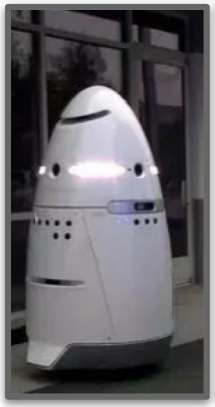
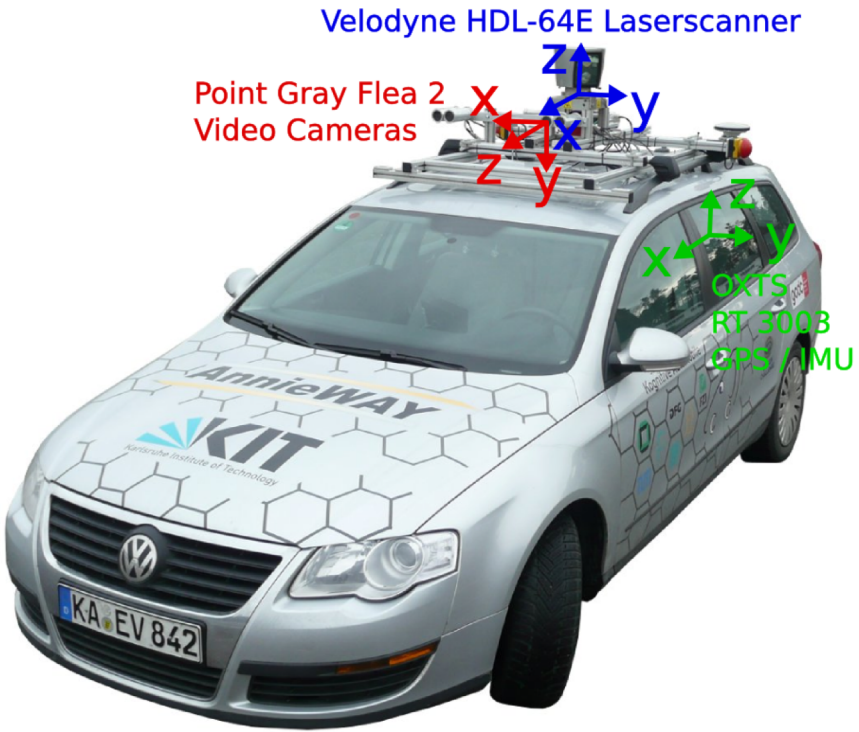
3D Multi-Object Tracking and Beyond

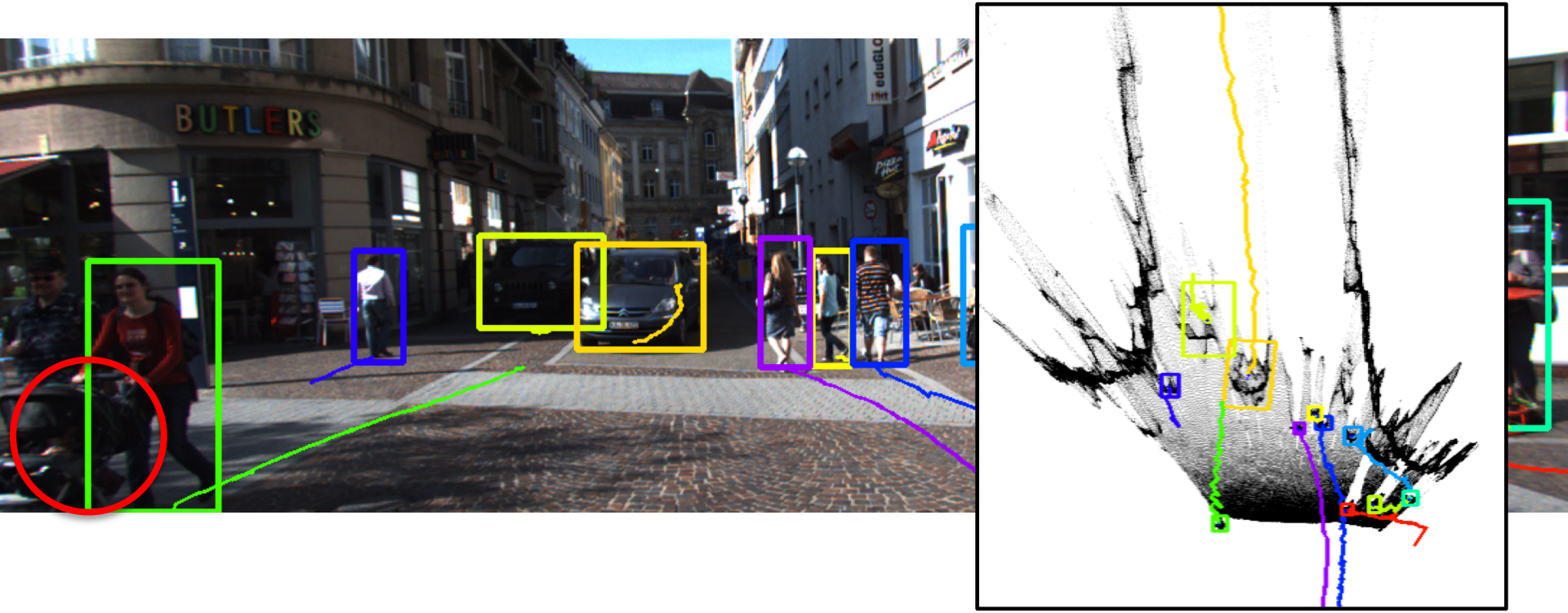
Motivation



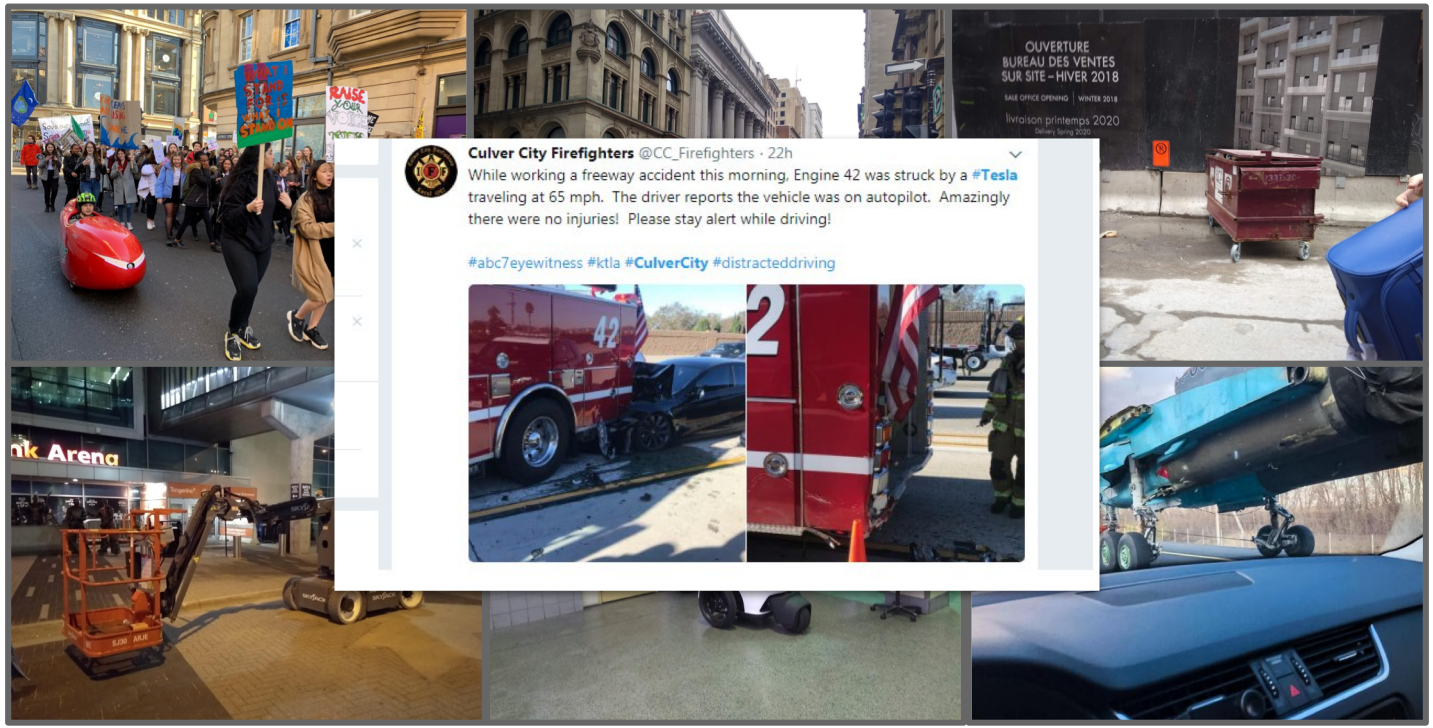


Geiger et al, CVPR'14

Motivation

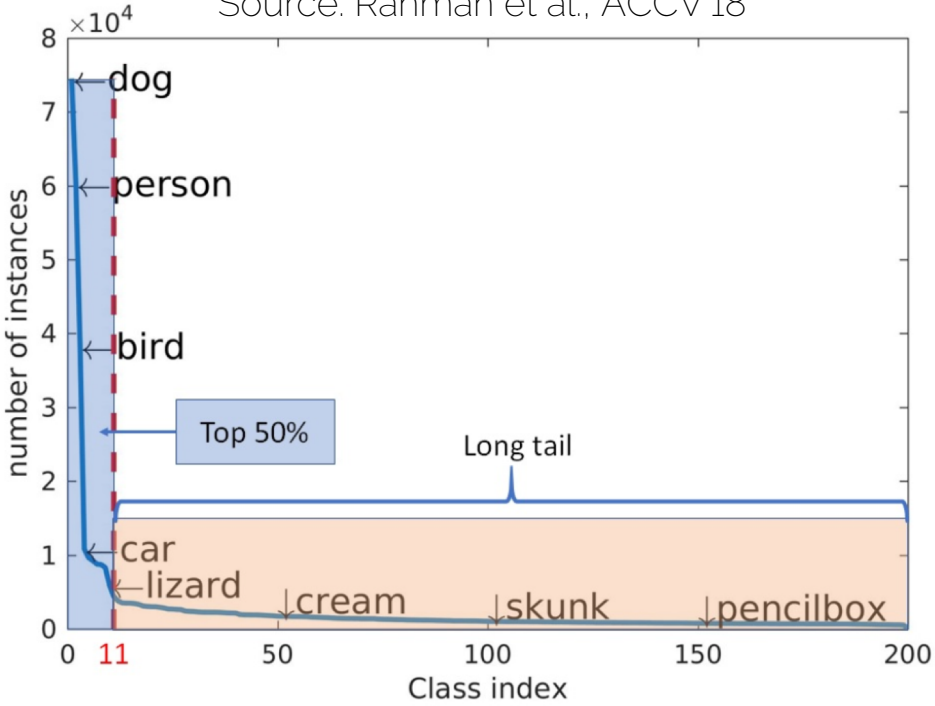


But How About These?

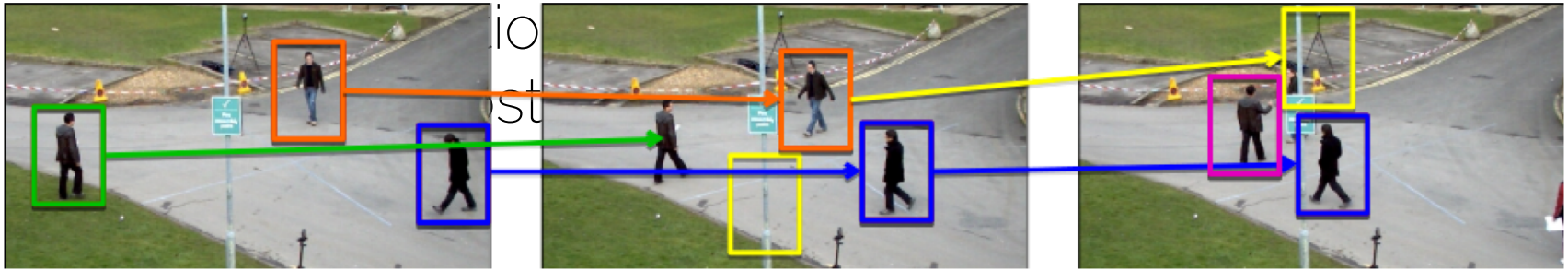


The Long Tail

Source: Rahman et al., ACCV'18



Reminder: Vision-based MOT



- IoU
- ...

