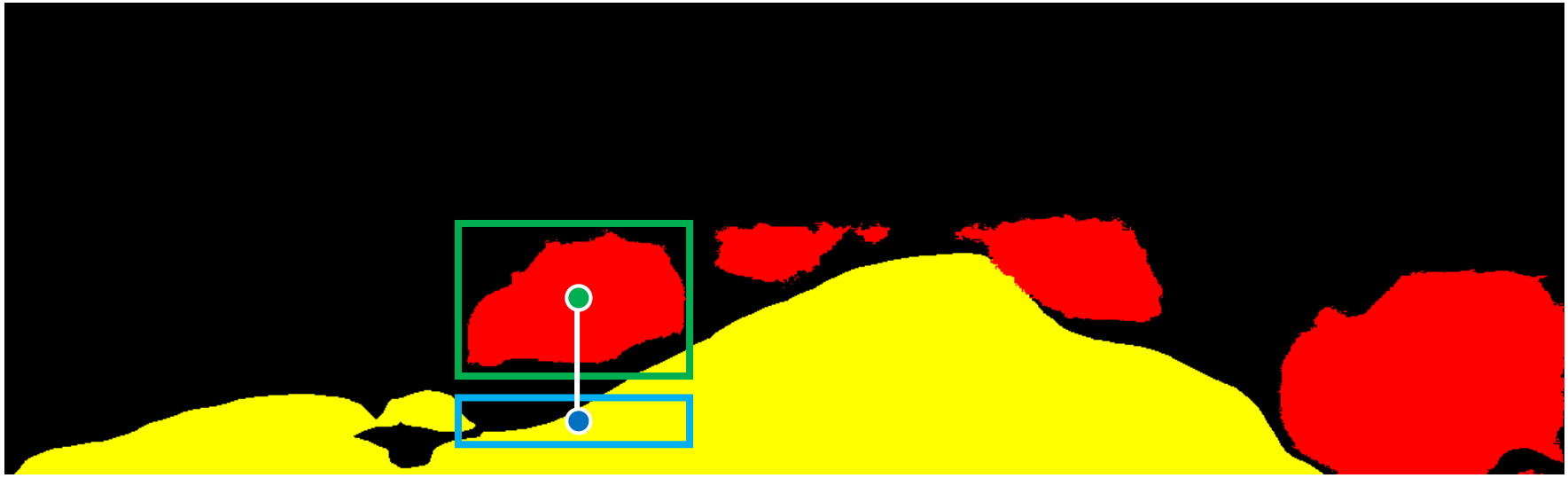
Shape: length of contours in a (1 + 3x3) Pyramid



# Energy Terms

$$E(\mathbf{x},\mathbf{y}) = E_{sem}(\mathbf{x},\mathbf{y}) + E_{inst}(\mathbf{x},\mathbf{y}) + E_{shape}(\mathbf{x},\mathbf{y}) + E_{context}(\mathbf{x},\mathbf{y}) + E_{loc}(\mathbf{x},\mathbf{y})$$

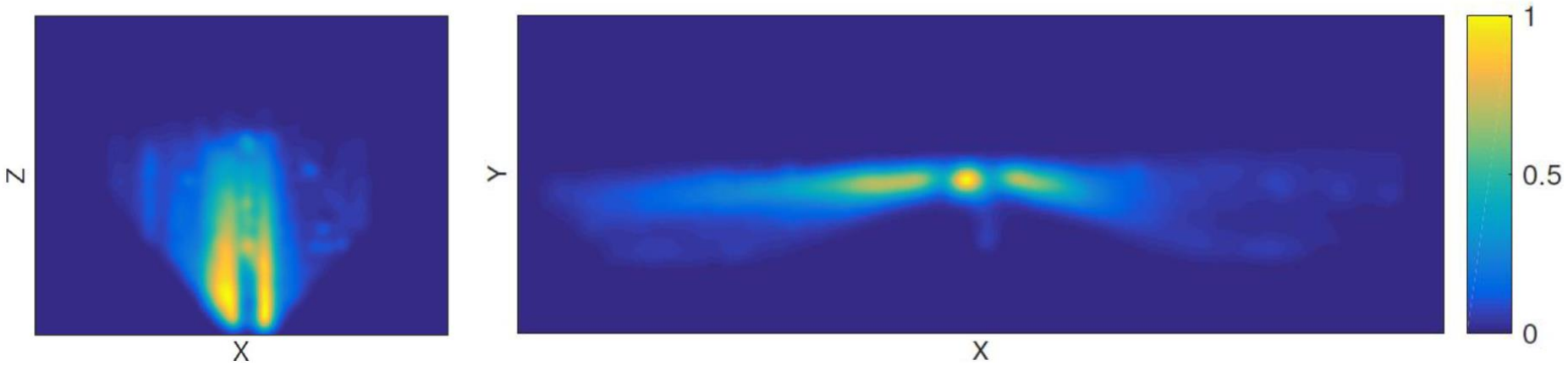
Context: semantic features in the bottom region



# Energy Terms

$$E(\mathbf{x},\mathbf{y}) = E_{sem}(\mathbf{x},\mathbf{y}) + E_{inst}(\mathbf{x},\mathbf{y}) + E_{shape}(\mathbf{x},\mathbf{y}) + E_{context}(\mathbf{x},\mathbf{y}) + E_{loc}(\mathbf{x},\mathbf{y})$$

**Location Prior**: Kernel Density Estimation of object location in 3D space and image plane

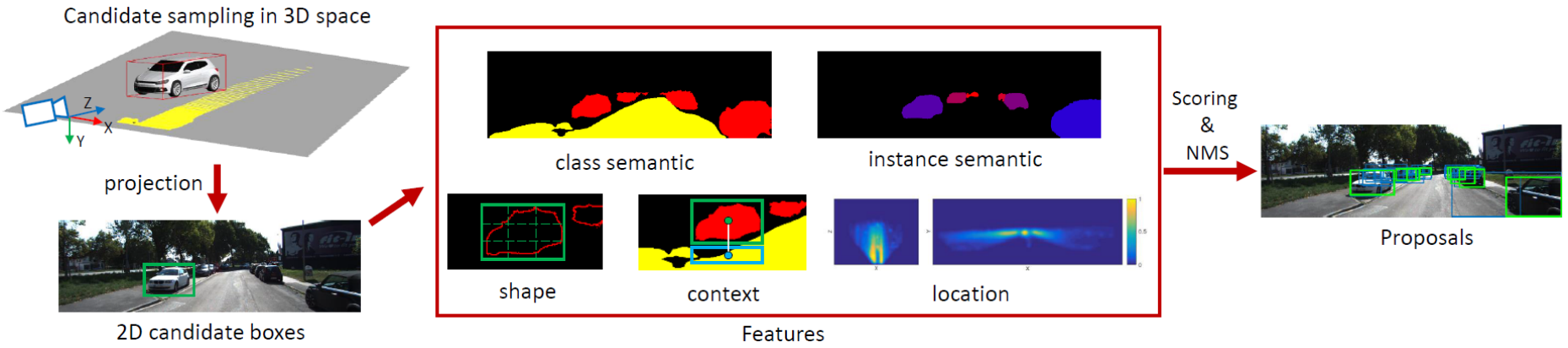


# Inference & Learning

## Inference:

- Exhaustive Search
- Computation: 2D Integral Images

$$\mathbf{y}^* = \underset{\mathbf{y}}{\operatorname{argmin}}\, E_{sem}(\mathbf{x},\mathbf{y}) + E_{inst}(\mathbf{x},\mathbf{y}) + E_{shape}(\mathbf{x},\mathbf{y}) + E_{context}(\mathbf{x},\mathbf{y}) + E_{loc}(\mathbf{x},\mathbf{y})$$



