

Training models for road scene understanding with automated ground truth

Dan Levi

With: Noa Garnett, Ethan Fetaya, Shai Silberstein, Rafi Cohen, Shaul Oron, Uri Verner, Ariel Ayash, Kobi Horn, Vlad Golder



Agenda

- Road scene understanding
- Acquiring training data with automated ground truth (AGT)
- Test cases:
 - General obstacle detection & classification
 - Car pose estimation
 - Freespace
 - Road segmentation
- Conclusion

On-board road scene understanding



Static:

- Road edge
- Road markings, complex lane understanding
- Signs
- Obstacles: clutter, construction zone cones

Dynamic:

- Classified objects (cars, pedestrians, bicycles, animals ...)
- General obstacles: animals, carts

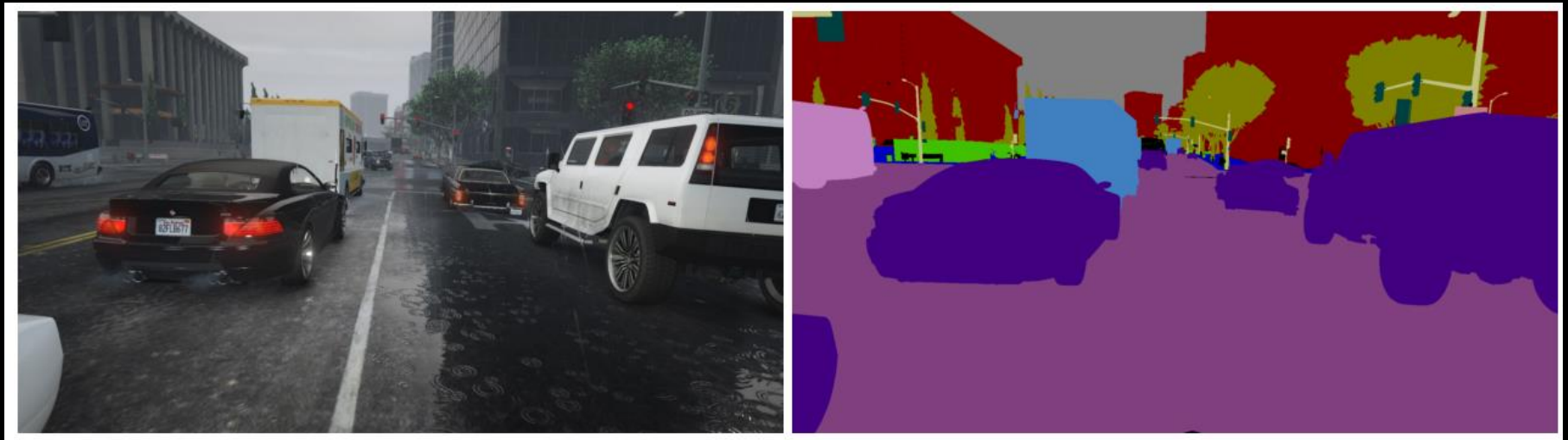
Manual Annotation



The Cityscapes Dataset for Semantic Urban Scene Understanding
[Cordts et al. 2016]

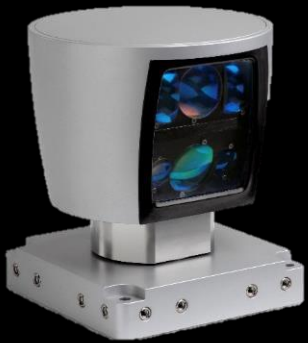
- Time: ~60 min per image
- ~1000 annotators

Computer graphics simulated data



- Photo-realism
- Scenario generation

Automated ground truth(AGT) / Cross-sensor learning



Velodyne LIDAR



AGT for road scene understanding – general setup

“Supervising” sensors



“Target” sensors

Perquisite: Full alignment and synchronization between sensors

AGT for road scene understanding: scheme

“Supervising”
sensors:



“Target”
sensors:



Task: object detection



Data

AGT:

1. Compute **Task** on
Supervising
sensors:

- Offline
- Temporal

AGT:

2. Project
output to target
sensor domain

Ground truth

Automated ground truth / Cross-sensor learning

1. Solve an “easier” problem

- Run time
- Completeness

2. Promise

- Scalability
- Continuous (un-bounded) improvement

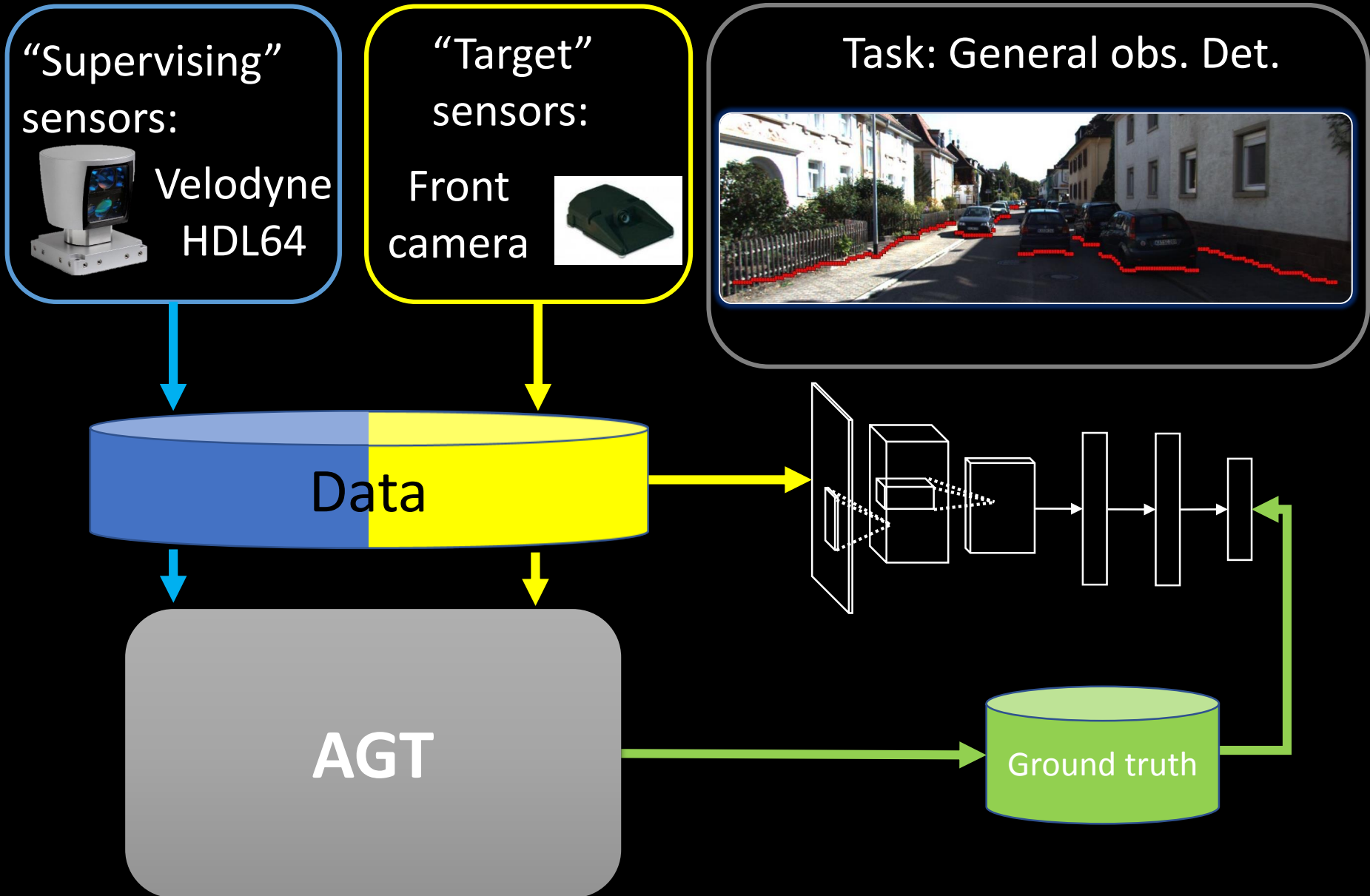
1. Challenging setup

2. Annotation quality / accuracy

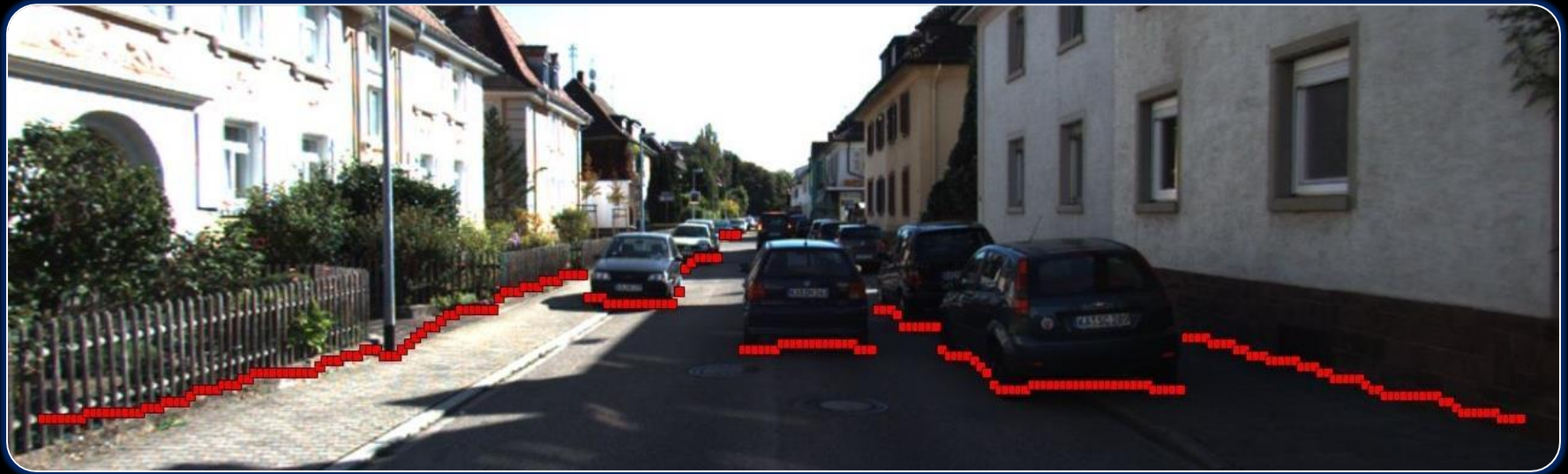
3. Inherent limitations of “supervisor”:

- Learning beyond supervisor capabilities
- Learning from the same sensor (bootstrapping)

AGT for General obstacle detection



StixelNet: Monocular obstacle detection

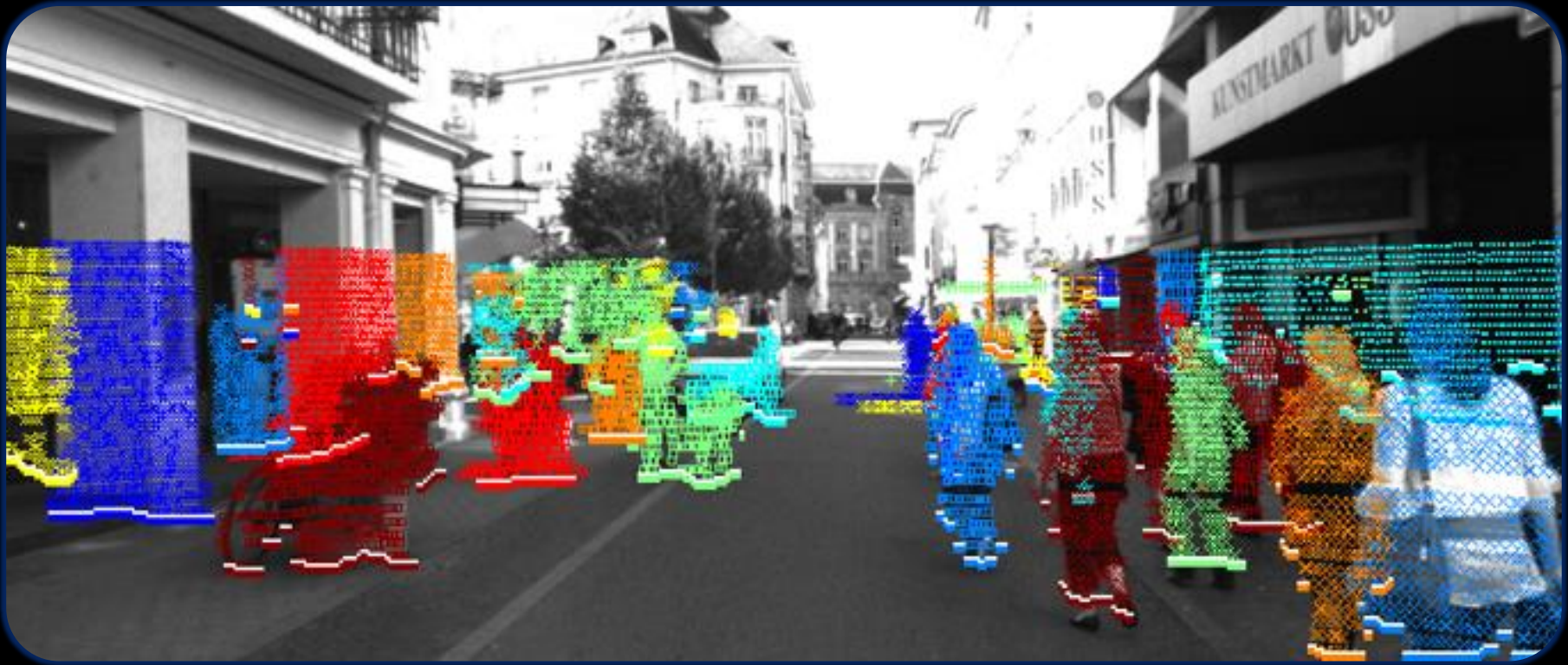


Levi, Dan, Noa Garnett, Ethan Fetaya. **StixelNet : A Deep Convolutional Network for Obstacle Detection and Road Segmentation**. In *BMVC* 2015.

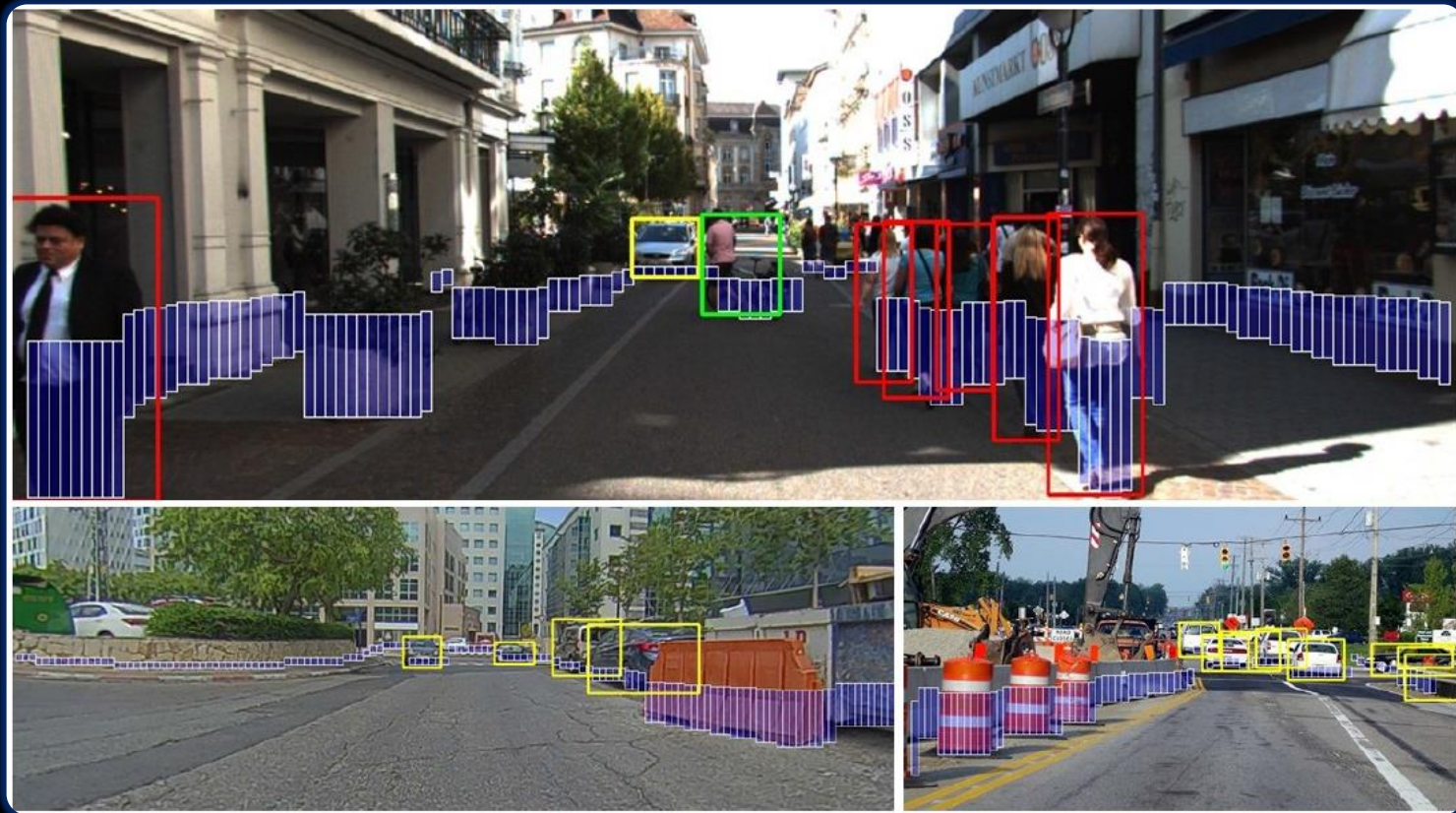
Limitations:

- Cannot handle: close obstacles, “clear” columns
- Low coverage (~30%)

Object-centric obstacle detection AGT

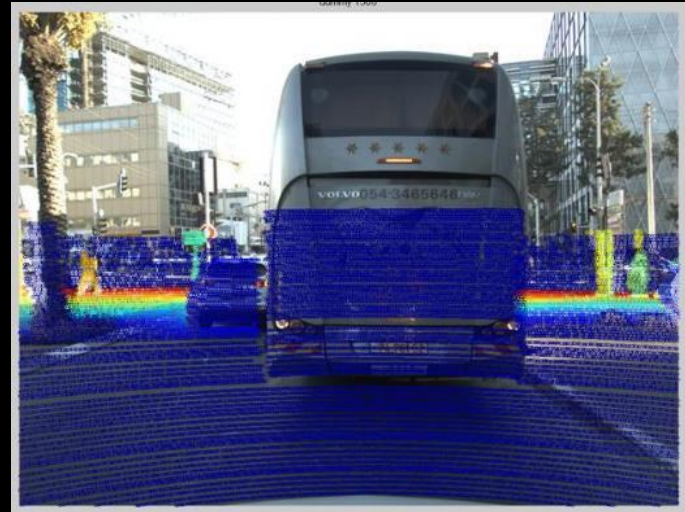


Unified network: StixelNet + Object detection + Object pose estimation



Noa Garnett, Shai Silberstein, Shaul Oron, Ethan Fetaya, Uri Verner, Ariel Ayash, Vlad Goldner, Rafi Cohen, Kobi Horn, Dan Levi. **Real-time category-based and general obstacle detection for autonomous driving. CVRSUAD Workshop, ICCV2017.**

New general obstacle dataset with fisheye lens camera

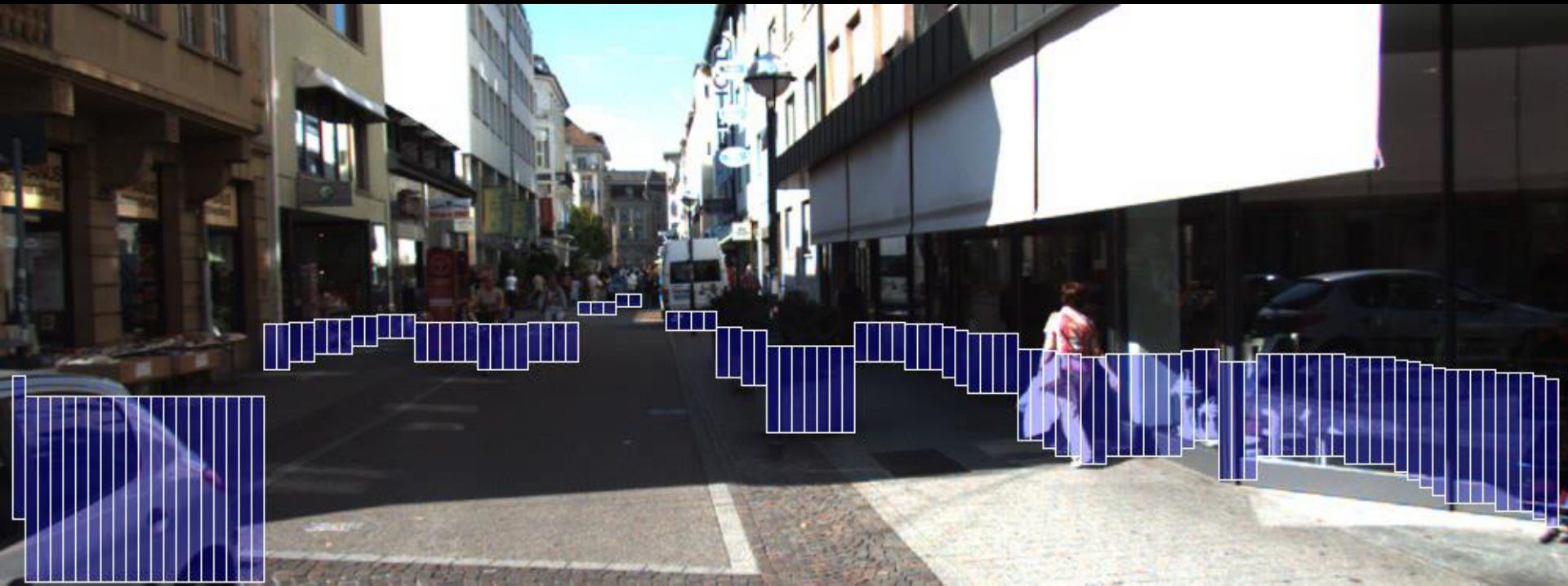


	#images	#instances (columns)
Kitti--train	6K	5M
Internal- train	16K	20M
Kitti-test	760	11K
Internal-test	910	19K

The diagram illustrates the StixelNet architecture, which is designed for detecting and tracking objects in video sequences. It consists of three main stages:

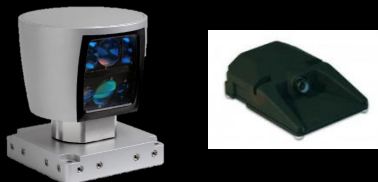
- Input Image:** The process starts with an input image of size 370x800x3.
- GoogleLeNet through inception_4 layer:** The input image is processed by the GoogleLeNet architecture, specifically through the inception_3 and inception_4 layers. The output of the inception_4 layer is a feature map of size 46x100x480.
- SSD + Pose Layers:** The feature map is then processed by the SSD + Pose Layers. This stage involves several convolutional layers (Conv) and pooling operations. The output of this stage is a set of SSD Detections, confidence, and pose information, which is then used for Non-maximal suppression.
- StixelNet Layers:** The output from the SSD + Pose Layers is fed into the StixelNet Layers. This stage involves a series of convolutional layers (Conv) and pooling operations, including vertical pooling. The final output is a set of StixelNet Detections, confidence, and pose information, which is then used for Non-maximal suppression.