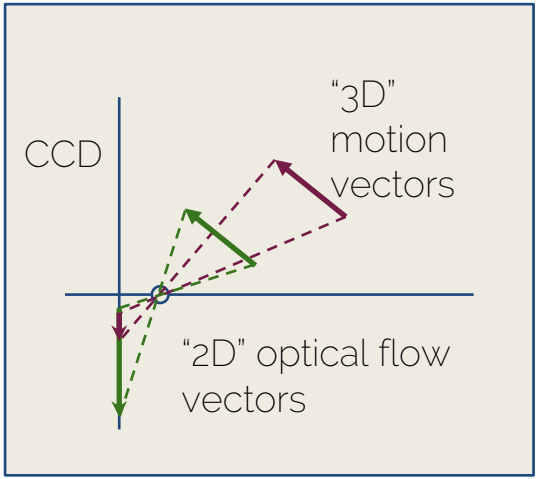
Predictions

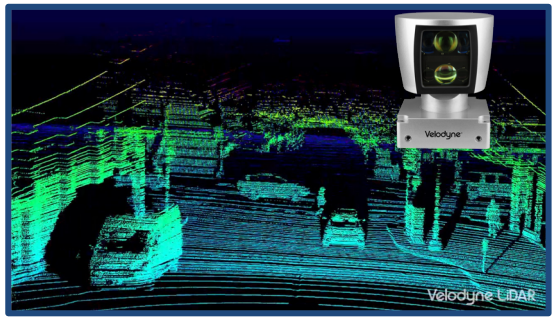
Detections

0.9	0.8	0.8	0.3	0.3
0.5	0.4	0.7	0.3	0.3
0.2	0.1	0.4	0.3	0.3
0.1	0.2	0.5	0.3	0.3
0.3	0.3	0.3	0.3	0.3

3D MOT: Advantages

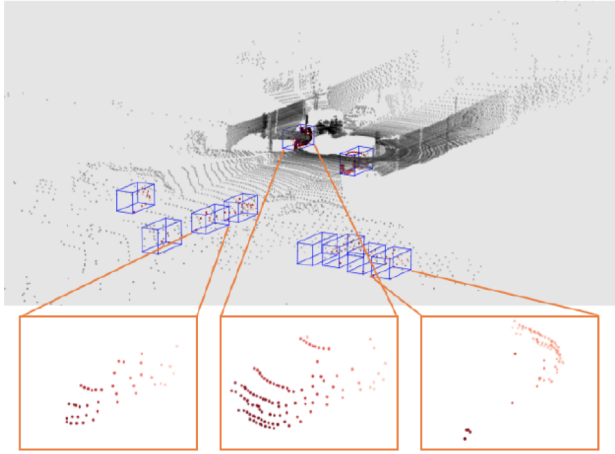
- No distortion due to projection
  - Object velocity vs. "apparent" velocity
  - Less sensitivity to illumination changes
- Incorporate geometric constraints
  - In 2020, cars don't fly …
  - 3D size of target classes, max velocity
- Additional sources of information
  - Stereo, RGB-D cameras
  - LiDAR



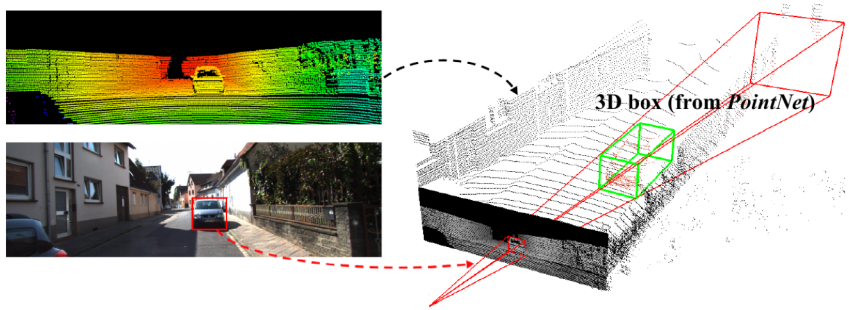


Challenges

- Depth sensor characteristics
  - Limited scan range
  - Reflective surfaces
  - Resolution decreasing with distance
- Mobile platform
- Precise localization of objects
  - 3D bounding box:
    - position,
    - size,
    - orientation



Source: Yuan et al., 3DV'19

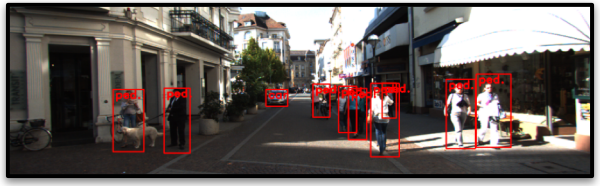


Source: Qi et al., CVPR'18

3D Tracking-by-Detection

CIWT: Stereo-Vision Based 3D MOT

- Input: stereo images
- Object detections
  - 2013 - 2016 rapid progress in the field of (image-based) object detection (R-CNN family)



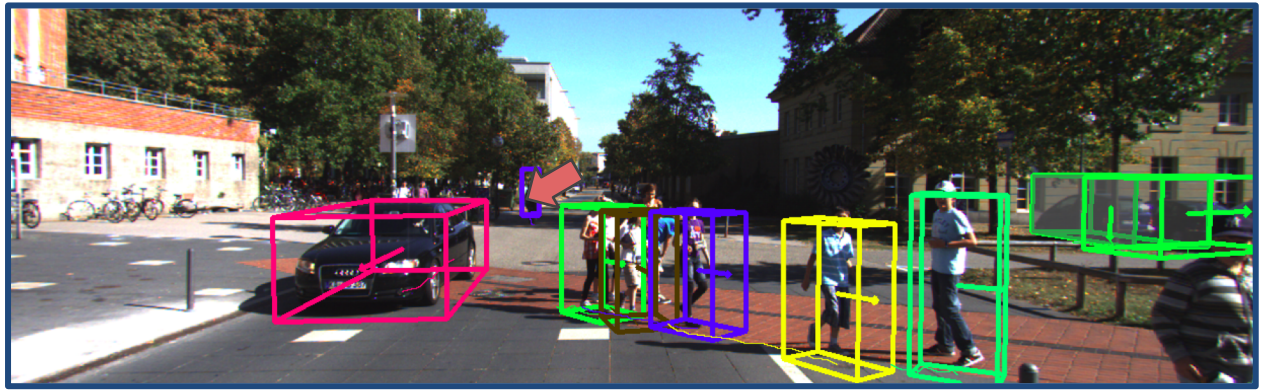
- Goal: 2D MOT, but:
  - Utilize stereo
  - Infer 3D trajectories of objects

Osep et. al., Combined Image- and World-Space Tracking, ICRA'17

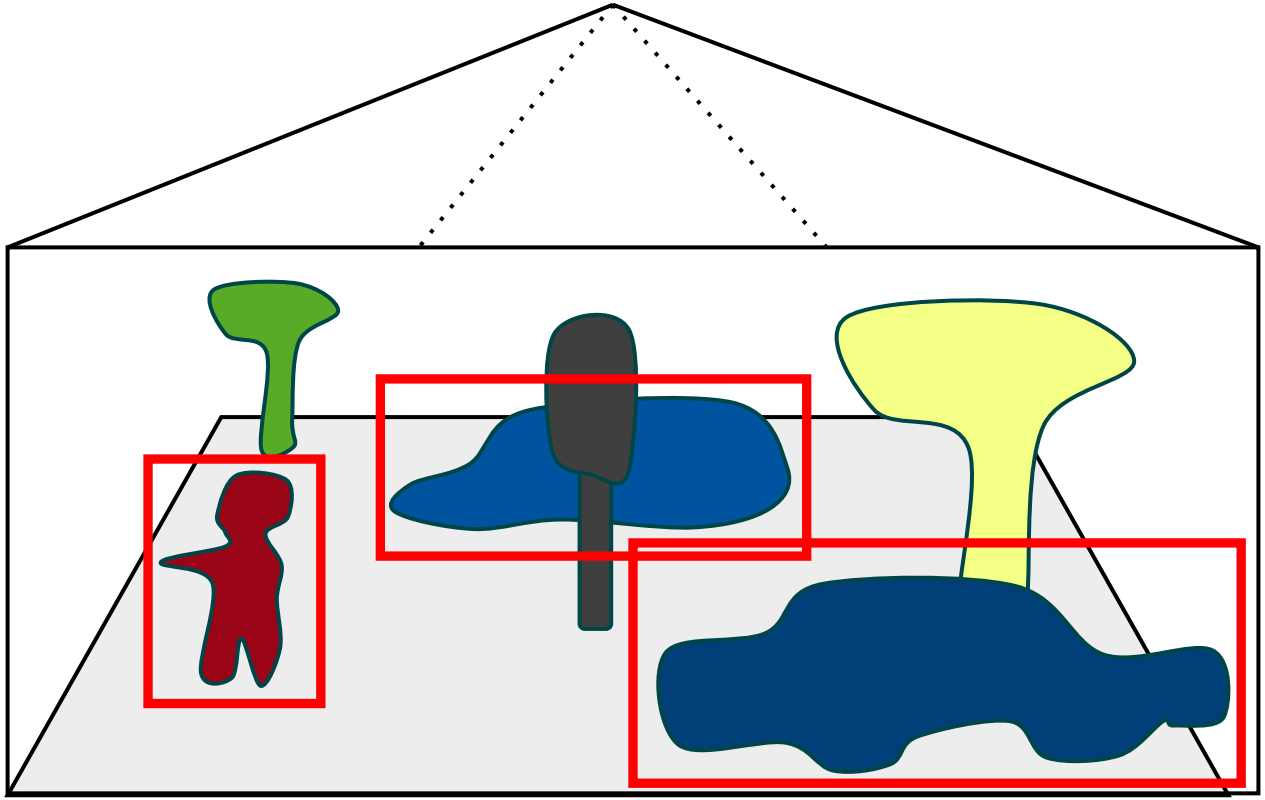
CIWT: Stereo-Vision Based 3D MOT

- Objects should be localized in 3D space precisely
- Objects should be tracked even when far away from camera
  - "reliable depth": 40xbaseline (KITTI: ~20m)
  - OK to be "a bit off" when object is >100m away …
  - But we need to see them coming!