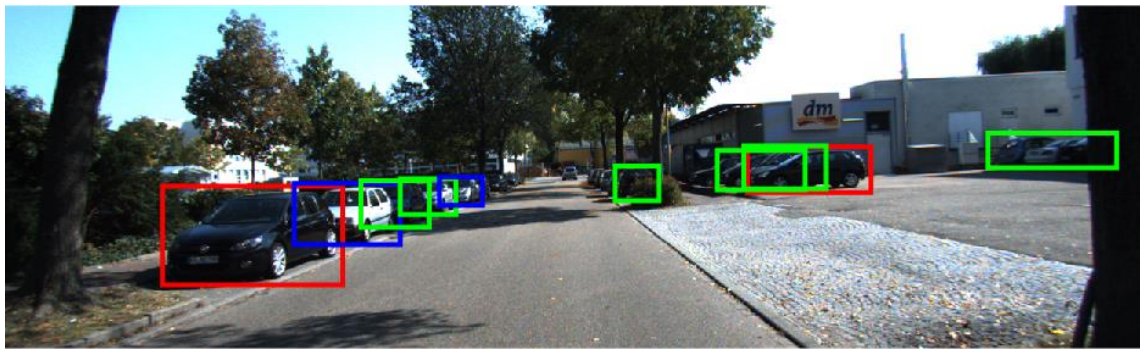

## Learning: Structured SVM

- **Task loss**: 3D IoU

# Results

# Results: Experiment Settings

- KITTI object detection benchmark
  - **Categories**: *car, pedestrian, cyclist*
  - **Test**: 7418 images for training, 7518 images for testing
  - **Validation**: 3712 images for training, 3769 images for validation
  - **Tasks**:
    - object detection
    - object detection and orientation estimation
  - **Overlap criteria**: 0.7 for Car, 0.5 for Pedestrian/Cyclist
  - **Difficulties**: easy/moderate/hard



Easy
Moderate
Hard

# Experiments

- **Proposal Recall**
- **KITTI Tasks:**
  - object detection
  - object detection and orientation estimation
- **3D Evaluation**
  - 3D object localization
  - 3D object detection
- **Stereo vs LIDAR**
- **Comparison of Network Architectures**

# Results: Proposal Recall

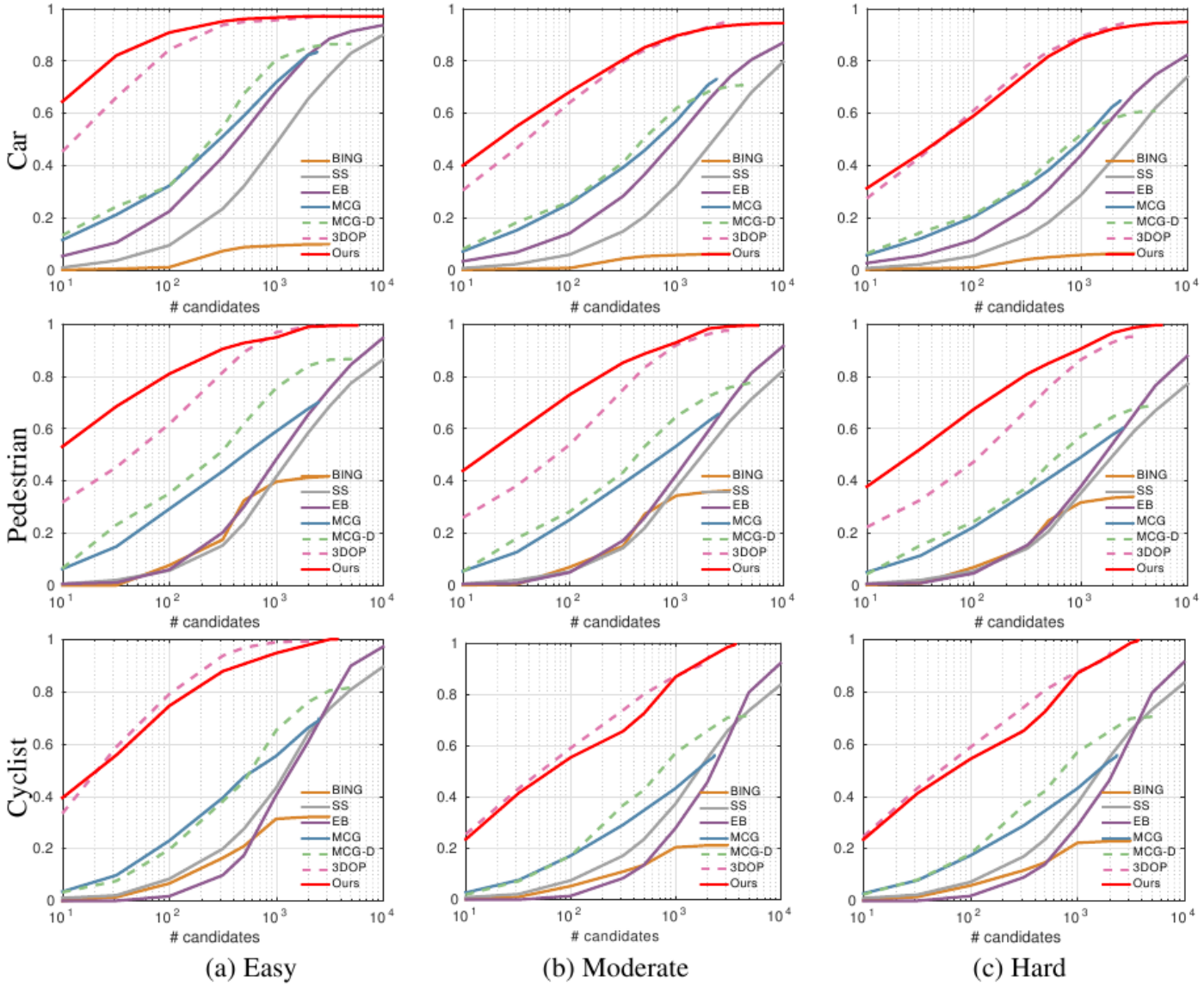
**2D Recall vs #Proposals**: IoU = 0.7 for *Car*, and 0.5 for *Pedestrian/Cyclist*

2D methods:

- **BING**
- **SS**
- **EB**
- **MCG**
- **Mono3D**

3D methods:

- **MCG-D**
- **3DOP**



# Results: Proposal Recall

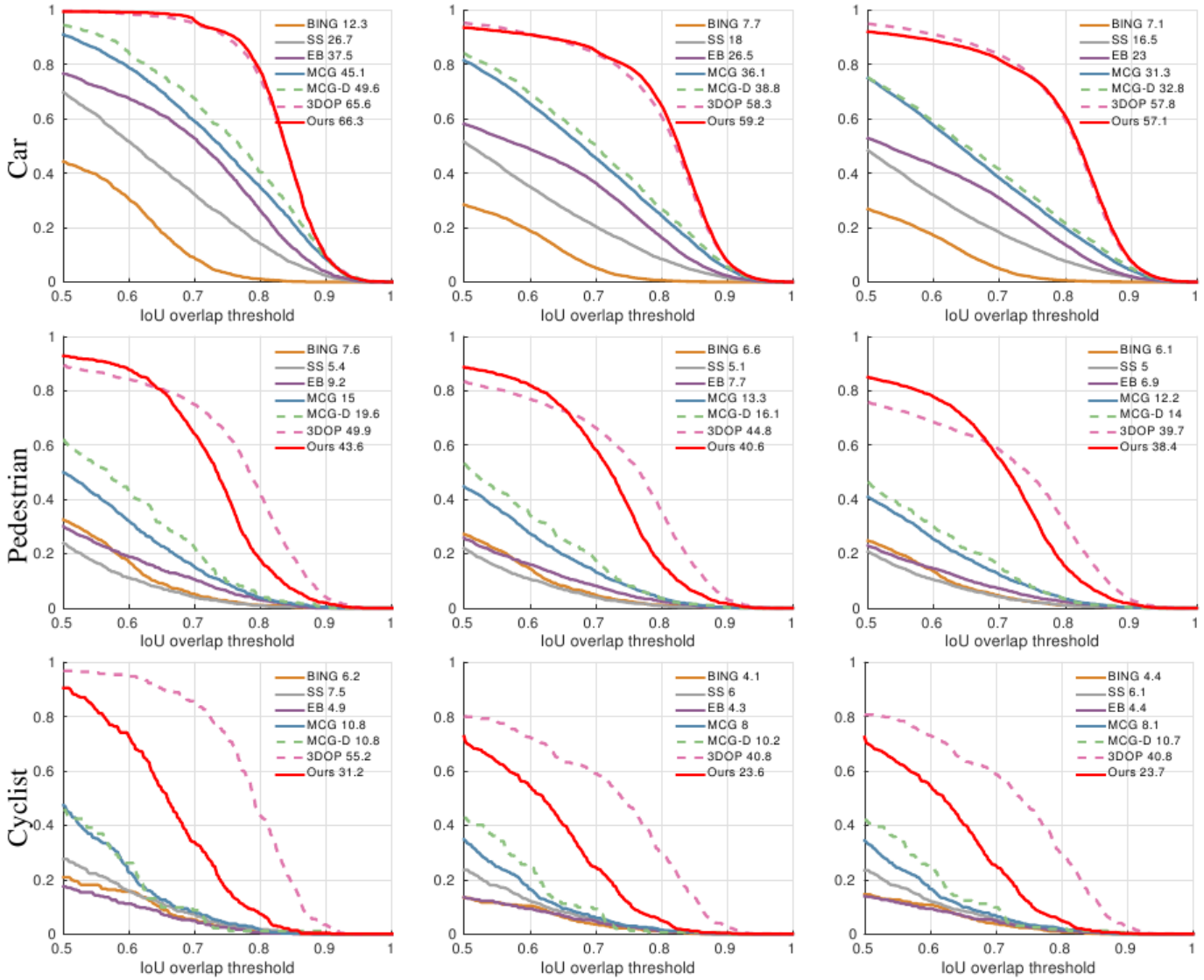
**2D Recall vs #Proposals**: IoU = 0.7 for *Car*, and 0.5 for *Pedestrian/Cyclist*

2D methods:

- **BING**
- **SS**
- **EB**
- **MCG**
- **Mono3D**

3D methods:

- **MCG-D**
- **3DOP**



# Results: Proposal Recall

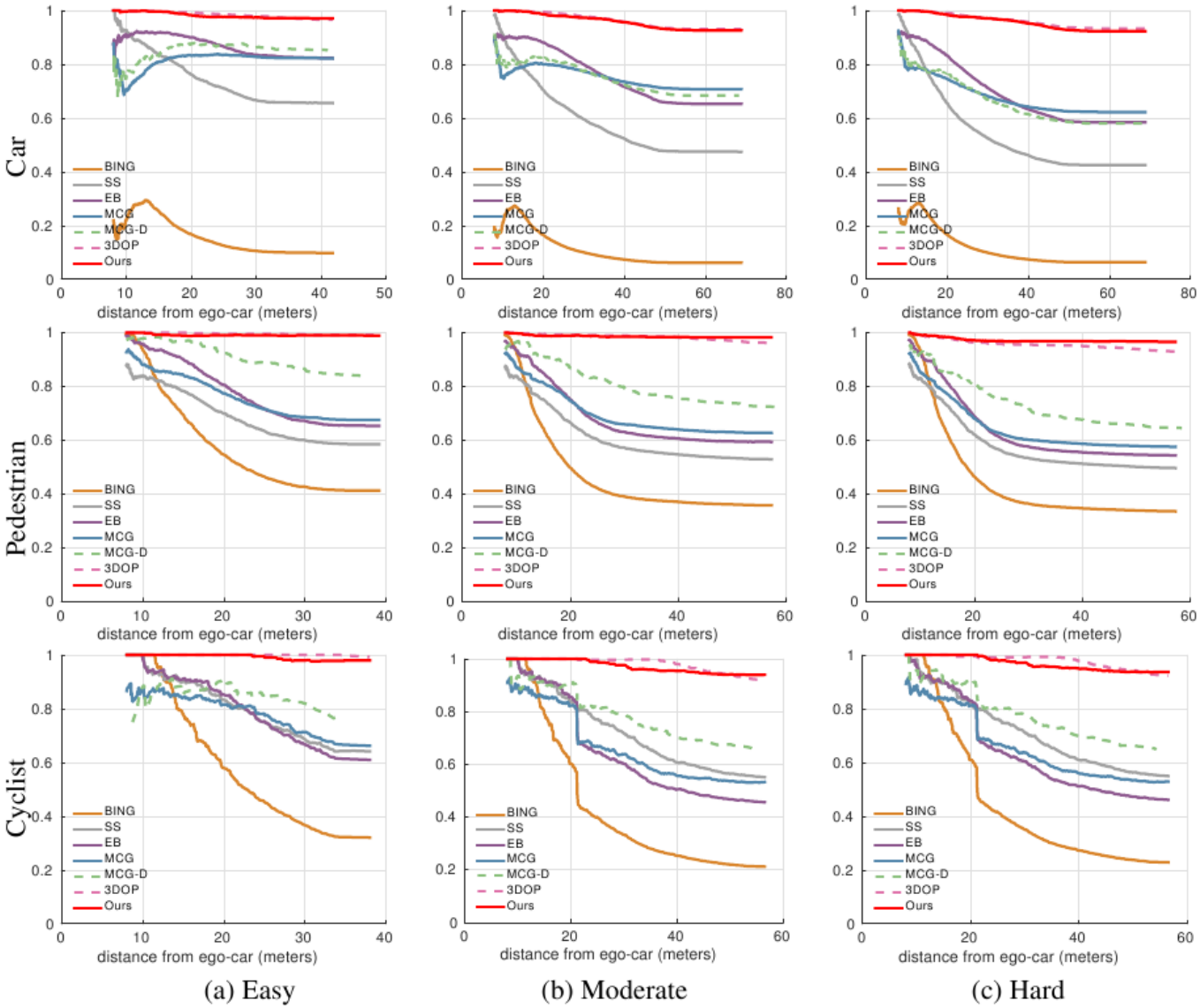
**2D Recall vs #Proposals**: IoU = 0.7 for *Car*, and 0.5 for *Ped./Cyc.*

2D methods:

- **BING**
- **SS**
- **EB**
- **MCG**
- **Mono3D**

3D methods:

- **MCG-D**
- **3DOP**



# Results: Object Detection and Orientation Estimation

## Results on KITTI test: Car

| | Object detection (AP) | | | Object detection and orientation estimation (AOS) | | |
|---|---|---|---|---|---|---|
| **Method** | **Easy** | **Moderate** | **Hard** | **Easy** | **Moderate** | **Hard** |
| SubCat [T-ITS'15] | 84.14 | 75.46 | 59.71 | 74.42 | 83.41 | 58.83 |
| 3DVP [CVPR'15] | 87.46 | 75.77 | 65.38 | 86.92 | 74.59 | 64.11 |
| AOG [ECCV'14] | 84.80 | 75.94 | 60.70 | - | - | - |
| Regionlets [TPAMI'15] | 84.75 | 76.45 | 59.70 | - | - | - |
| spLBP [T-ITS'15] | 87.19 | 77.40 | 60.60 | - | - | - |
| Faster R-CNN [NIPS'15] | 86.71 | 81.84 | 71.12 | - | - | - |
| **3DOP [NIPS'15]** | **93.04** | 88.64 | 79.10 | **91.44** | 86.10 | 76.52 |
| **Mono3D [CVPR'16]** | 92.33 | 88.66 | 78.96 | 91.01 | 86.62 | 76.84 |
| SDP+RPN [CVPR'16] | 90.14 | 88.85 | 78.38 | - | - | - |
| MS-CNN [ECCV'16] | 90.03 | 89.02 | 76.11 | - | - | - |
| SubCNN [arXiv'16] | 90.81 | **89.04** | **79.27** | 90.67 | **88.62** | **78.68** |

# Results: Object Detection and Orientation Estimation

## Results on KITTI test: Pedestrian

| | Object detection (AP) | | | Object detection and orientation estimation (AOS) | | |
|---|---|---|---|---|---|---|
| **Method** | **Easy** | **Moderate** | **Hard** | **Easy** | **Moderate** | **Hard** |
| DPM-VOC+VP [TPAMI'15] | 59.48 | 44.86 | 40.37 | 53.55 | 39.83 | 35.73 |
| FilteredICF [CVPR'15] | 67.65 | 56.75 | 51.12 | - | - | - |
| DeepParts [ICCV'15] | 70.49 | 58.67 | 52.78 | - | - | - |
| CompACT-Deep [ICCV'15] | 70.69 | 58.74 | 52.71 | - | - | - |
| Regionlets [TPAMI'15] | 73.14 | 61.15 | 55.21 | - | - | - |
| Faster R-CNN [NIPS'15] | 78.86 | 65.90 | 61.18 | - | - | - |
| **Mono3D [CVPR'16]** | 80.35 | 66.68 | 63.44 | 71.15 | 58.15 | 54.94 |
| **3DOP [NIPS'15]** | 81.78 | 67.47 | 64.70 | 72.94 | 59.80 | 57.03 |
| SDP+RPN [CVPR'16] | 80.09 | 70.16 | 64.82 | - | - | - |
| SubCNN [arXiv'16] | 83.28 | 71.33 | 66.36 | **78.45** | **66.28** | **61.36** |
| MS-CNN [ECCV'16] | **83.92** | **73.70** | **68.31** | - | - | - |

# Results: Object Detection and Orientation Estimation

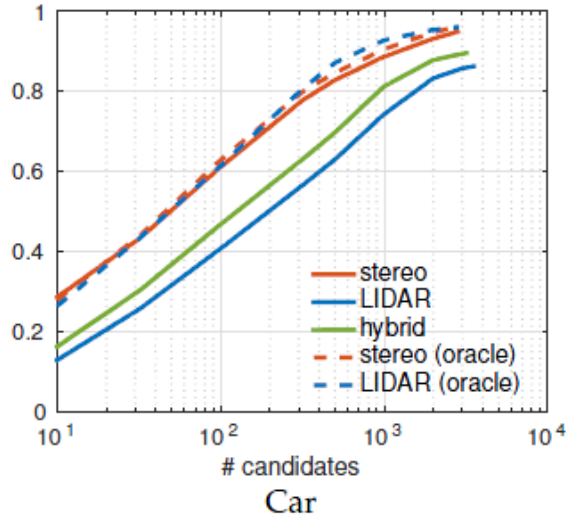
## Results on KITTI test: Cyclist

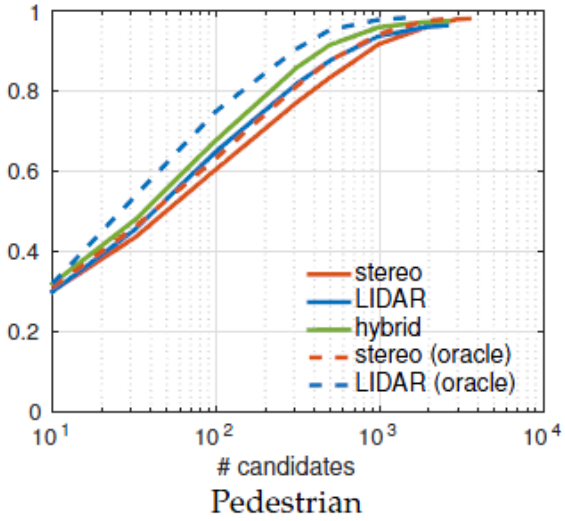
| | Object detection (AP) | | | Object detection and orientation estimation (AOS) | | |
|---|---|---|---|---|---|---|
| **Method** | **Easy** | **Moderate** | **Hard** | **Easy** | **Moderate** | **Hard** |
| DPM-VOC+VP [TPAMI'15] | 42.43 | 31.08 | 28.23 | 30.52 | 23.17 | 21.58 |
| pAUCEnsT [arXiv'14] | 51.62 | 38.03 | 33.38 | - | - | - |
| MV-RGBD-RF [IV'15] | 52.97 | 42.61 | 37.42 | - | - | - |
| Regionlets [TPAMI'15] | 70.41 | 58.72 | 51.83 | - | - | - |
| Faster R-CNN [NIPS'15] | 72.26 | 63.35 | 55.90 | - | - | - |
| **Mono3D [CVPR'16]** | 76.04 | 66.36 | 58.87 | 65.56 | 54.97 | 48.77 |
| **3DOP [NIPS'15]** | 78.39 | 68.94 | 61.37 | 70.13 | 58.68 | 52.35 |
| SubCNN [arXiv'16] | 79.48 | 71.06 | 62.68 | **72.00** | **63.65** | **56.32** |
| SDP+RPN [CVPR'16] | 81.37 | 73.74 | 65.31 | - | - | - |
| MS-CNN [ECCV'16] | **84.06** | **75.46** | **66.07** | - | - | - |