Results on KITTI



AGT for Obstacle classification

“Supervising”
sensors:

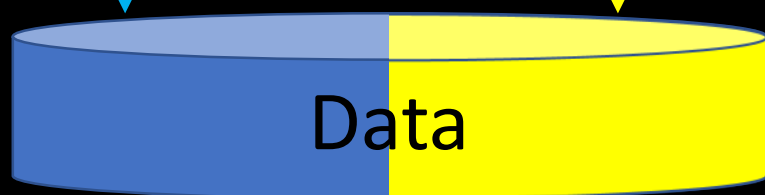
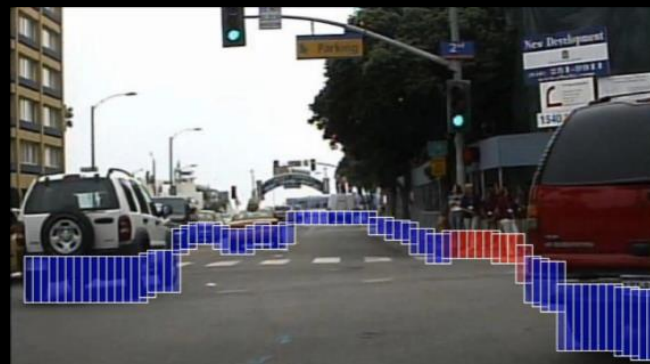


“Target”
sensors:

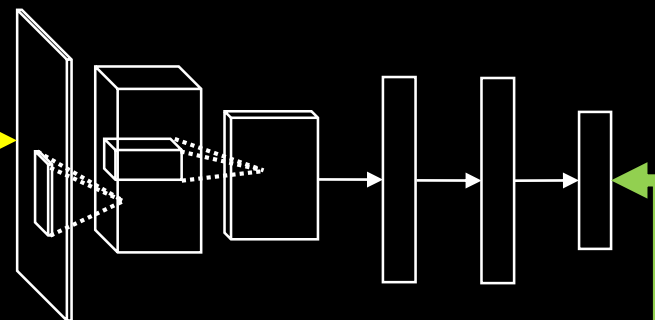
Front
camera



Task: Obstacle classification



AGT



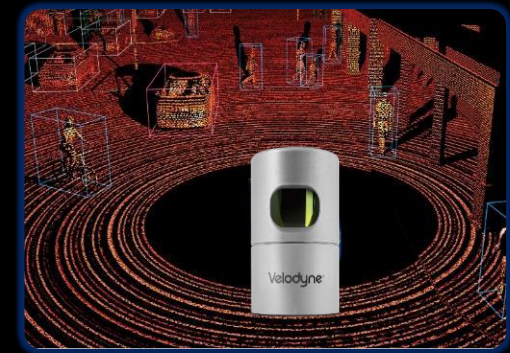
Ground truth

AGT for obstacle classification

Image based detection

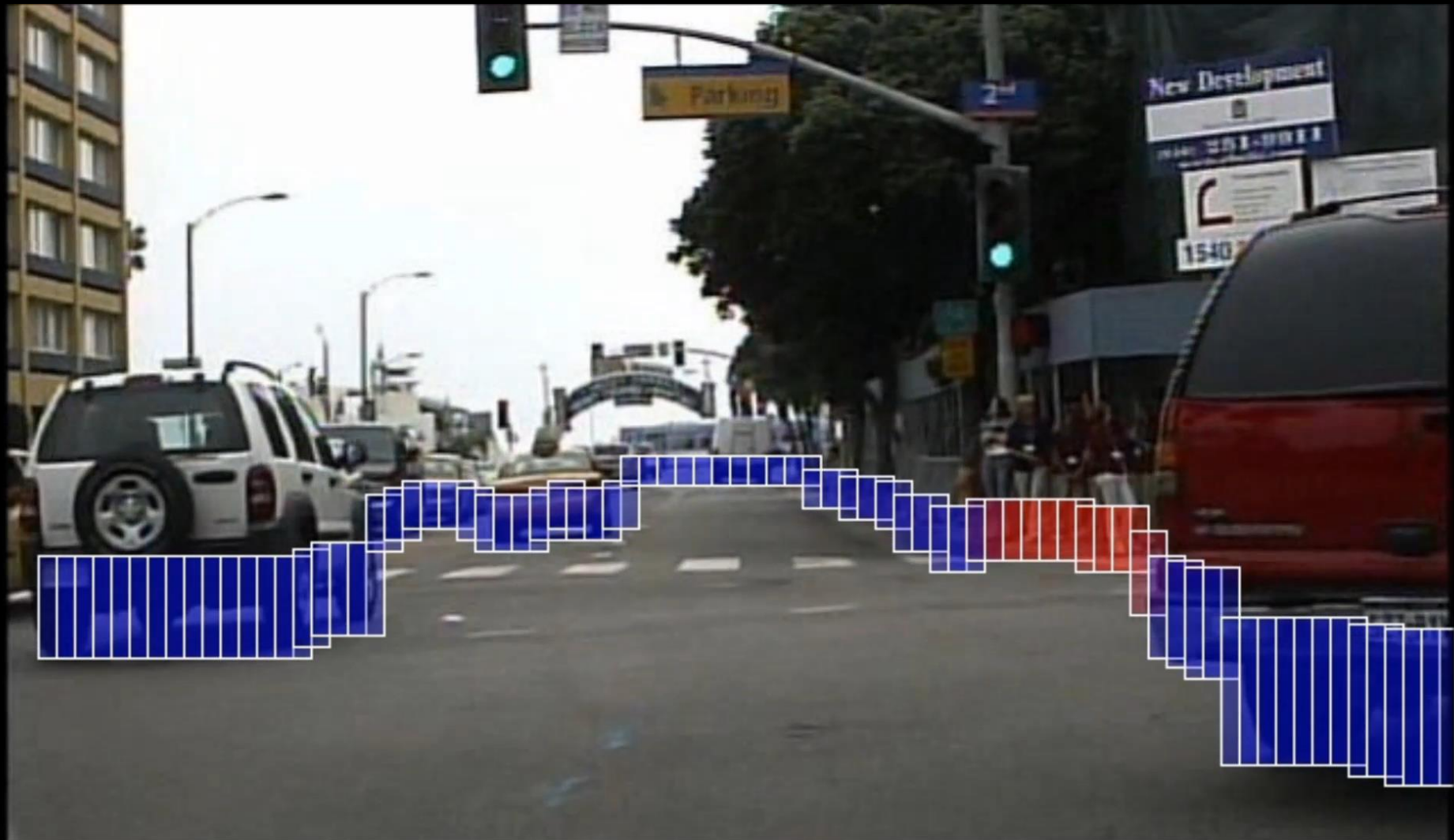


Lidar based verification

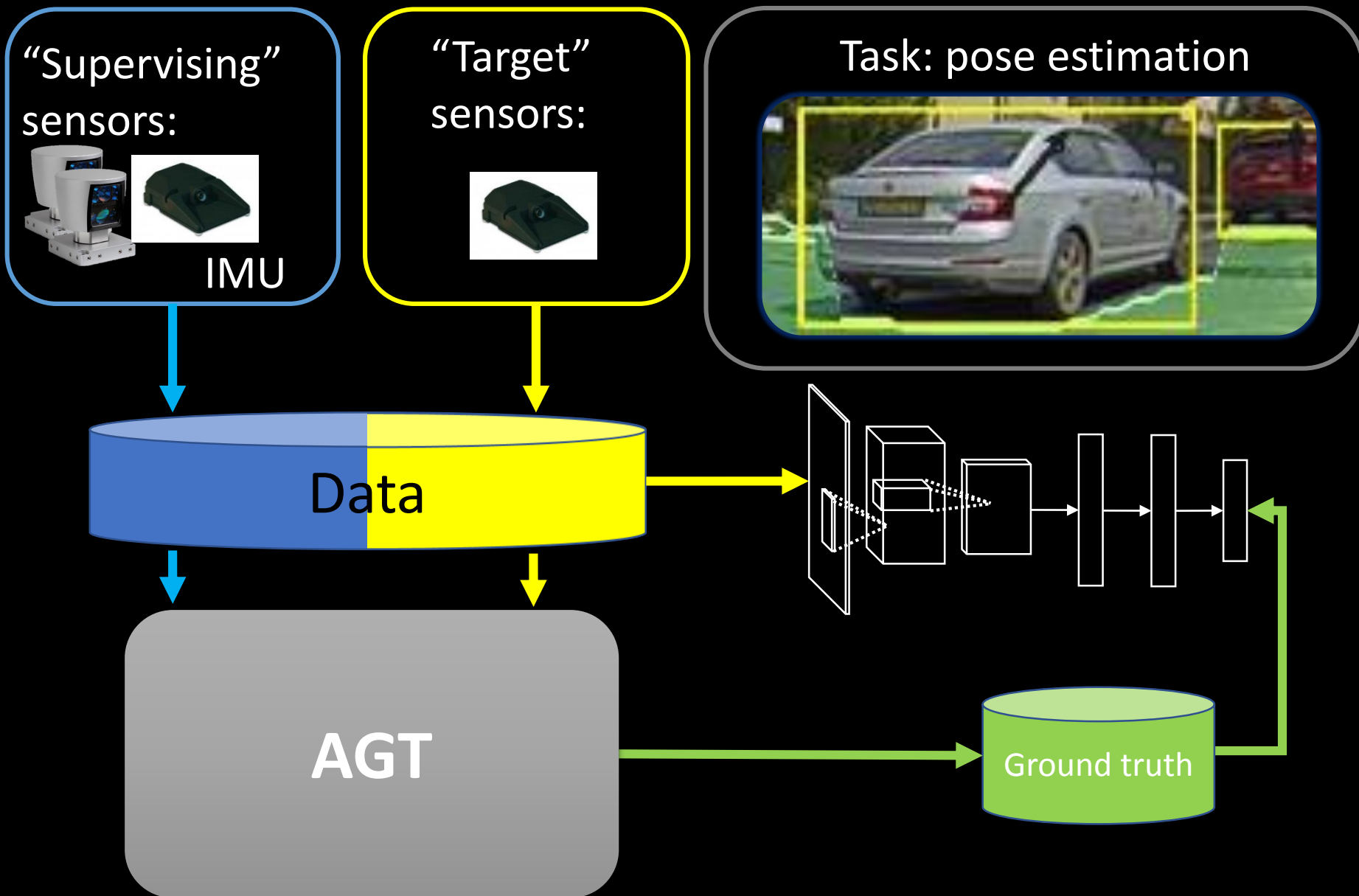


Source: <http://self-driving-future.com/the-eyes/velodyne/>

Obstacle classification trained net result: pedestrians

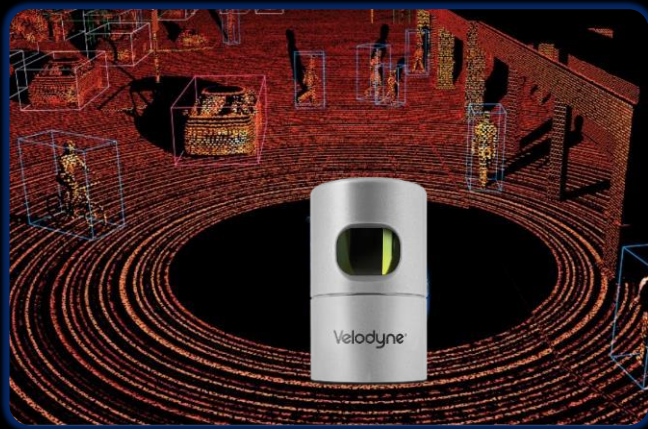


AGT for car pose estimation



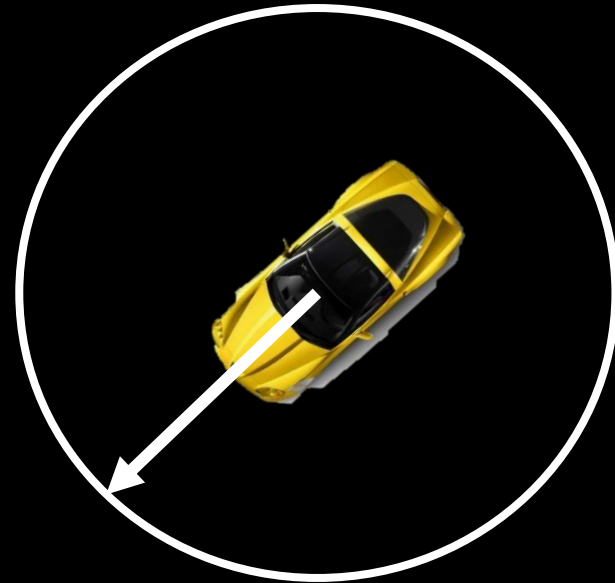
AGT for pose estimation

Multi sensor,
temporal object
detection



Source: <http://self-driving-future.com/the-eyes/velodyne/>