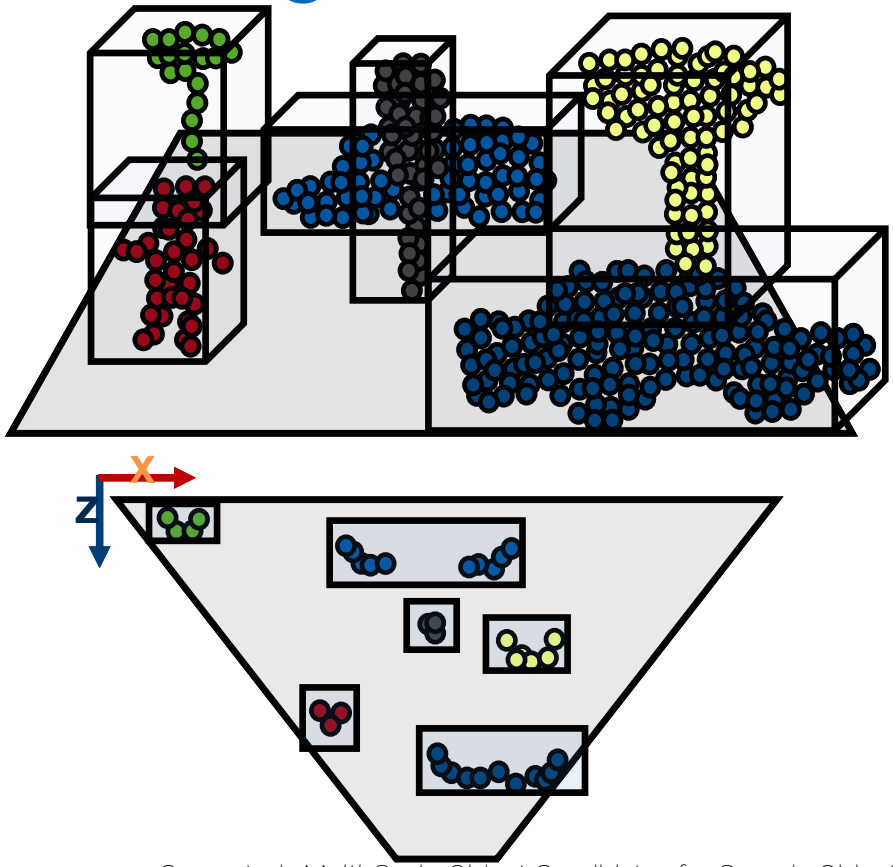
Paz et al., IEEE Transactions on Robotics'08



CIWT - 3D Localization



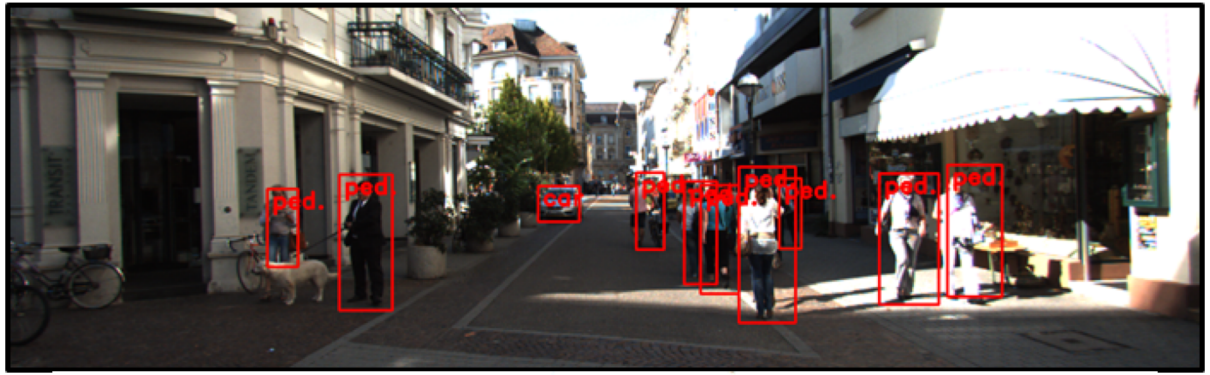
CIWT - 3D Localization



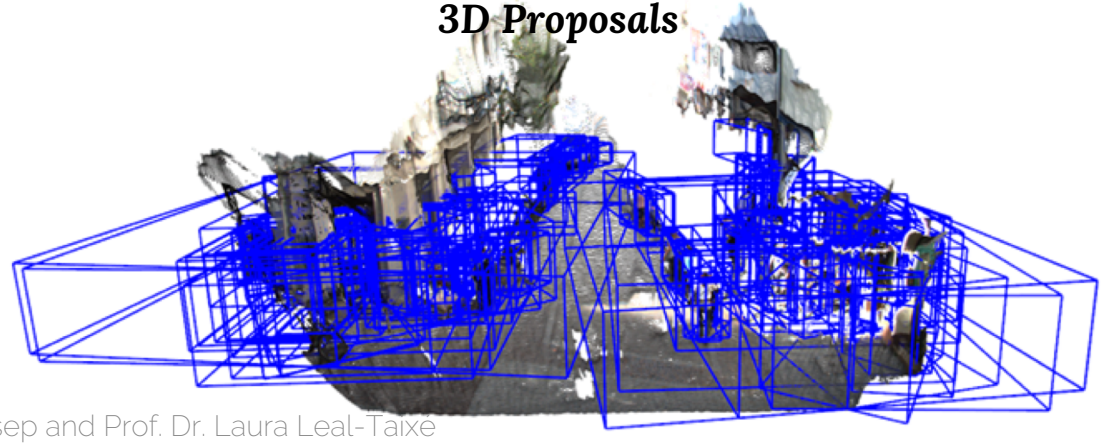
Osep et. al., Multi-Scale Object Candidates for Generic Object Tracking in Street Scenes. ICRA'16

CIWT - 3D Localization

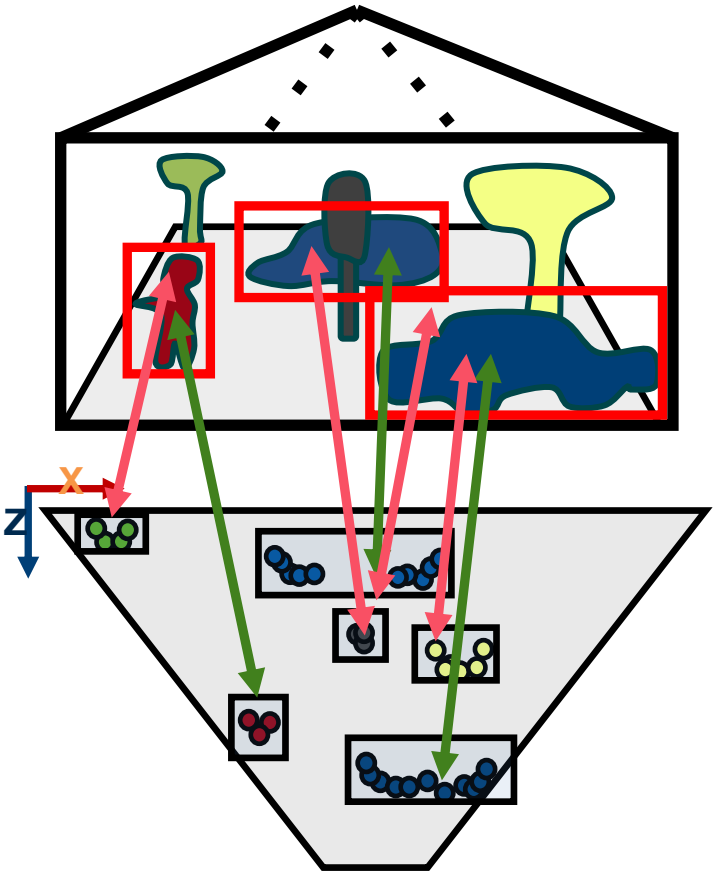
Detections



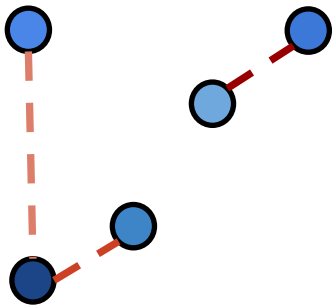
3D Proposals

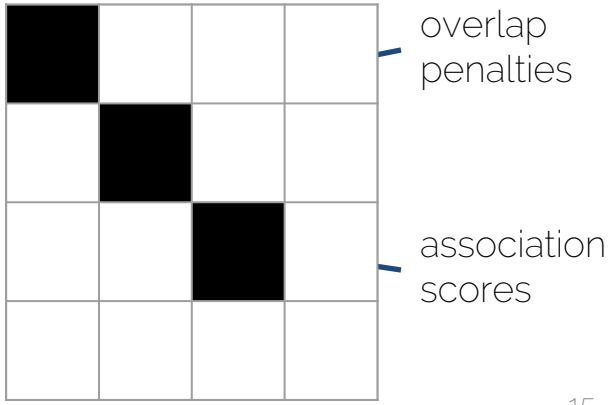


CIWT - 3D Localization



QPBO





CIWT - 3D Localization

CIWT - 3D Localization

- Why object proposals?

What do we do with these?
Stay tuned ...

(a) Reference image.

(b) Stereo - boxes.

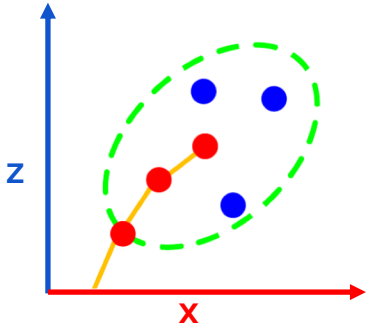
(c) Stereo - proposals.

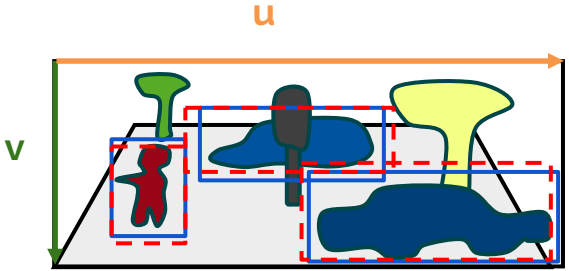
CIWT - Tracking

- We localized objects in 3D space!
- Now:
  - Associate (3D-localized) detections to tracks
  - Estimate track states
- Intuition
  - Image - we can localize bounding boxes well
  - 3D - we can localize object centers well
  - 3D bounding box IoU won't do -- **why?**
- Association costs
  - How well predicted 3D positions match observations?
  - How well redicted (image) boxes match observations?

CIWT - Tracking

- Association costs
  - How well predicted 3D positions match observations?
  - How well redicted (image) boxes match observations?





- How to obtain predictions?
  - Need to know how these things move!

Tracking with Dynamics

- Key idea
  - Given a model of expected motion, predict where objects will occur in next frame, even before seeing the image

- Goals
  - Restrict search for the object
  - Improved estimates since measurement noise is reduced by trajectory
  - Smoothness

Reminder: MOT

- State vector
  $\mathbf{x} = [x, y, v_x, v_y]^T$
- As the track evolves:
  $\mathbf{x}^k = [x^k, y^k, v_x^k, v_y^k]^T$
- Observations (set of measurements):
  $\mathbf{Y}^k = \{\mathbf{y}_1, \ldots, \mathbf{y}_{M_k}\}$
- Data association
  $\{\mathbf{x}_1, \ldots, \mathbf{x}_{N_k}\}, \{\mathbf{y}_1, \ldots, \mathbf{y}_{M_k}\}$

Reminder: (Extended) Kalman Filter

- Prediction

$$\hat{\mathbf{x}}^k = f(\mathbf{x}^{k-1}, \boxed{\mathbf{u}^k})$$

$$\hat{\mathbf{\Sigma}}_k = \mathbf{F}^k \mathbf{\Sigma}^{k-1} \mathbf{F}^{k^T} + \mathbf{Q}^k$$

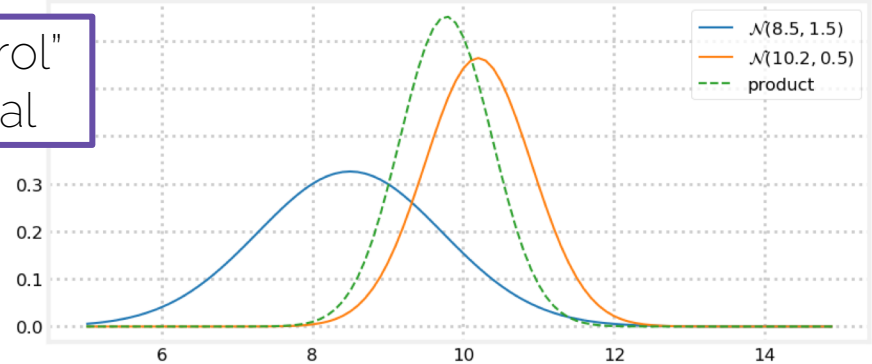
Kalman "gain" ction

$$\mathbf{K}^k = \hat{\mathbf{\Sigma}}^k \mathbf{G}^{t^T} (\mathbf{G}^t \hat{\mathbf{\Sigma}}^k \mathbf{G}^k + \boxed{\mathbf{R}^k})^{-1}$$

$$\mathbf{x}^k = \hat{\mathbf{x}}^k + \boxed{\mathbf{K}^k}(\boxed{\mathbf{z}^k} - g(\hat{\mathbf{x}}^k))$$

$$\mathbf{\Sigma}^k = (\mathbf{I} - \mathbf{K}^k \mathbf{G}^k)\hat{\mathbf{\Sigma}}^k$$