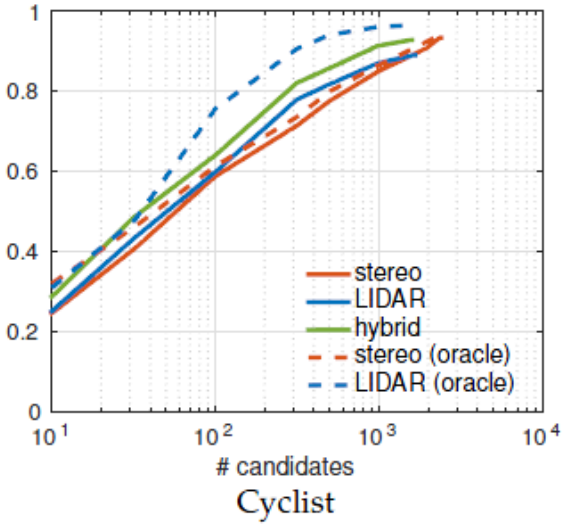
# Stereo vs LIDAR

*Moderate data

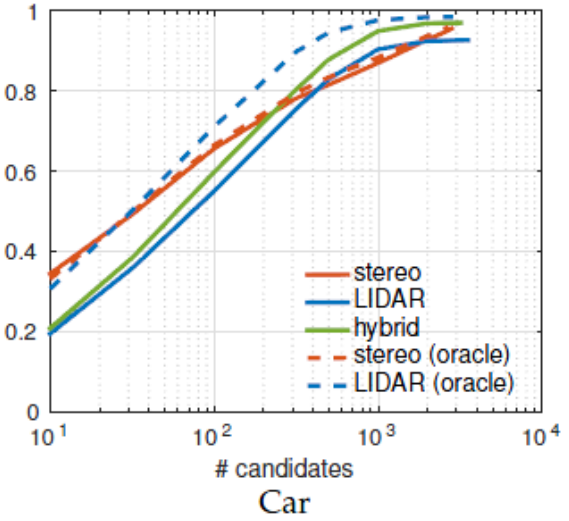
## 2D Recall vs #Proposals: IoU = 0.7 for *Car*, and 0.5 for *Ped./Cyc.*

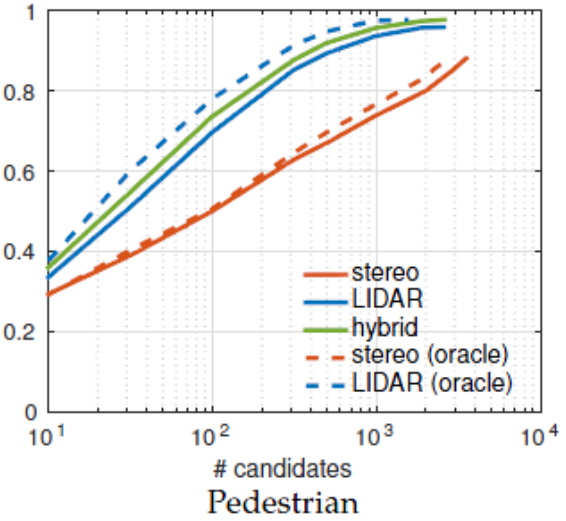


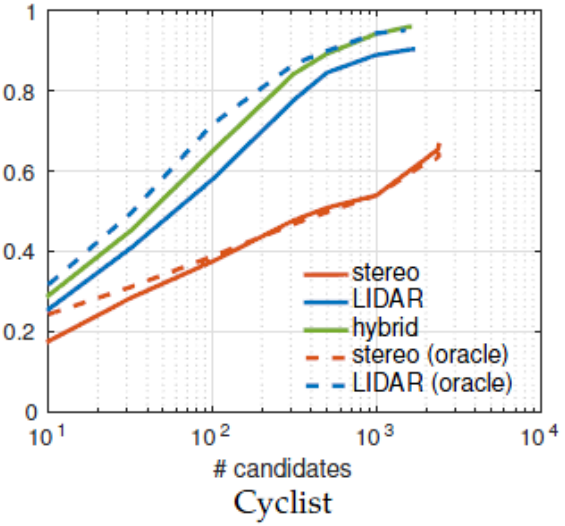




## 3D Recall vs #Proposals: IoU = 0.25



# Stereo vs LIDAR

## 2D Object Detection and Orientation Estimation (Car)

Stereo works better!

| Data | $AP_{2D}$ | | | AOS | | |
|---|---|---|---|---|---|---|
| | Easy | Moderate | Hard | Easy | Moderate | Hard |
| stereo (oracle) | 92.73 | 88.30 | 79.48 | 90.98 | 86.08 | 76.90 |
| LIDAR (oracle) | **93.21** | **88.77** | **79.70** | **91.67** | **86.61** | **77.22** |
| stereo | **93.08** | **88.07** | **79.39** | **91.58** | **85.80** | **76.80** |
| LIDAR | 87.78 | 79.51 | 70.74 | 85.90 | 77.24 | 68.23 |
| hybrid | 92.17 | 86.52 | 78.37 | 90.62 | 84.44 | 75.91 |

## 3D Object Detection (Car)

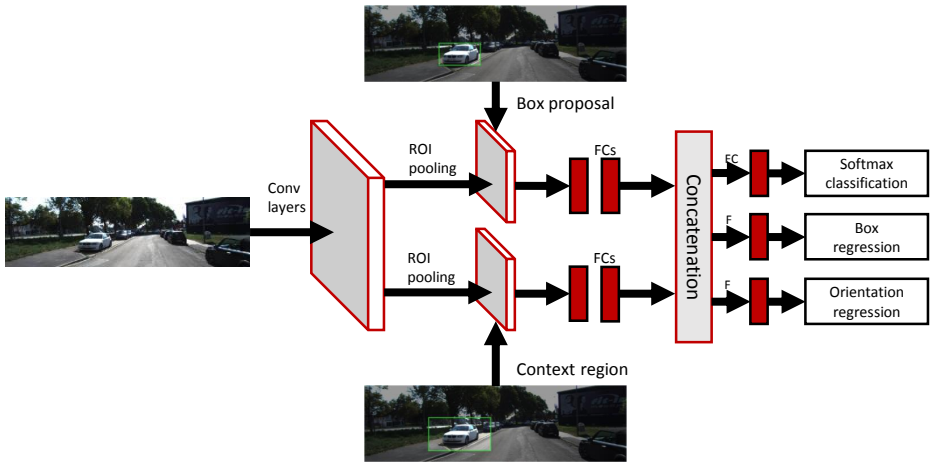
LIDAR works better!

| Data | $AP_{3D}$ | | | ALP (< 1m) | | | ALP (< 2m) | | |
|---|---|---|---|---|---|---|---|---|---|
| | Easy | Moderate | Hard | Easy | Moderate | Hard | Easy | Moderate | Hard |
| stereo (oracle) | **87.33** | 70.46 | 63.82 | **75.98** | 58.01 | 52.37 | **94.12** | 78.64 | 73.95 |
| LIDAR (oracle) | 84.63 | **82.04** | **74.92** | 75.32 | **74.54** | **68.50** | 91.88 | **89.29** | **84.06** |
| stereo | **88.45** | 69.52 | 62.65 | **77.90** | 58.09 | 52.17 | **94.39** | 77.57 | 72.69 |
| LIDAR | 80.73 | 73.56 | 66.83 | 72.26 | 67.77 | 62.42 | 87.98 | 81.07 | 76.52 |
| hybrid | 86.47 | **80.56** | **73.71** | 77.19 | **73.85** | **67.99** | 93.00 | **87.52** | **82.51** |