8 orientation bins pose representation



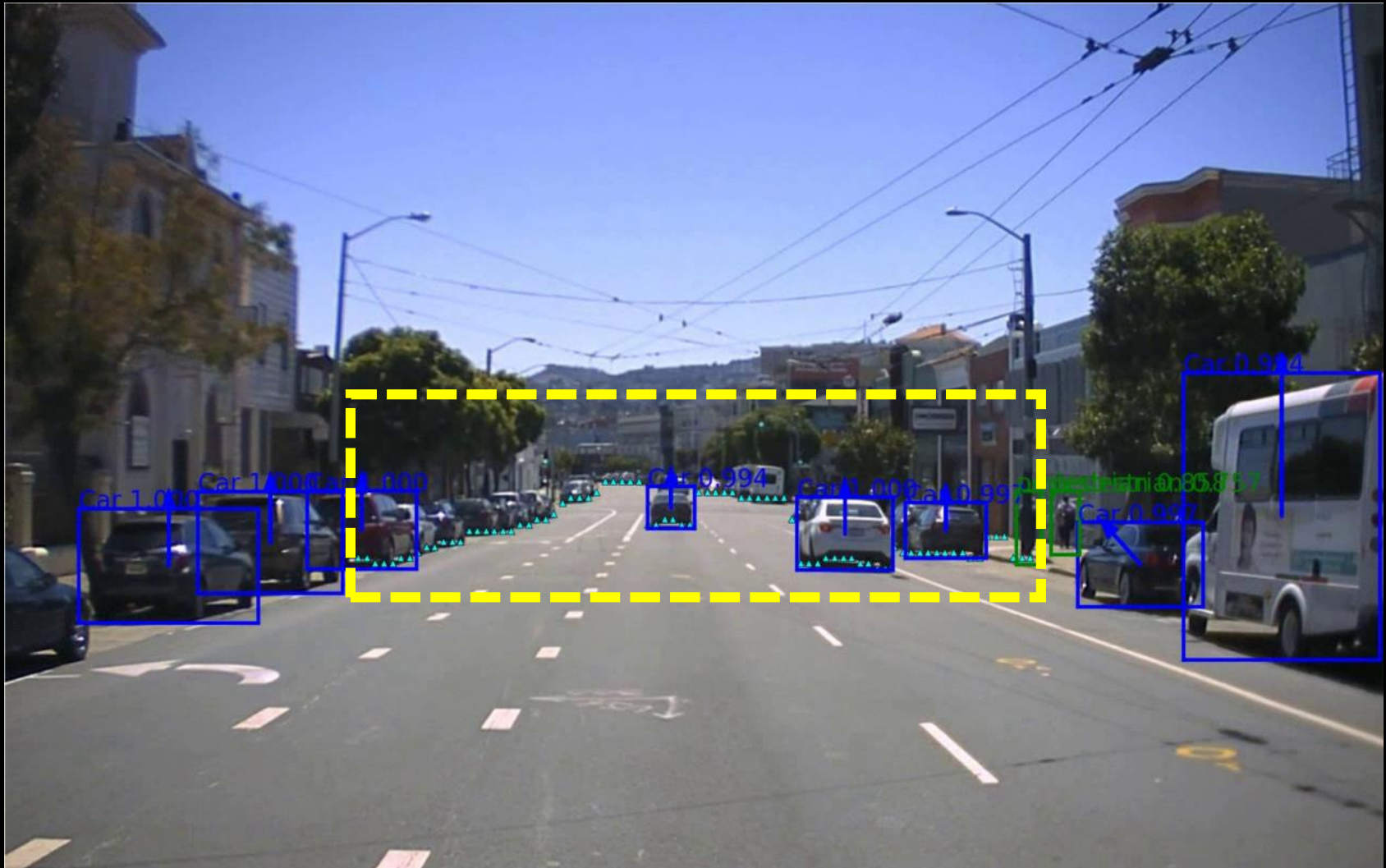
Dynamic → Static

Pose estimation



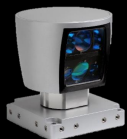
trained with mixed AGT and Manual

Far range general obstacle detection



AGT for freespace

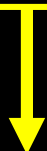
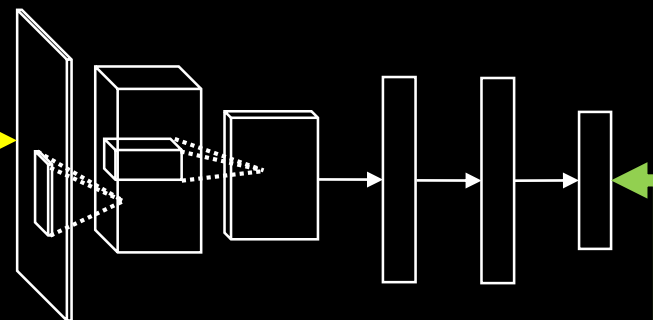
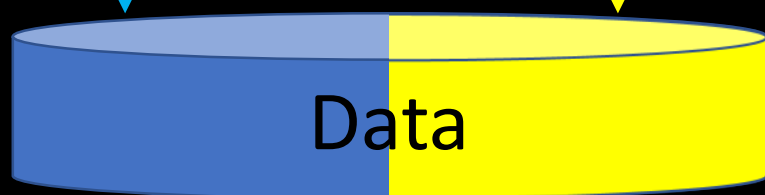
“Supervising”
sensors:



“Target”
sensors:



Task: freespace



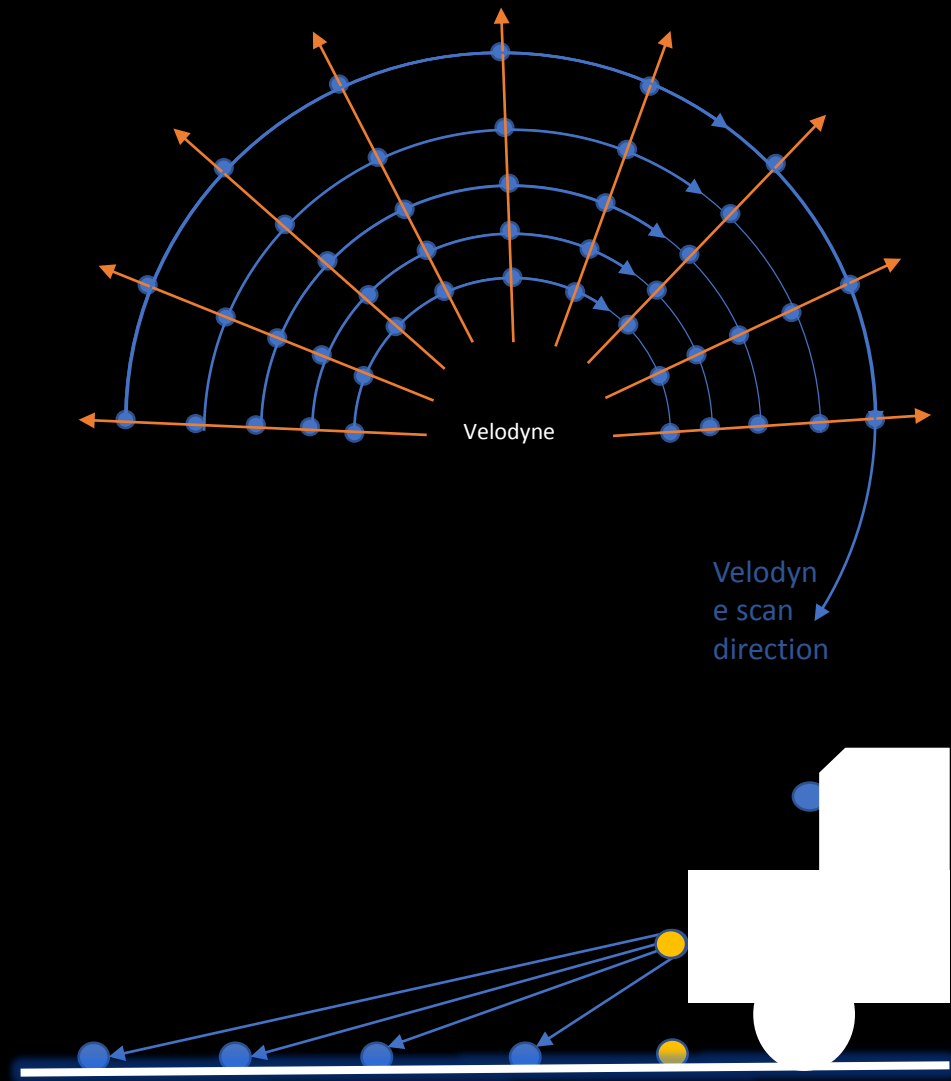
AGT for freespace with 3D beams

Estimate and subtract
road plane

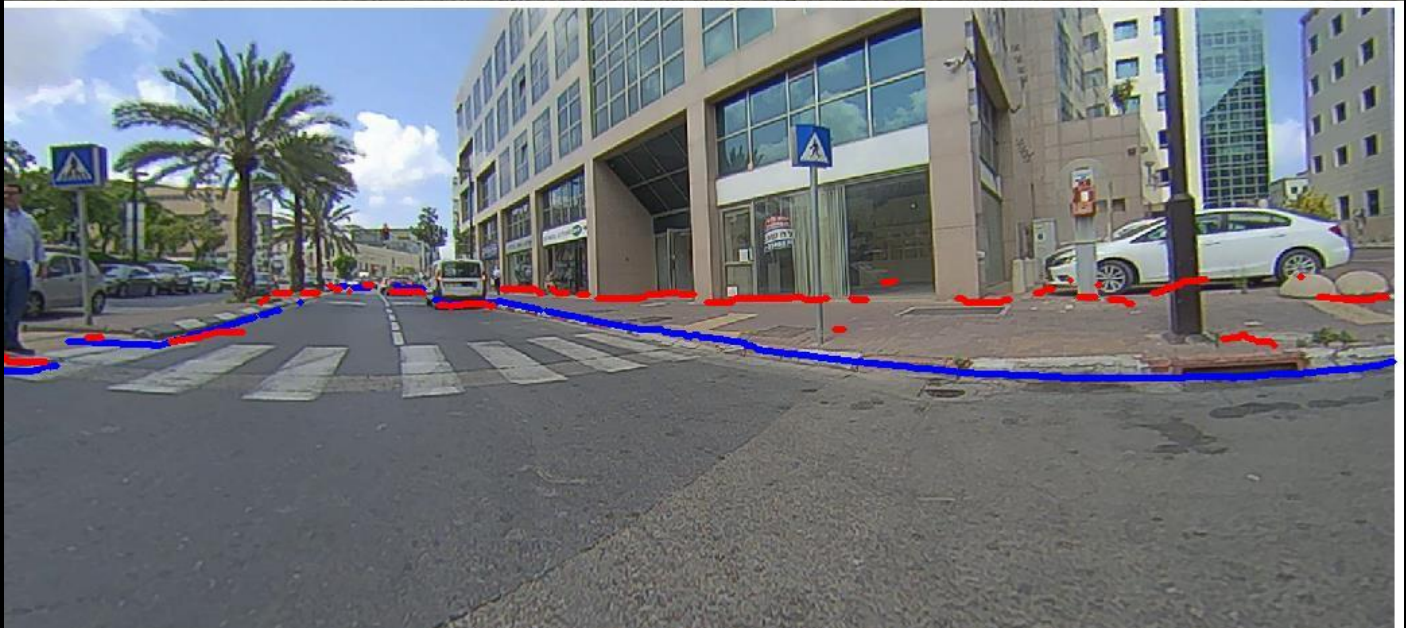
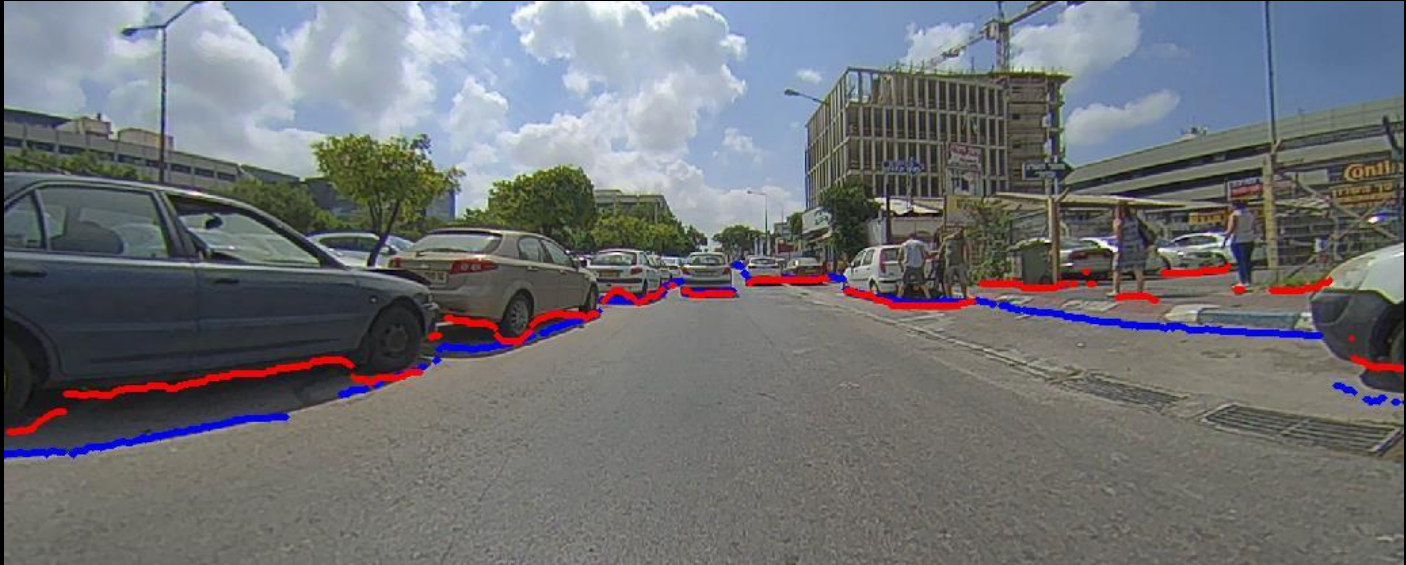
Analyze single Lidar
"Beam"