"control" signal

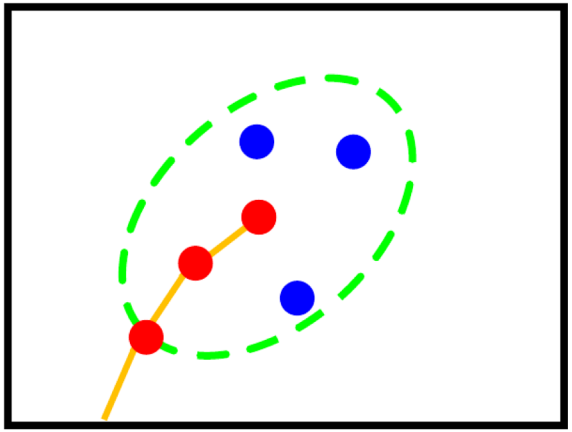


observation uncertainty

measurement

Reminder: MOT

- Kalman filter
  $\hat{\mathbf{x}_\mathbf{l}}^k, \Sigma_l^k$
- Innovation
  $\mathbf{v}_{j,l}^k = (\mathbf{y}_j^k - \mathbf{x}_l^k)$
- Observation likelihood
  $p(\mathbf{y}_j^k | \mathbf{x}_l^k) = \exp({\mathbf{v}_{j,l}^k}^T {\Sigma_l^k}^{-1} \mathbf{v}_{j,l}^k)$ Recognize this?
- Gaiting volume:
  $V^k(\gamma) = \{\mathbf{y} | (\mathbf{y} - \mathbf{x}_l^k)^T {\Sigma_l^k}^{-1} (\mathbf{y} - \mathbf{x}_l^k) \leq \gamma\}$



CIWT

- Challenge: moving robot platform
  - Perceived motion of objects is relative motion
- Our approach
  - Extended Kalman filter for modelling object dynamics
  - Scene geometry -- ground plane
  - Explicitly estimate ego-motion
  - “Tie” image bounding box predictions to the geometry

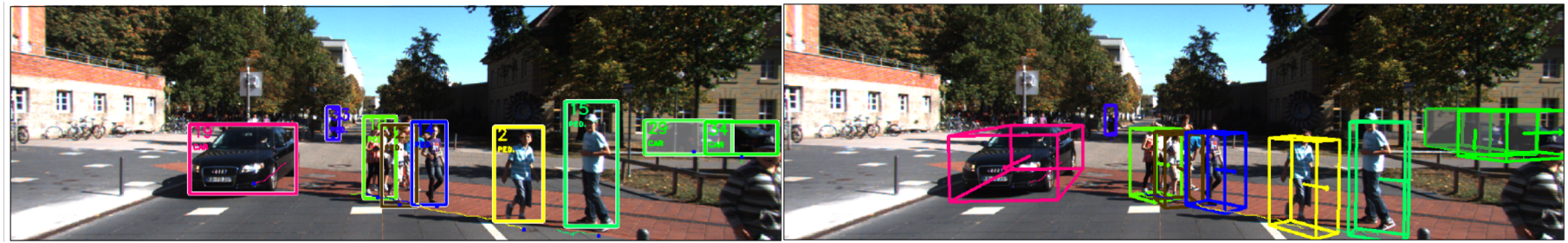
t

t+1

CIWT

- Joint 2D-3D state representation

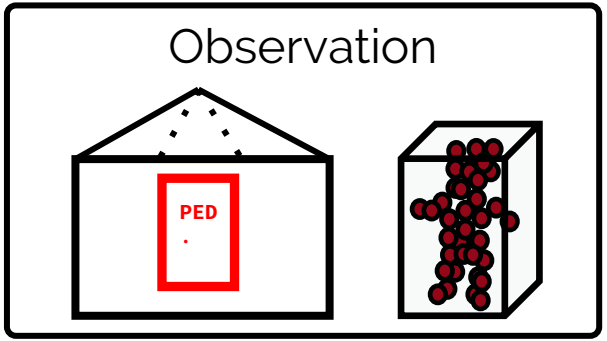
$$\mathbf{x}^k = [\mathbf{B}_{2D}, \dot{\mathbf{B}}_{2D}, x, y, v_x, v_y, w, h, l]$$

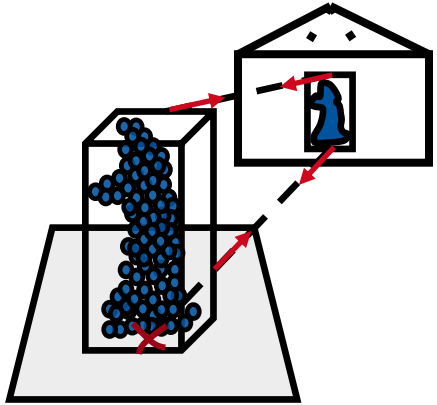


- Why?
  - Need to have a reliable prediction of 2D bounding box
  - Need to have a 3D estimate of the object
  - Keep tracking objects outside of the stereo range

CIWT

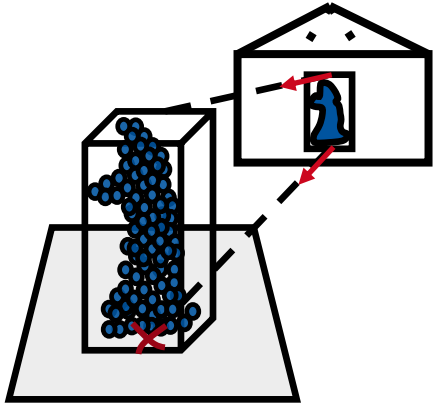
- State transition function $\mathbf{x}^k = f(\mathbf{x}^{k-1}, \mathbf{u}^k)$
  - $\mathbf{u}^k$ is ego-motion estimate, obtained using visual odometry estimator (could also use robot odometry)
  - Weakly couple 2D-3D state (projection-backprojection)
  - Provide 2D bounding box prediction, **tied** to the estimated 3D geometry





CIWT

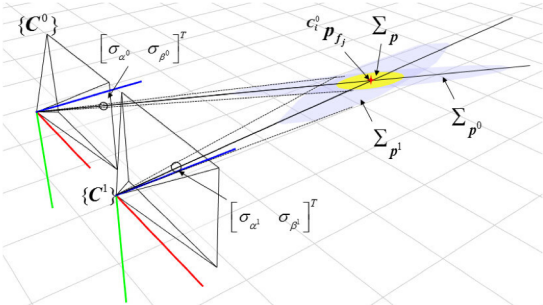
- Observation
  - Opportunistic approach -- update the state with all you've got
  - May as well just be a 2D bounding box
- 3D measurement uncertainty
  - Impact of a "small change" in the image domain to 3D localization uncertainty?
  - Linear error propagation

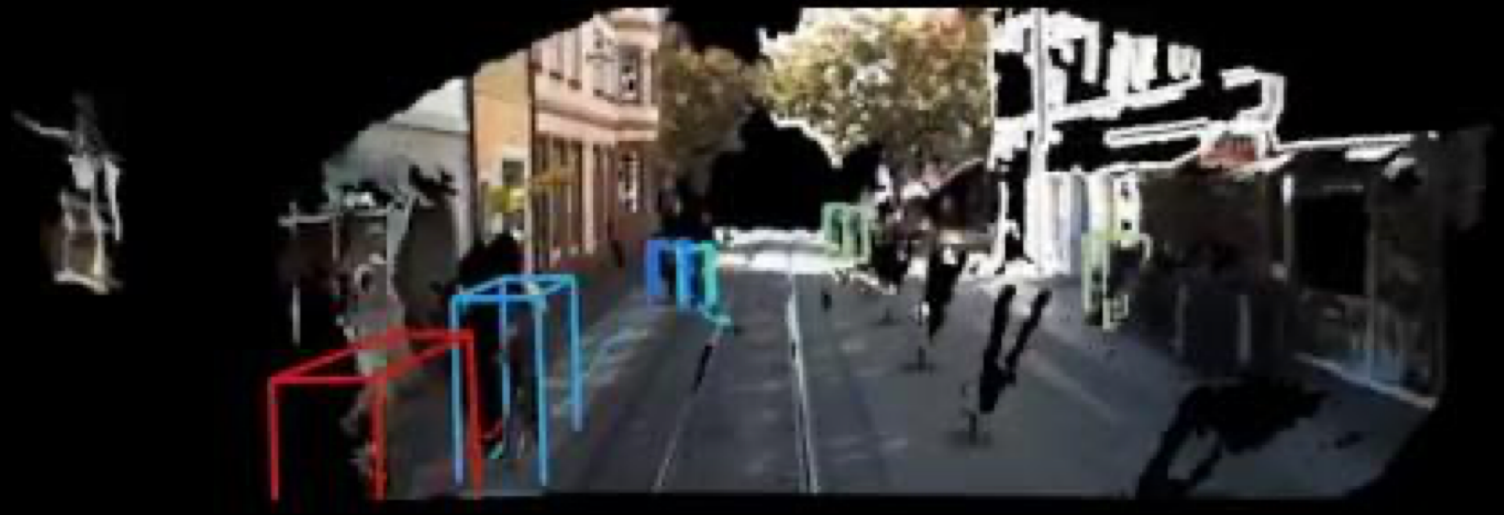


$$\Sigma_y = JSJ^T$$

Jacobians of the projection left and right camera matrices

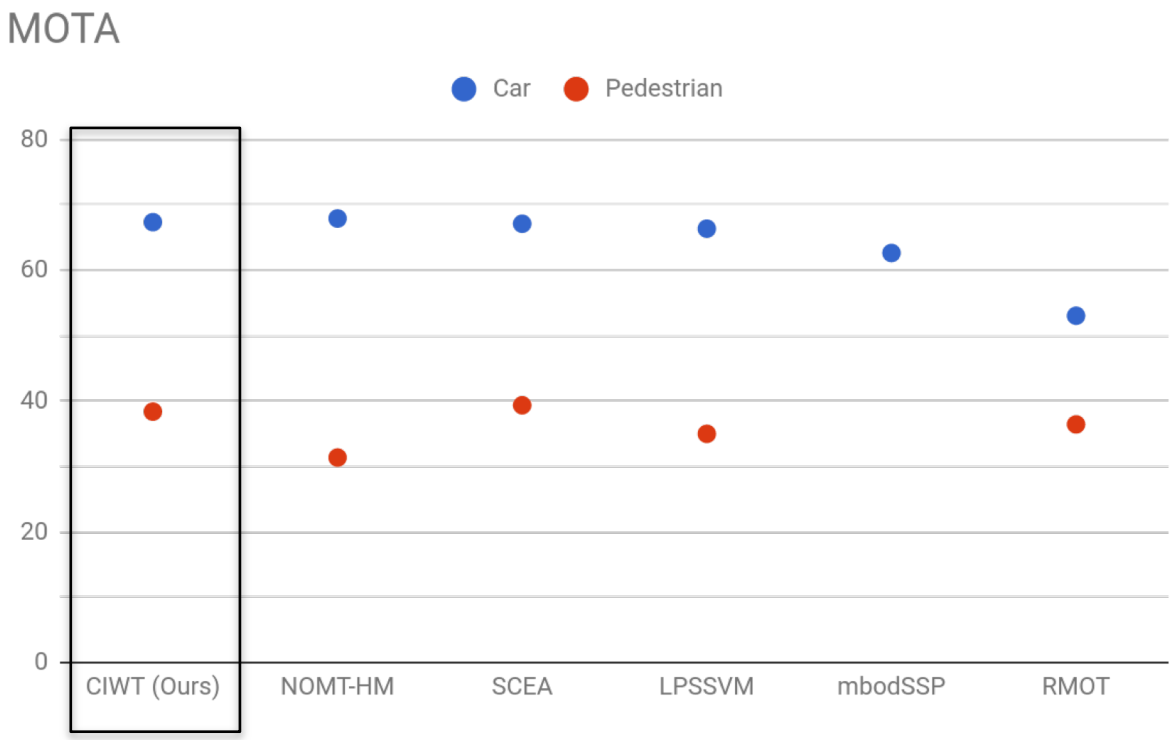
Diagonal measurement noise matrix (measurement noise of 0.5 pixel)