- ALP: Average Localization Precision

# Comparison of Network Architectures

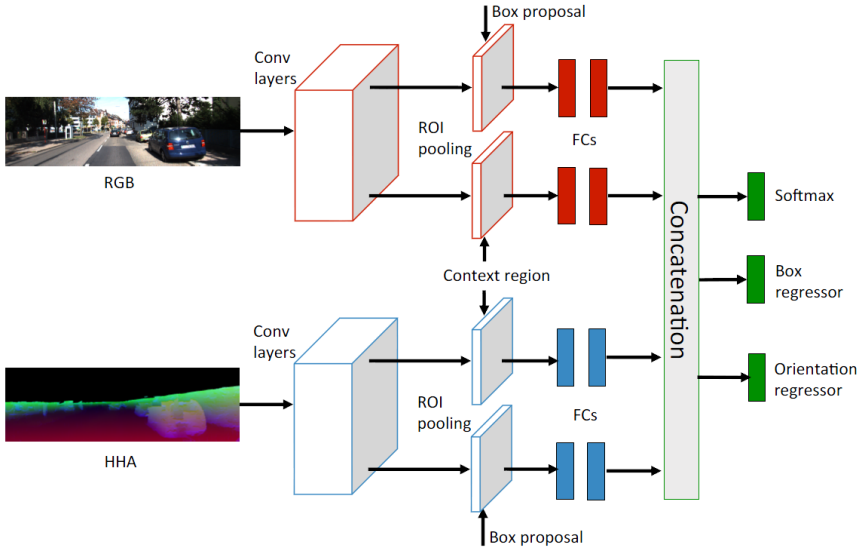
## Single-Stream Network




**Input:**

- RGB
- RGB-HHA

## Two-Stream Network




**Input:**

- RGB, HHA

# Comparison of Network Architectures

## 2D Object Detection and Orientation Estimation (Car)

| Data | Approach | $AP_{2D}$ | | | AOS | | |
|---|---|---|---|---|---|---|---|
| | | Easy | Moderate | Hard | Easy | Moderate | Hard |
| stereo | RGB | 92.56 | 87.27 | 78.38 | 90.24 | 83.98 | 74.69 |
| | RGB-HHA, one-stream | 90.81 | 87.41 | 78.82 | **90.80** | **84.28** | **75.39** |
| | RGB-HHA, two-stream | **93.03** | **87.97** | **78.98** | 90.34 | 84.27 | 74.90 |
| LIDAR | RGB | 87.12 | 78.64 | 69.85 | 84.30 | 75.08 | 66.13 |
| | RGB-HHA, one-stream | 87.42 | 78.98 | 70.11 | 84.88 | 75.51 | 66.53 |
| | RGB-HHA, two-stream | **88.04** | **79.39** | **70.48** | **85.05** | **75.70** | **66.63** |
| hybrid | RGB | 90.99 | 84.40 | 76.02 | 87.93 | 80.47 | 71.76 |
| | RGB-HHA, one-stream | 90.81 | 84.40 | **77.12** | 88.45 | **80.98** | **73.20** |
| | RGB-HHA, two-stream | **92.69** | **84.78** | 76.43 | **89.23** | 80.53 | 71.73 |

*VGG_CNN_M_1024 network is used

# Comparison of Network Architectures

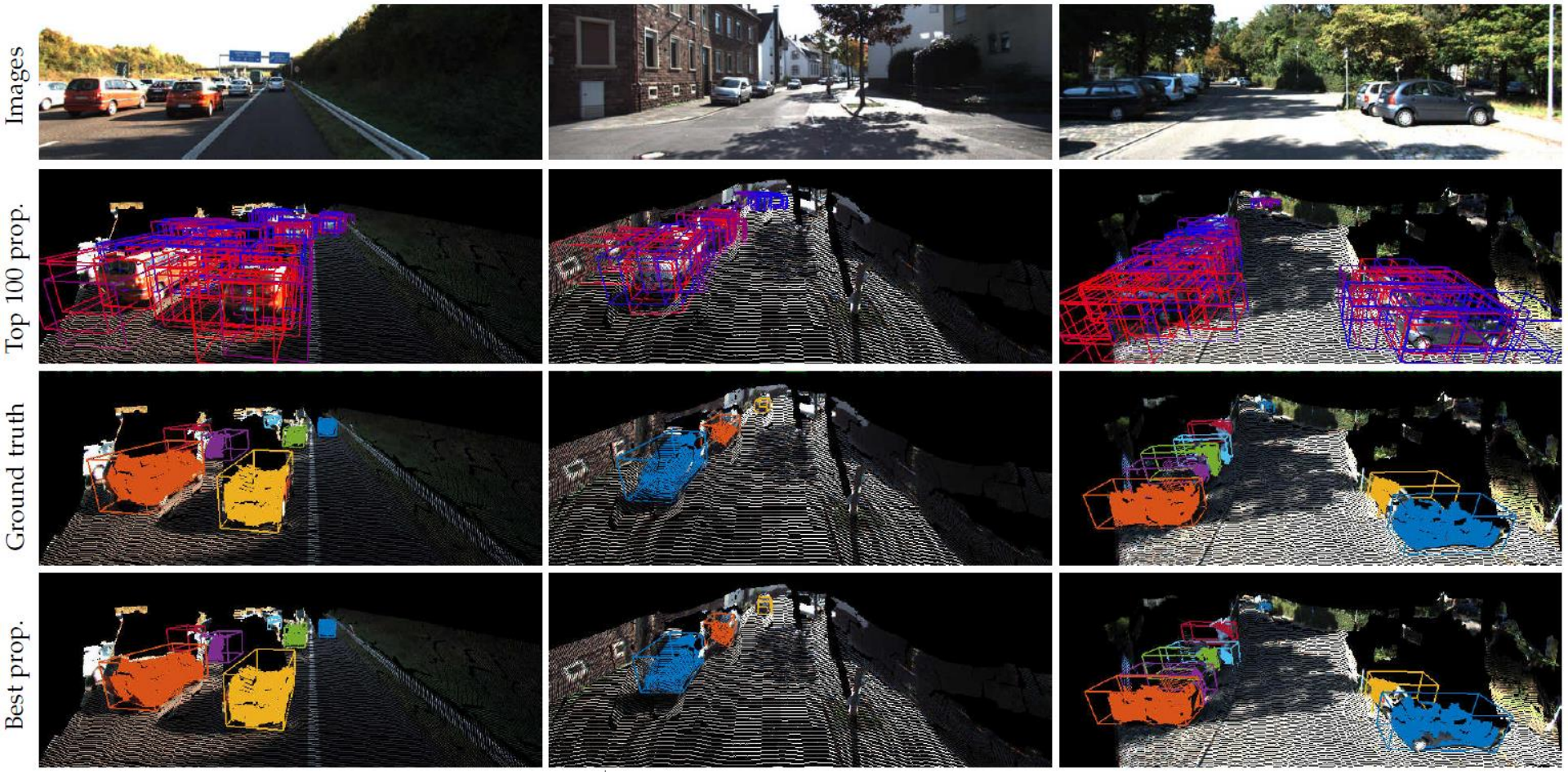
## 3D Object Detection (Car)

| Data | Approach | $AP_{3D}$ | | | ALP (< 1m) | | | ALP (< 2m) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Easy | Moderate | Hard | Easy | Moderate | Hard | Easy | Moderate | Hard |
| stereo | RGB | 77.50 | 56.97 | 50.84 | 64.89 | 47.34 | 42.20 | 89.02 | 69.27 | 64.74 |
| | RGB-HHA, one-stream | 89.22 | 68.19 | **62.24** | 81.00 | 58.53 | **52.76** | 95.23 | 76.57 | 72.50 |
| | RGB-HHA, two-stream | **90.43** | **68.90** | 62.22 | **82.25** | **58.99** | 52.71 | **95.74** | **77.66** | **73.03** |
| LIDAR | RGB | 76.51 | 68.77 | 62.10 | 66.46 | 62.14 | 57.53 | 86.22 | 78.67 | 74.44 |
| | RGB-HHA, one-stream | **84.83** | 73.92 | 67.55 | 76.61 | 68.52 | 63.25 | **92.16** | 82.56 | 77.95 |
| | RGB-HHA, two-stream | 84.02 | **74.11** | **67.62** | **77.40** | **69.62** | **63.92** | 91.32 | **82.72** | **78.13** |
| hybrid | RGB | 81.83 | 75.86 | 68.66 | 71.12 | 68.46 | 62.89 | 90.59 | 85.10 | 79.86 |
| | RGB-HHA, one-stream | 87.86 | 79.61 | 72.86 | 79.54 | 74.06 | 68.58 | 94.88 | 87.94 | 83.16 |
| | RGB-HHA, two-stream | **89.49** | **81.21** | **74.32** | **82.16** | **75.44** | **69.27** | **95.46** | **88.83** | **83.75** |