Project limit to *ground
plane*

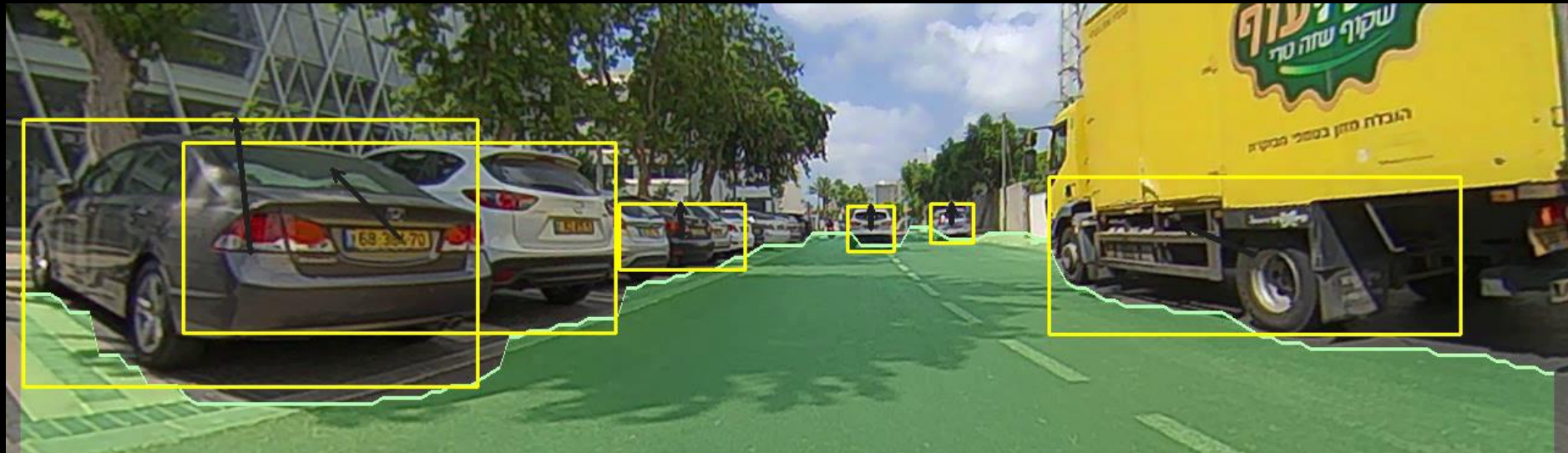
Project freespace limit
to *image plane*, find
"near" and "clear"



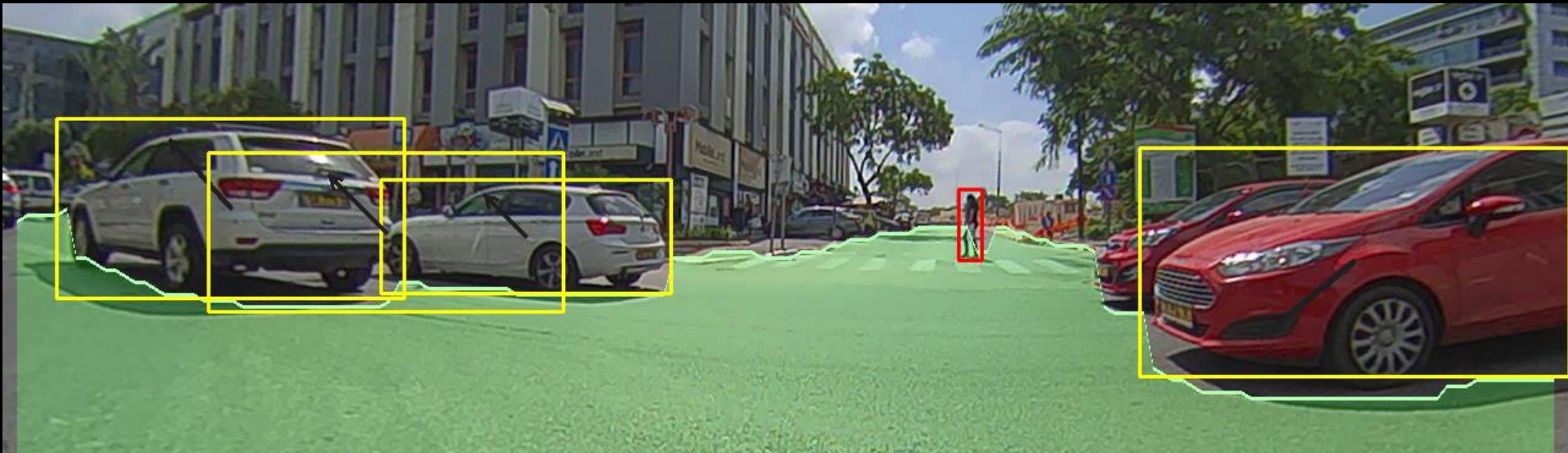
Obstacles vs. Freespace AGT



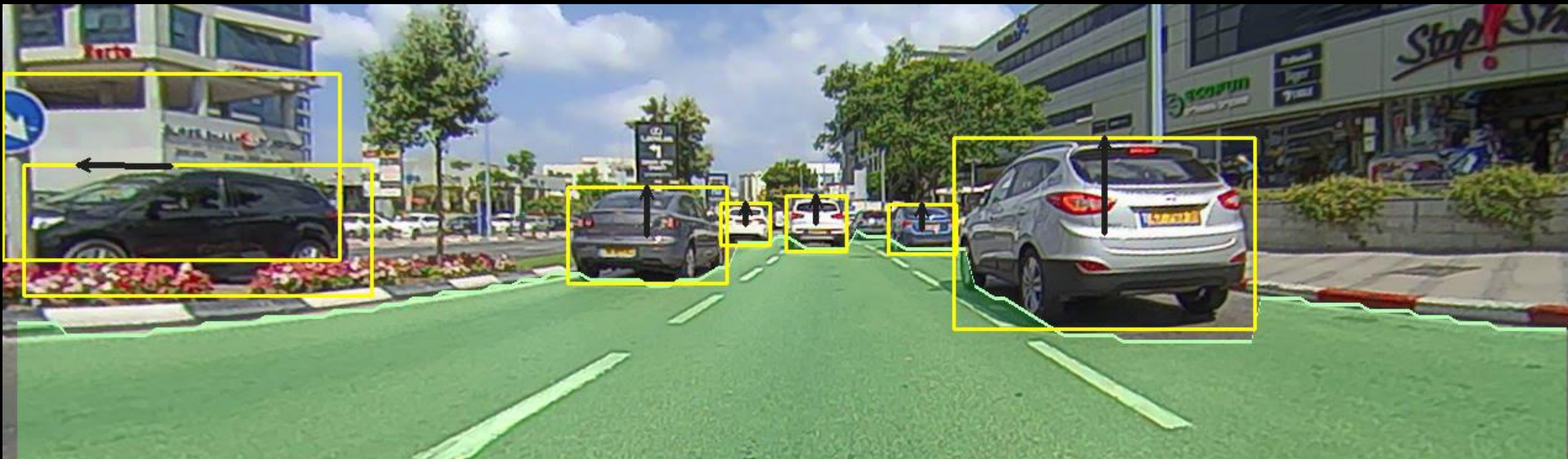
Freespace + object detection + car 3D pose