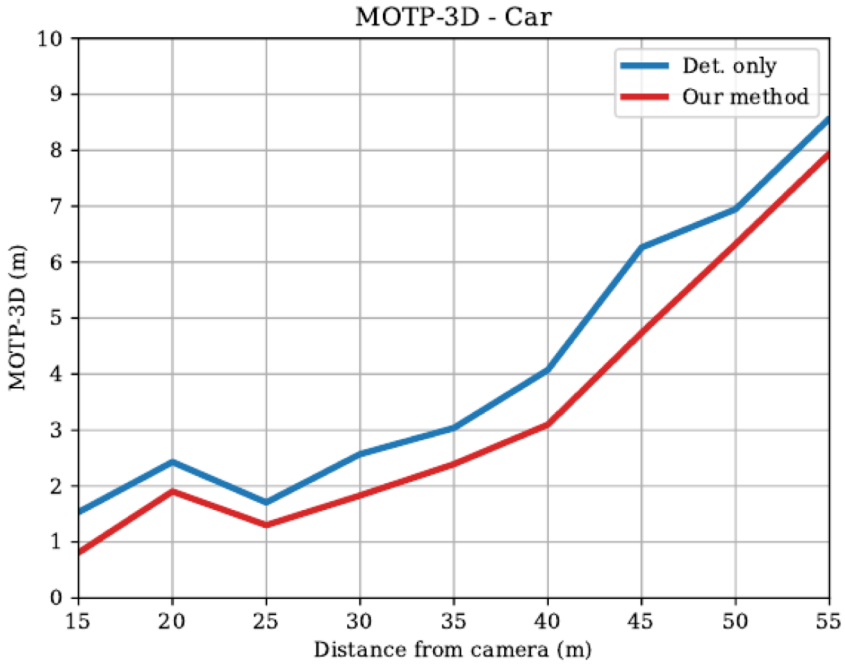

56
52

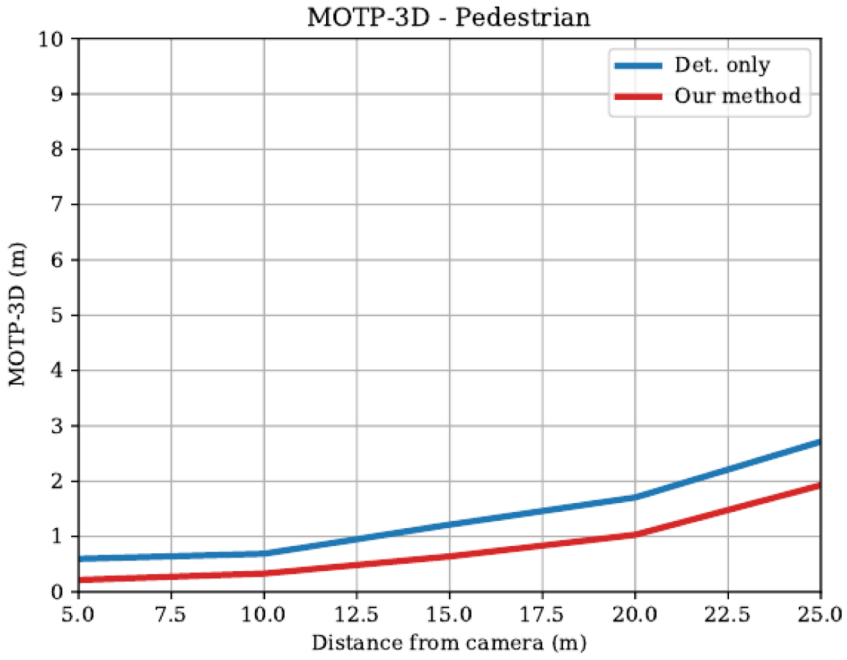
Results (2D)



KITTI Tracking Benchmark, February 2017

Results (3D)





Lower is better!

Deep Learning on Unordered Sets

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

Charles R. Qi* Hao Su* Kaichun Mo Leonidas J. Guibas
Stanford University

Abstract

Point cloud is an important type of geometric data structure. Due to its irregular format, most researchers transform such data to regular 3D voxel grids or collections of images. This, however, renders data unnecessarily voluminous and causes issues. In this paper, we design a novel type of neural network that directly consumes point clouds, which well respects the permutation invariance of points in the input. Our network, named PointNet, provides a unified architecture for applications ranging from object classification, part segmentation, to scene semantic parsing. Though simple, PointNet is highly efficient and

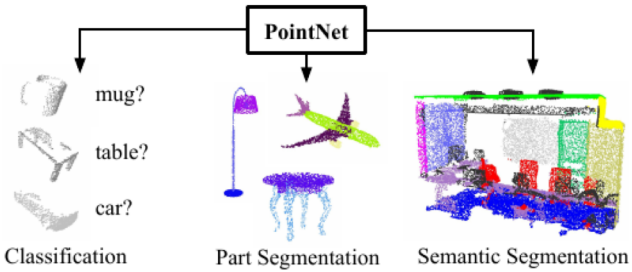
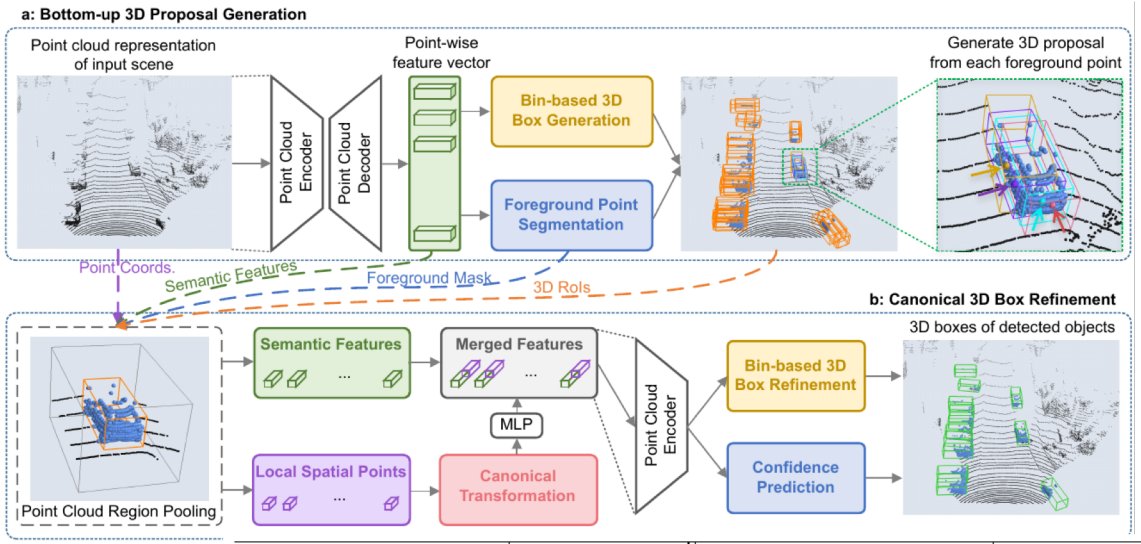
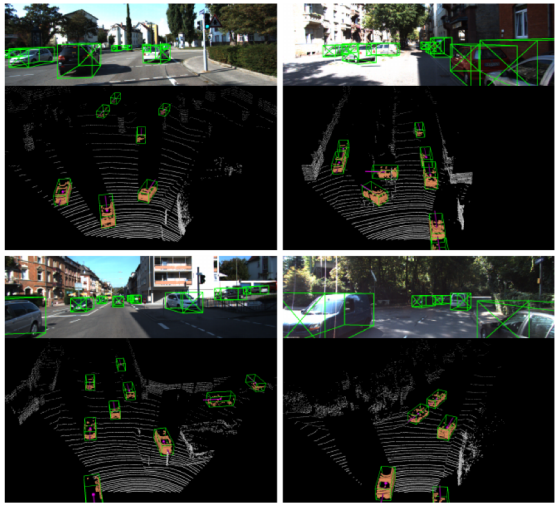


Figure 1. **Applications of PointNet.** We propose a novel deep net architecture that consumes raw point cloud (set of points) without voxelization or rendering. It is a unified architecture that learns both global and local point features, providing a simple, efficient and effective approach for a number of 3D recognition tasks.

- Seminal paper by Qi et al., CVPR'17
- Game-changer

PointRCNN

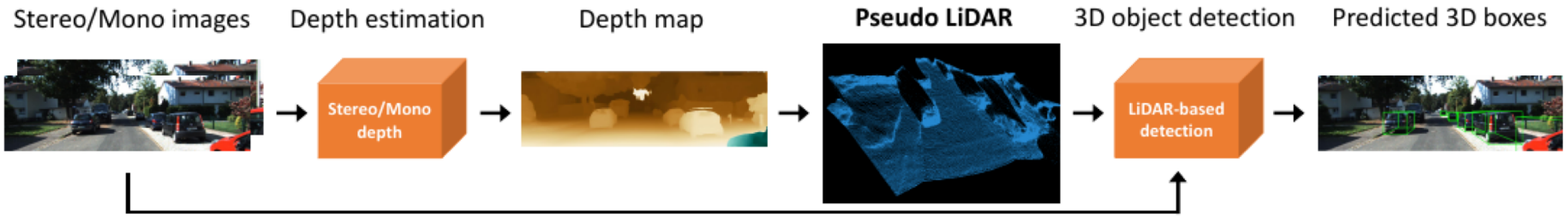




Method	Modality	Car (IoU=0.7)			Pedestrian (IoU=0.5)			Cyclist (IoU=0.5)		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
MV3D [4]	RGB + LiDAR	71.09	62.35	55.12	-	-	-	-	-	-
UberATG-ContFuse [17]	RGB + LiDAR	82.54	66.22	64.04	-	-	-	-	-	-
AVOD-FPN [14]	RGB + LiDAR	81.94	71.88	66.38	50.80	42.81	**40.88**	64.00	52.18	46.61
F-PointNet [25]	RGB + LiDAR	81.20	70.39	62.19	**51.21**	**44.89**	40.23	71.96	56.77	50.39
VoxelNet [43]	LiDAR	77.47	65.11	57.73	39.48	33.69	31.51	61.22	48.36	44.37
SECOND [40]	LiDAR	83.13	73.66	66.20	51.07	42.56	37.29	70.51	53.85	46.90
Ours	LiDAR	**85.94**	**75.76**	**68.32**	49.43	41.78	38.63	**73.93**	**59.60**	**53.59**

Shi et al., PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud, CVPR'19

Pseudo LiDAR

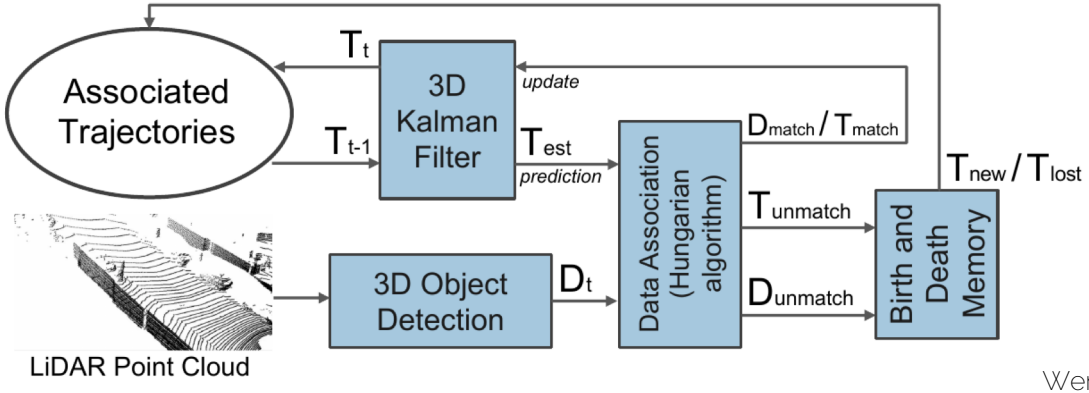


Detection algorithm	Input signal	IoU = 0.5			IoU = 0.7		
		Easy	Moderate	Hard	Easy	Moderate	Hard
MONO3D [4]	Mono	30.5 / 25.2	22.4 / 18.2	19.2 / 15.5	5.2 / 2.5	5.2 / 2.3	4.1 / 2.3
MLF-MONO [33]	Mono	55.0 / 47.9	36.7 / 29.5	31.3 / 26.4	22.0 / 10.5	13.6 / 5.7	11.6 / 5.4
AVOD	Mono	61.2 / 57.0	45.4 / 42.8	38.3 / 36.3	33.7 / 19.5	24.6 / 17.2	20.1 / 16.2
F-POINTNET	Mono	70.8 / 66.3	49.4 / 42.3	42.7 / 38.5	40.6 / 28.2	26.3 / 18.5	22.9 / 16.4
3DOP [5]	Stereo	55.0 / 46.0	41.3 / 34.6	34.6 / 30.1	12.6 / 6.6	9.5 / 5.1	7.6 / 4.1
MLF-STEREO [33]	Stereo	-	53.7 / 47.4	-	-	19.5 / 9.8	-
AVOD	Stereo	89.0 / 88.5	77.5 / 76.4	68.7 / 61.2	74.9 / 61.9	56.8 / 45.3	49.0 / 39.0
F-POINTNET	Stereo	89.8 / 89.5	77.6 / 75.5	68.2 / 66.3	72.8 / 59.4	51.8 / 39.8	44.0 / 33.5
AVOD [17]	LiDAR + Mono	90.5 / 90.5	89.4 / 89.2	88.5 / 88.2	89.4 / 82.8	86.5 / 73.5	79.3 / 67.1
F-POINTNET [25]	LiDAR + Mono	96.2 / 96.1	89.7 / 89.3	86.8 / 86.2	88.1 / 82.6	82.2 / 68.8	74.0 / 62.0