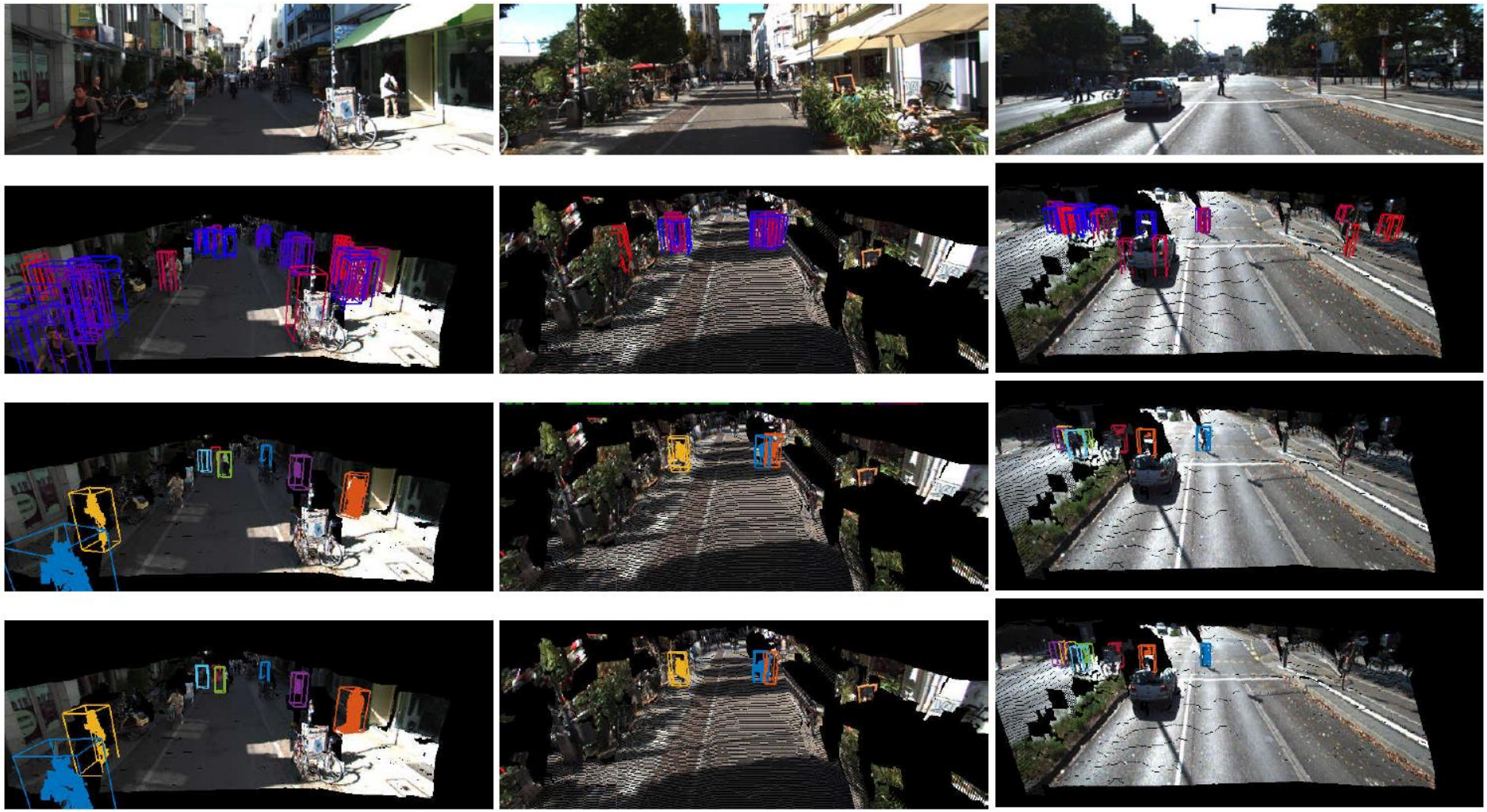
*VGG_CNN_M_1024 network is used

Depth is important for 3D detection!