Freespace + object detection + car 3D pose



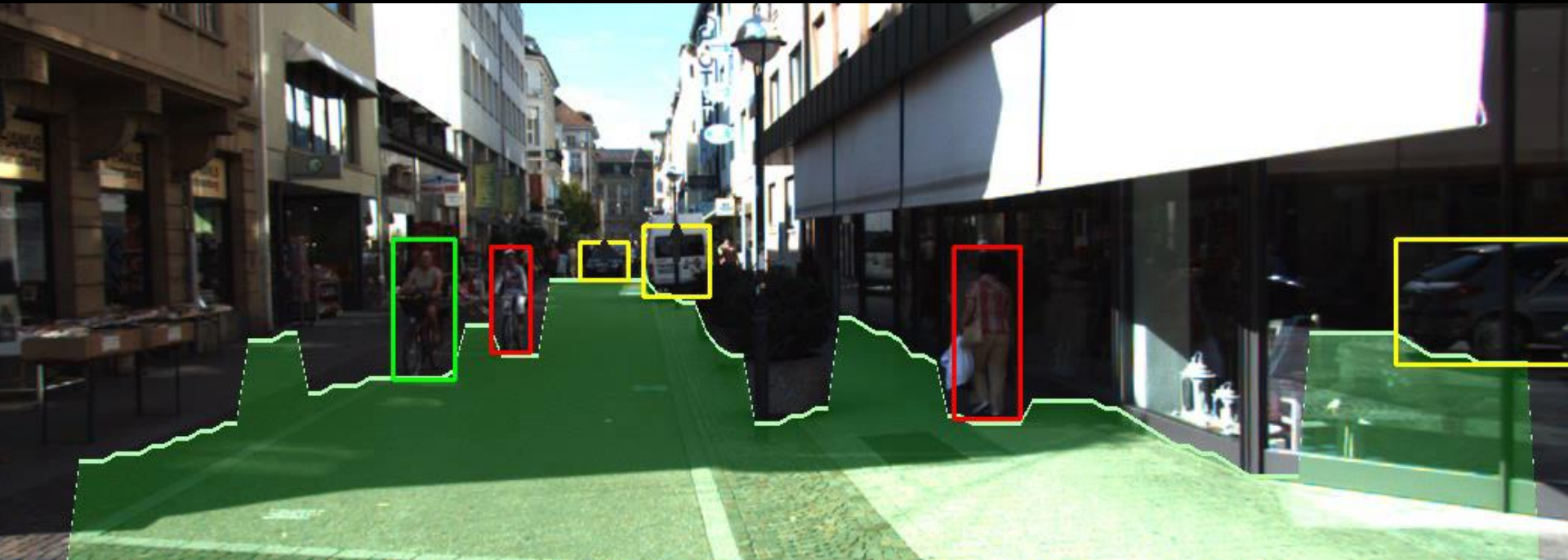
Freespace + object detection + car 3D pose



Freespace + object detection + car 3D pose



Freespace + object detection + car 3D pose

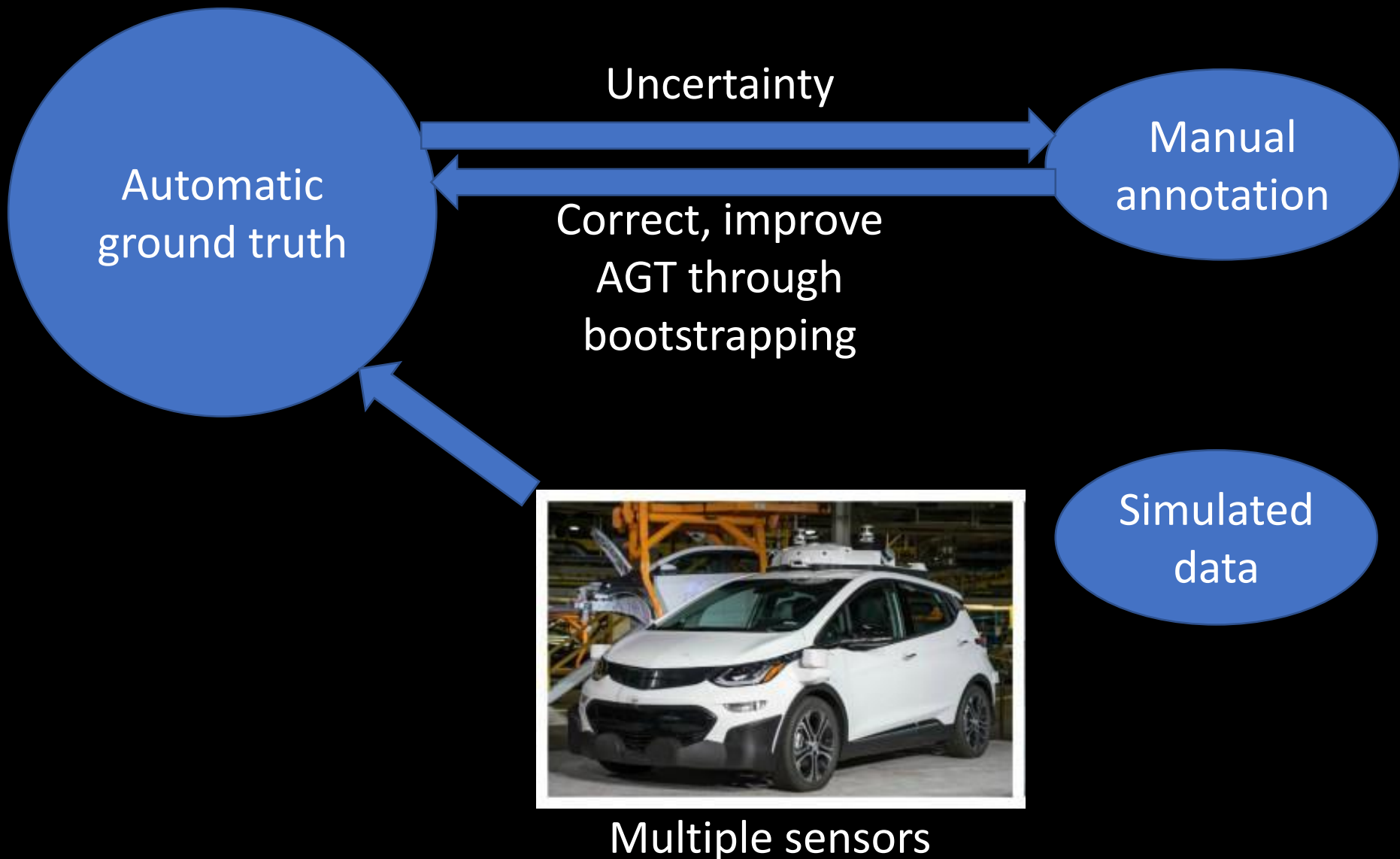


Finetuning from AGT: road segmentation



1. Fine-tune on KITTI Road segmentation (manually labelled)
2. Graph-cut segmentation
3. State-of-the-art accuracy among non-anonymous (94.88% MaxF)

What's next?



Thank you!