Wang et al., Pseudo-LiDAR from Visual Depth Estimation, CVPR'19

AB3D-MOT

- Strong 3D object detector, simple tracker
  - Association: bi-partite matching, **only 3D IoU**
  - Dynamic model: const-velocity Kalman Filter
  - Why does it "simple" approach work so well?
  - Why is 3D IoU now a good idea?



$$\mathbf{x}^k = [x, y, z, \theta, w, h, l, s, v_x, v_y, v_z]$$

$$f(\mathbf{x}):$$

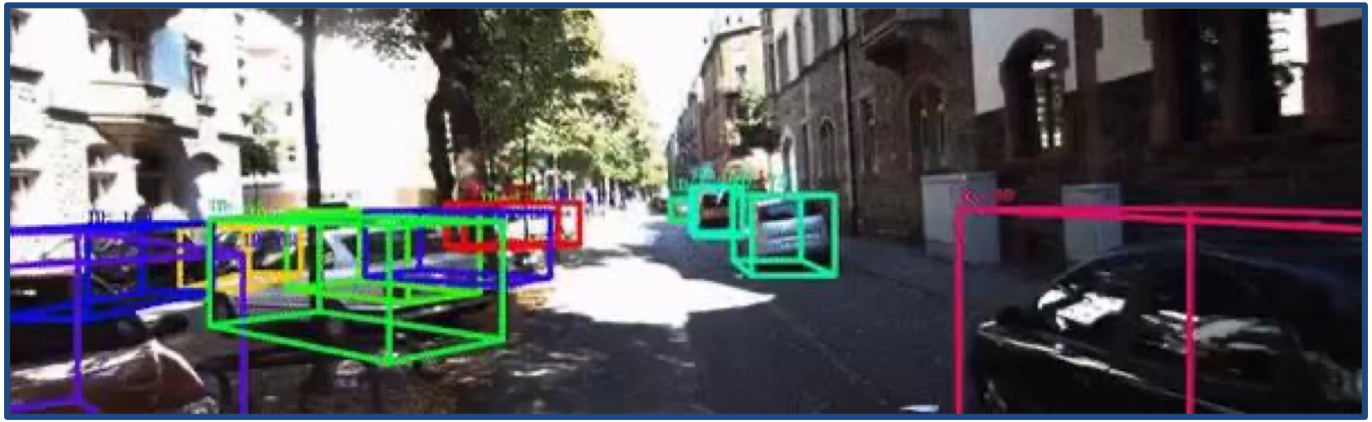
$$\hat{x}^{k+1} = x + v_x,$$

$$\hat{y}^{k+1} = y + v_y,$$

$$\hat{z}^{k+1} = z + v_z.$$

Weng et al., A Baseline for 3D Multi-Object Tracking, arXiv'19

AB3D-MOT

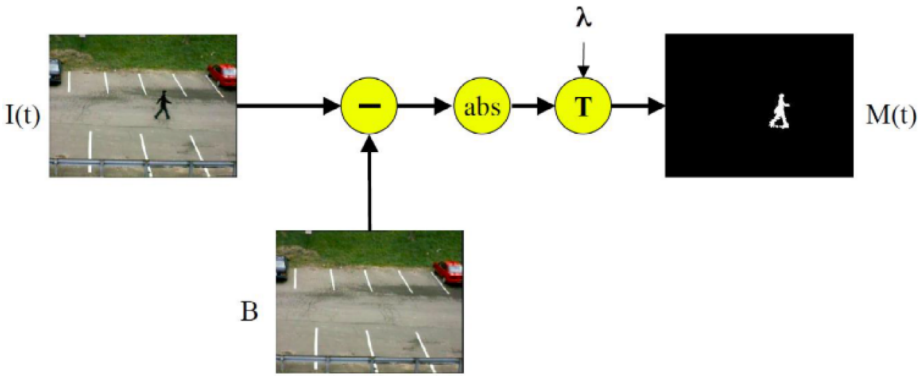


Method	Type	**MOTA** (%) ↑	MOTP (%) ↑	MT (%) ↑	ML (%) ↓	IDS ↓	FRAG ↓	FPS ↑
Complexer-YOLO [26]	3D	75.70	78.46	58.00	**5.08**	1186	2092	**100.0**
DSM [22]	3D	76.15	83.42	60.00	8.31	296	868	10.0 (GPU)
MDP [46]	2D	76.59	82.10	52.15	13.38	130	387	1.1
LP-SSVM [27]	2D	77.63	77.80	56.31	8.46	62	539	50.9
FANTrack [21]	3D	77.72	82.32	62.61	8.76	150	812	25.0 (GPU)
NOMT [38]	2D	78.15	79.46	57.23	13.23	**31**	**207**	10.3
MCMOT-CPD [28]	2D	78.90	82.13	52.31	11.69	228	536	**100.0**
extraCK [24]	2D	79.99	82.46	62.15	5.54	343	938	33.9
3D-CNN/PMBM [23]	2.5D	80.39	81.26	62.77	6.15	121	613	71.4
JCSTD [25]	2D	80.57	81.81	56.77	7.38	61	643	14.3
BeyondPixels [20]	2D	**84.24**	**85.73**	**73.23**	**2.77**	468	944	3.3
Ours	3D	**83.84**	**85.24**	**66.92**	11.38	**9**	**224**	**214.7**

3D Open-set MOT

Making Tracking Category-Agnostic Again

- Vision-based tracking back in the days …

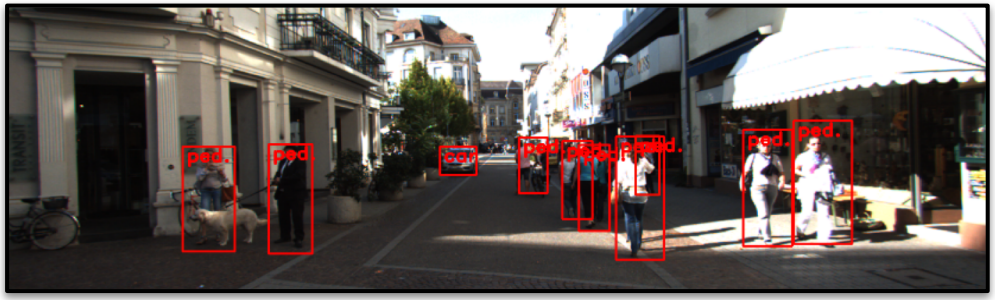


- Robotics
  - LiDAR bottom-up segmentation

Towards Open-set MOT

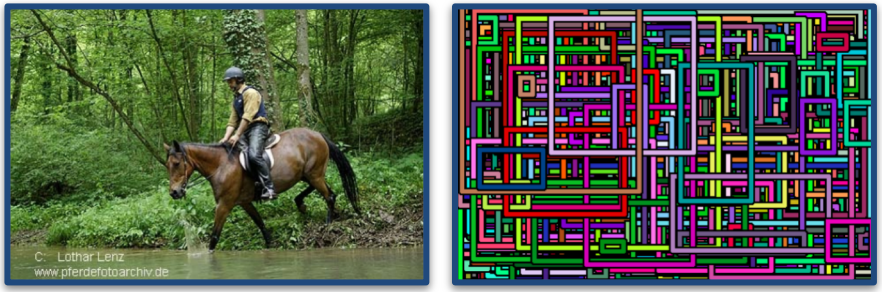
- SONAR, RADAR

- Vision-based tracking
  - How to obtain object cues? => Object detections!

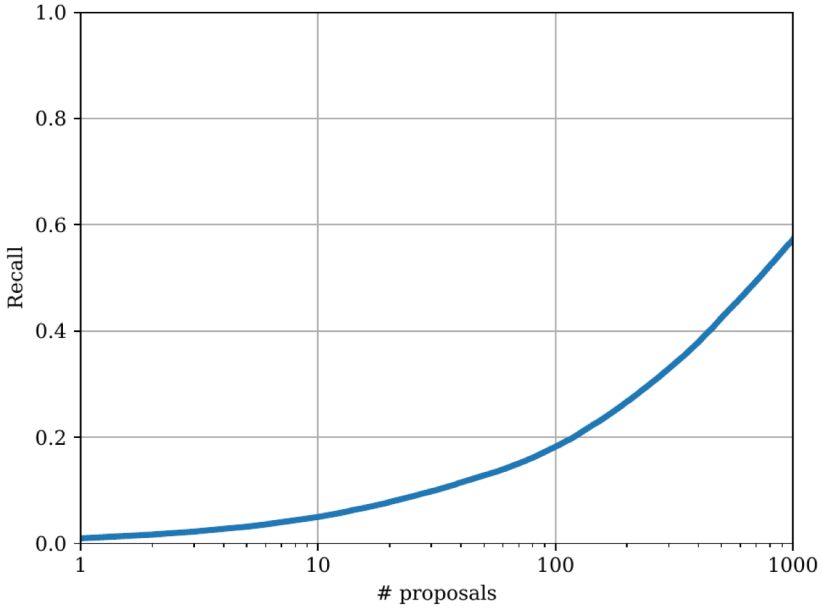


Towards Open-set MOT

- Goal: track *any* object
- How to obtain object cues?
  - Bottom-up object proposals



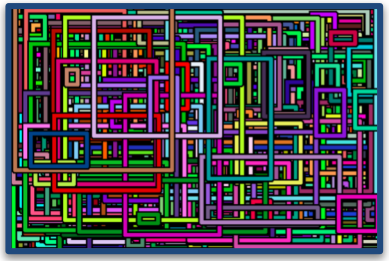
Cho et al., CVPR'15

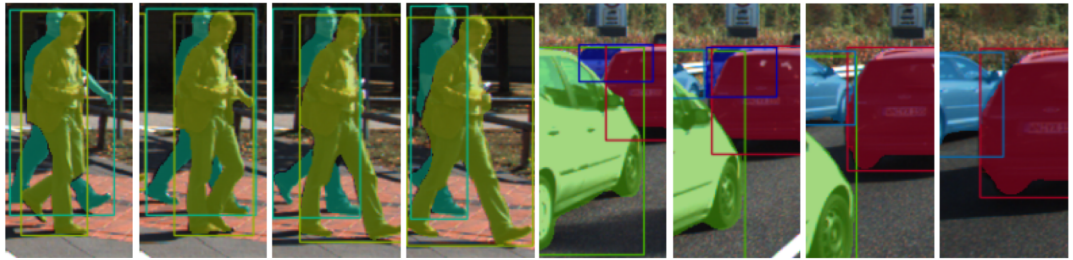


EdgeBoxes (Zitnick&Dollar et al., ECCV'14), KITTI

Challenges

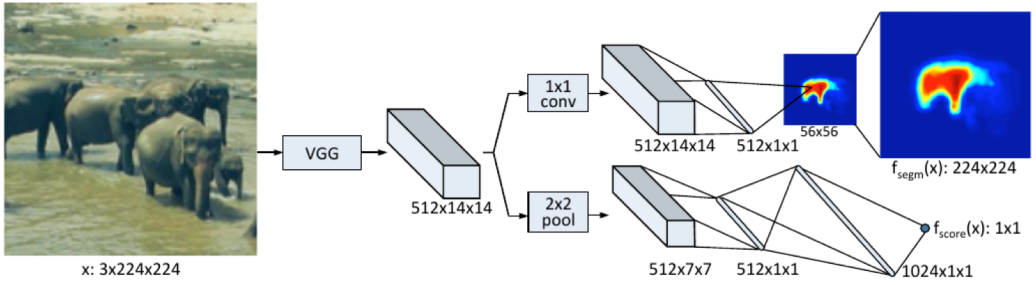
- Data association (several hundred observations/frame)
- Bounding boxes actually not that awesome representation for arbitrary objects





2016: Learning to Propose Objects using CNNs

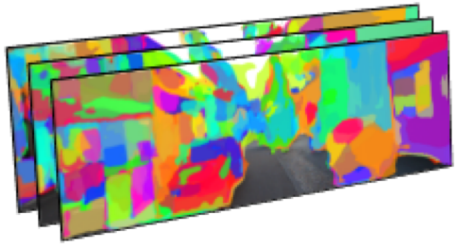
- Pinheiro et. al., NIPS'16 (Deepmask), ECCV'16, (Sharpmask), data-driven
- Big step forward -- but still many (>100) proposals/frame needed!