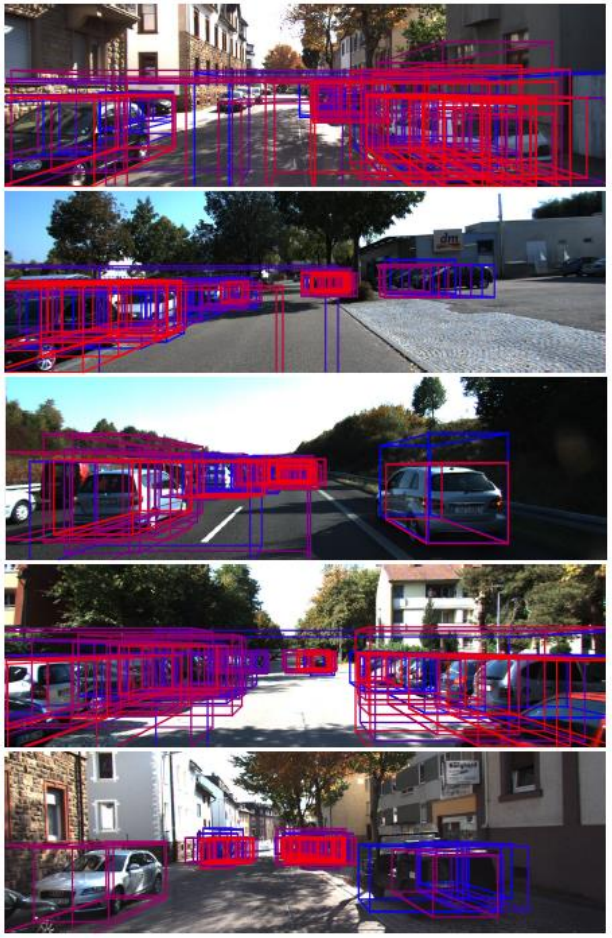
# Visualization: 3DOP



# Visualization: 3DOP



# Visualization: Mono3D



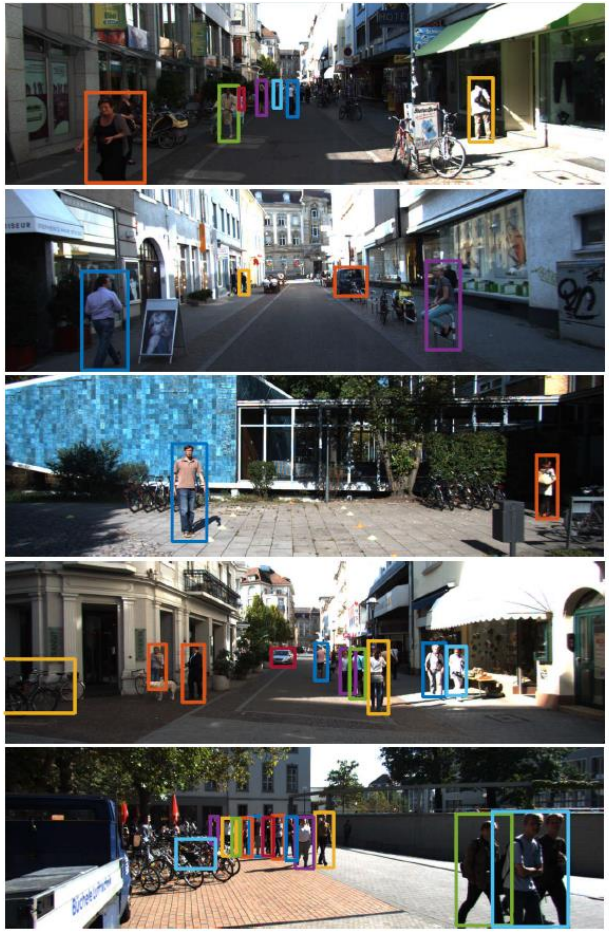
Top 50 prop.

2D detections

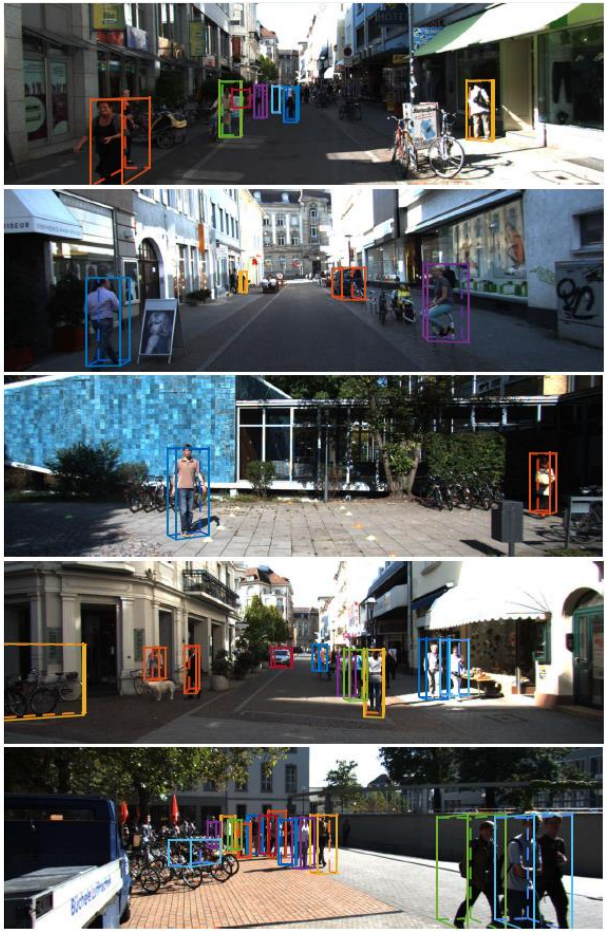
3D detections

# Visualization: Mono3D

Top 50 prop.



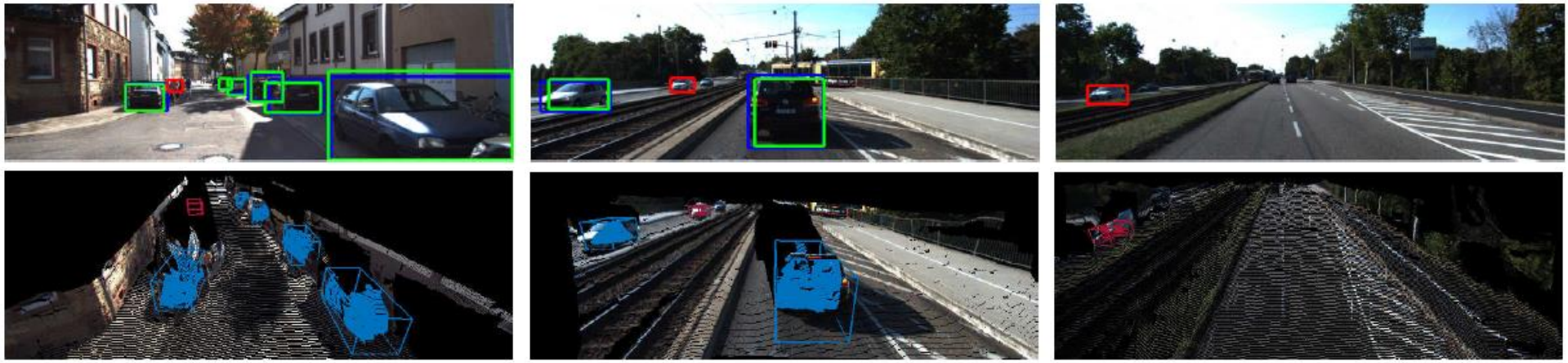
2D detections



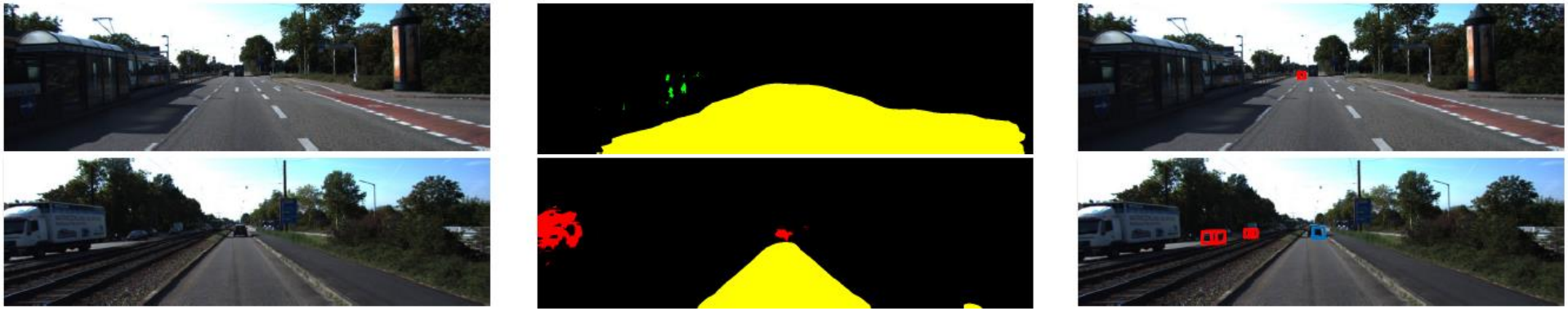
3D detections

# Failure Cases

## 3DOP



## Mono3D



# Conclusion

## What we have done:

- 3D object detection from stereo images
- 3D object detection from monocular images
- 3D evaluation is required for autonomous driving

## Future work

- Sensor fusion
- Combined with SLAM
- Combined with maps
- Etc.

## Code & Data

- **3DOP:** http://www.cs.toronto.edu/objprop3d/
- **Mono3D:** http://3dimage.ee.tsinghua.edu.cn/cxz/mono3d

Thank You!

Q&A