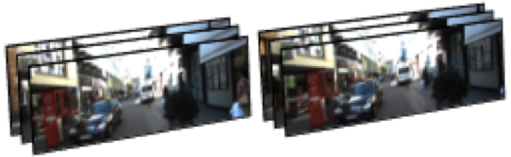
	Box Proposals			
	AR@10	AR@100	AR@1000	AUC
EdgeBoxes [34]	.074	.178	.338	.139
Geodesic [16]	.040	.180	.359	.126
Rigor [14]	-	.133	.337	.101
SelectiveSearch [31]	.052	.163	.357	.126
MCG [24]	.101	.246	.398	.180
DeepMask20	.139	.286	.431	.217
DeepMask20*	.152	.306	.432	.228
DeepMaskZoom	.150	**.326**	**.482**	**.242**
DeepMaskFull	.149	.310	.442	.231
DeepMask	**.153**	.313	.446	.233

2017: CAMOT

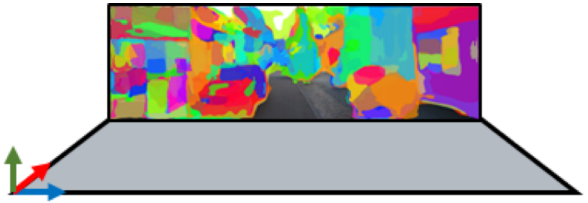
Object Proposals (learned)



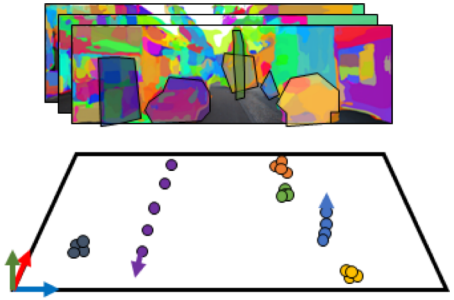
Stereo Video



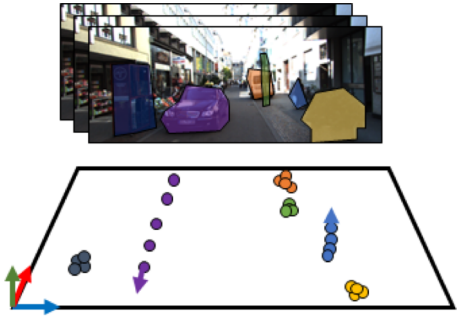
Proposal 3D localization



Proposal Track Generation (jointly 3D - image space)



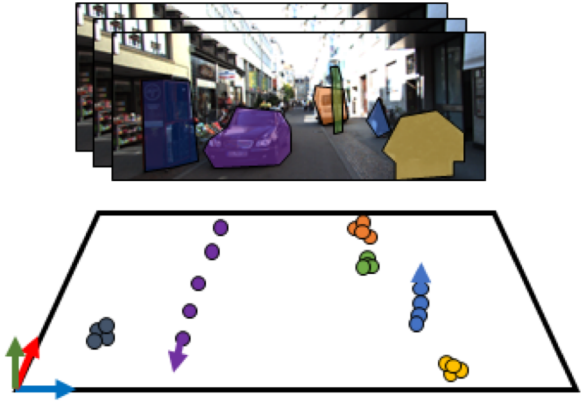
Track Classification + CRF



Osep et al., Track, Then Decide: Category-Agnostic Vision-Based Multi-Object Tracking, ICRA'18

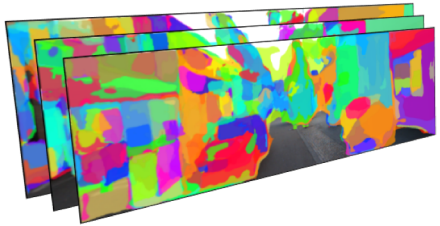
CAMOT

- Parametrize target state using 3D position and segmentation mask (const-velocity Kalman Filter will do)
- Mask - RLE (size ~ O (sqrt(#pixels))
  - Operations, such as mask IoU do not require decoding!
  - Mask IoU ~ O (sqrt(M)*sqrt(N))
    - M, N … mask area



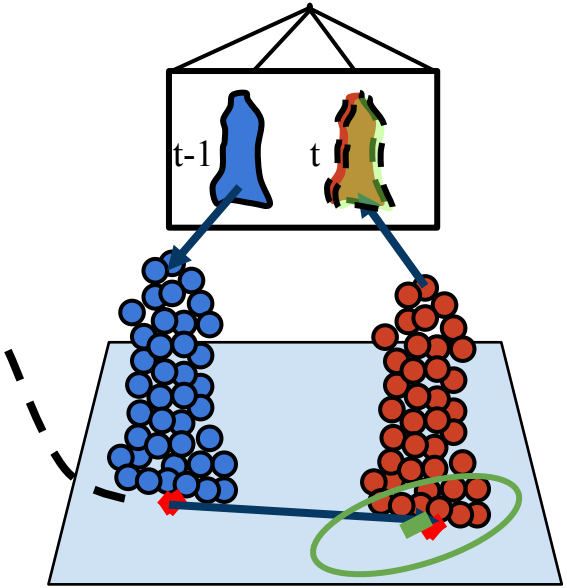
Putting Everything Together

Object Proposals (learned)

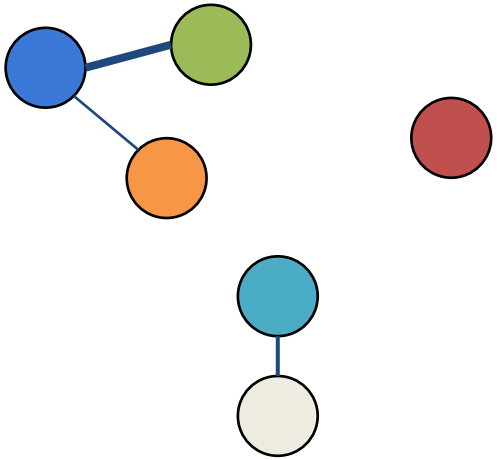


Stereo Video

Proposal Track Generation and Scoring (jointly 3D - image space)



Classification + Inference



Step 2: Selected Tracks

4D Generic Video-Object Proposals

- Based on CAMOT, offline object proposal generation
- Use Mask R-CNN, trained in category-agnostic setting
  - Additional “fine” classification head provides semantic information about objects seen during the training
  - 80 object classes seen during the training

Osep et al., 4D Generic Video Object Proposals, ICRA'20

Learning to Segment Every Thing

Learning to Segment Every Thing

Ronghang Hu[1,2,*] Piotr Dollár[2] Kaiming He[2] Trevor Darrell[1] Ross Girshick[2]
[1]BAIR, UC Berkeley [2]Facebook AI Research (FAIR)

Abstract

Most methods for object instance segmentation require all training examples to be labeled with segmentation masks. This requirement makes it expensive to annotate new categories and has restricted instance segmentation models to ~100 well-annotated classes. The goal of this paper is to propose a new partially supervised training paradigm, together with a novel weight transfer function, that enables training instance segmentation models on a large set of categories all of which have box annotations, but only a small fraction of which have mask annotations. These contributions allow us to train Mask R-CNN to detect and segment 3000 visual concepts using box annotations from the Visual Genome dataset and mask annotations from the 80 classes in the COCO dataset. We evaluate our approach in a controlled study on the COCO dataset. This work is a first

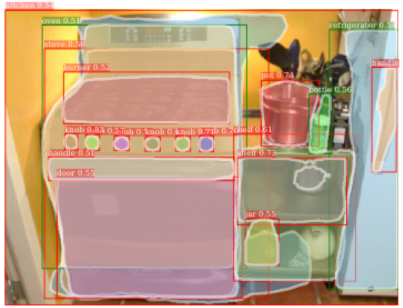
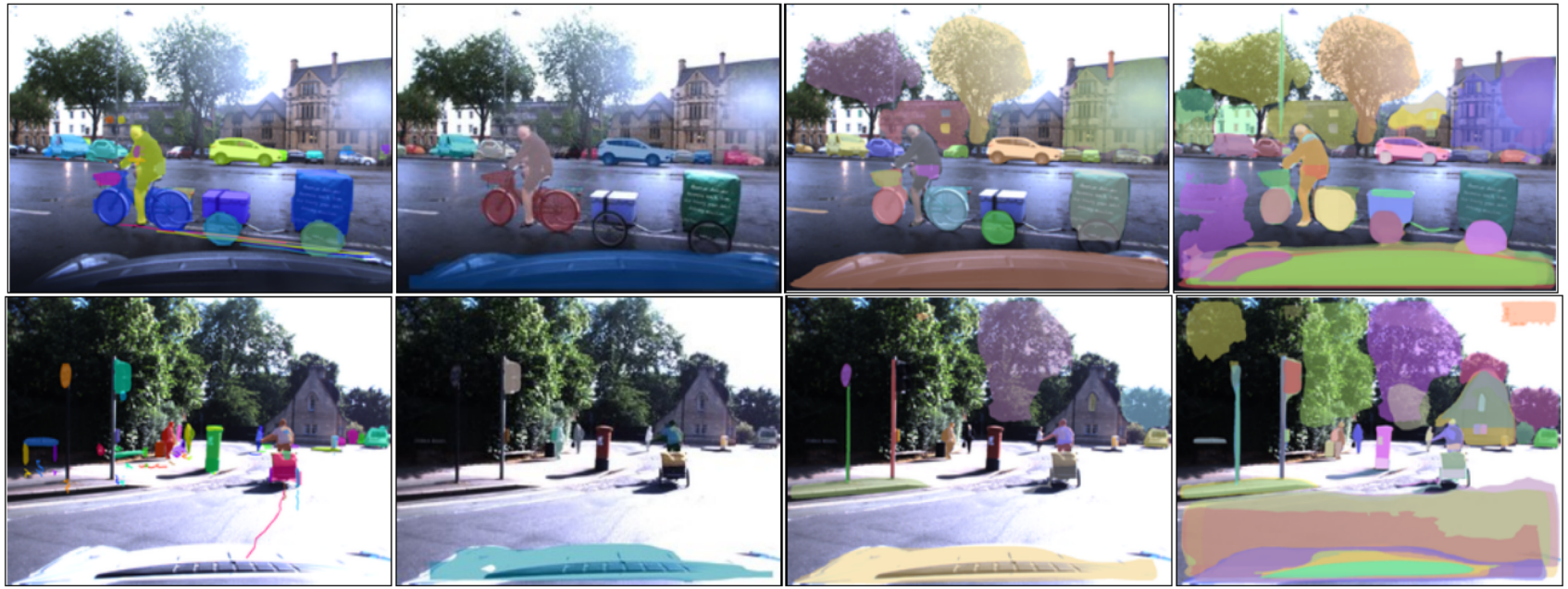


Figure 1. **We explore training instance segmentation models with partial supervision**: a subset of classes (green boxes) have instance mask annotations during training; the remaining classes (red boxes) have only bounding box annotations. This image

- Jointly train a variant of Mask R-CNN on COCO (80 classes) and a large-scale Visual Genome dataset (3K+ object classes with bounding box level supervision!)

Results

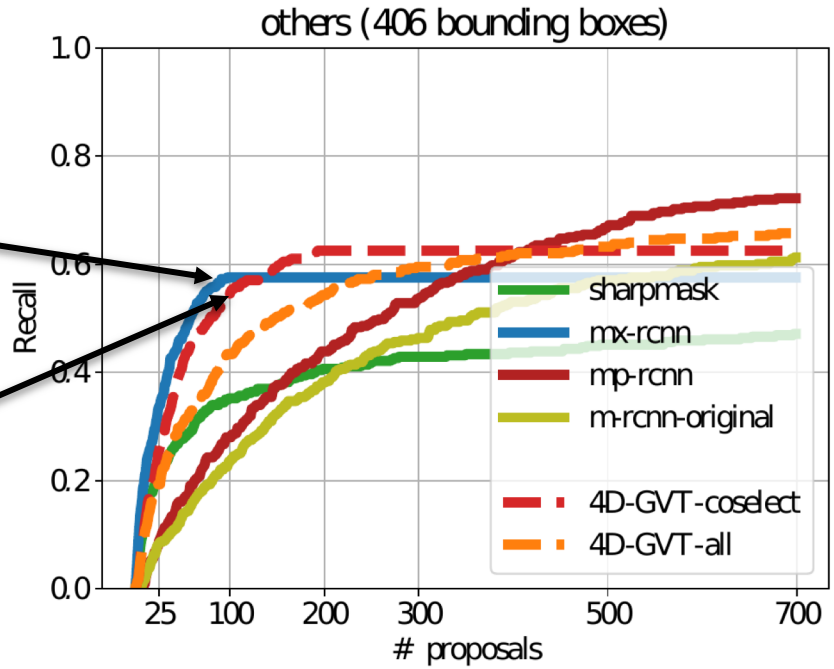


4D-GVT | *Mask R-CNN* | *Mask^X R-CNN (vanilla)* | *Mask^X R-CNN (all)*

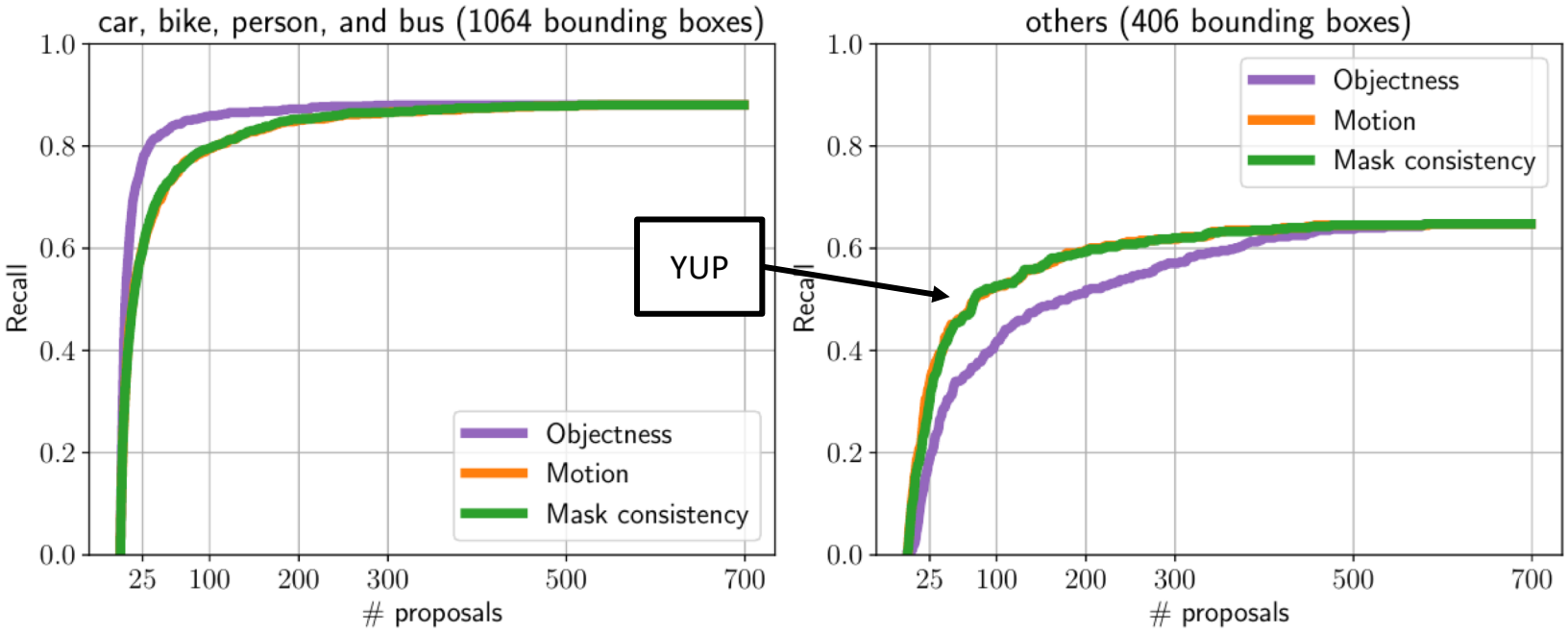
Can we Generalize Well?

Mask X R-CNN:
Visual Genome + COCO
(3,000+ classes)

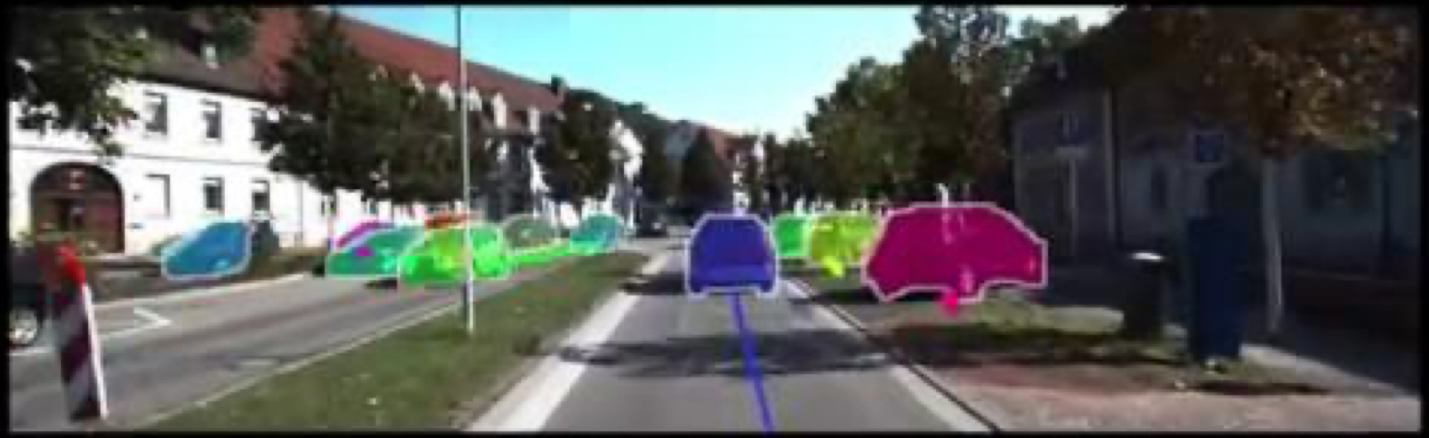
4D-GVT (and all others):
COCO (80 classes)



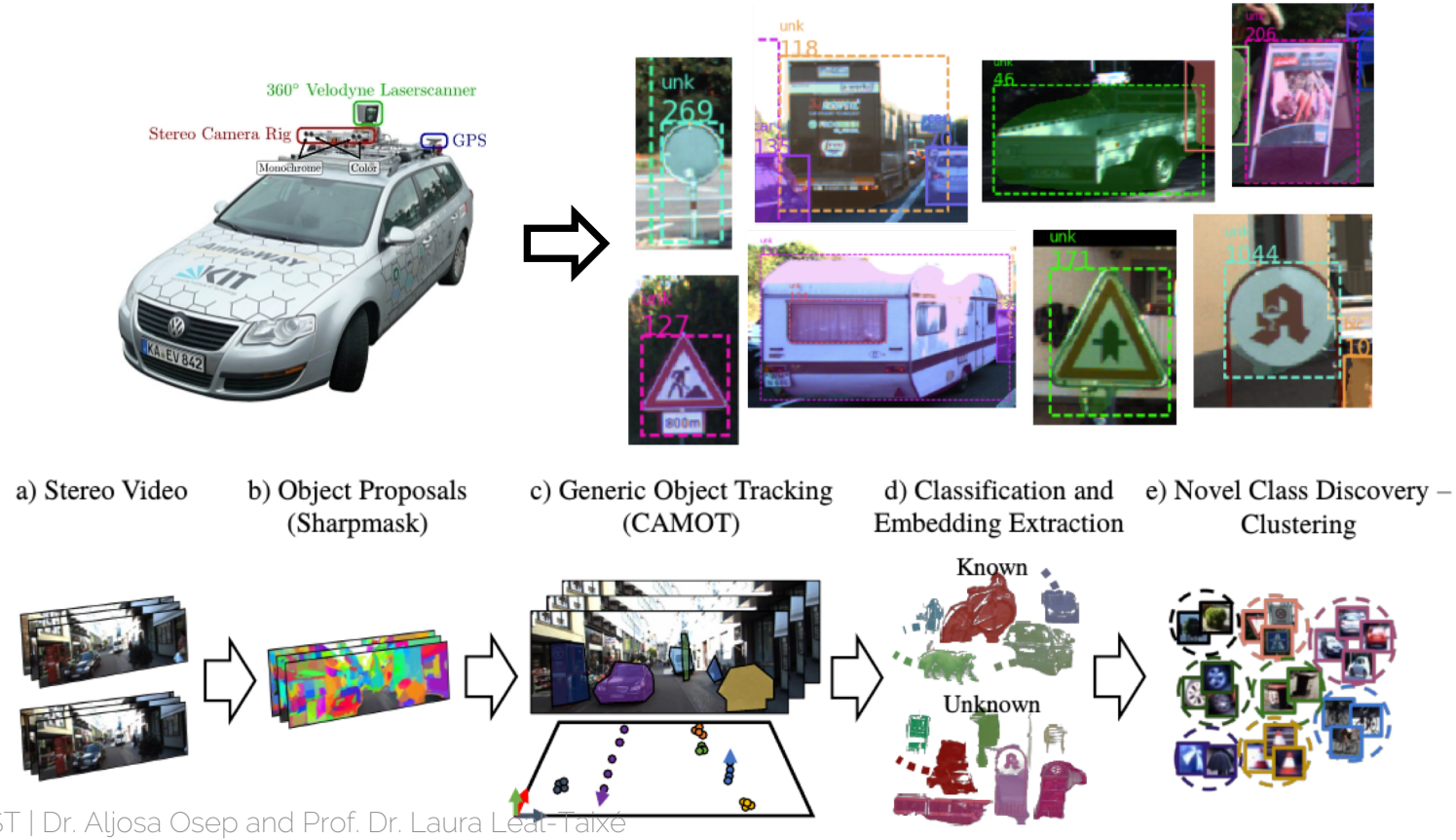
Are Mask and Motion Consistency Useful Objectness Cues?



Video Tubes (Cars and Pedestrians Only)



Why is All This so Cool?



Why is All This so Cool?

Slide Credits

- A few slides recycled from Bastian Leibe, RWTH Aachen University
- Slide 22, figure from Roger R Labbe Jr: Kalman and Bayesian Filters in Python
  - In case you want to learn more about Bayesian filtering, check out his cool Jupyter notebook: https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python

Thank you for your attention!