Multiple-Kernel Based Vehicle Tracking Using 3D Deformable Model and Camera Self-Calibration

Team 4
Information Processing Lab, University of Washington
Deep Learning Technology Center, Microsoft Research

Introduction

- Traffic surveillance
  - Accident prevention
  - Abnormal behavior detection
  - Traffic condition analysis
- Multiple Object Tracking (MOT)
  - Object detection/classification + data association
  - It provides information about the locations of multiple objects in time.

Introduction

- Occlusion problem

Introduction

- Constrained multiple-kernel (CMK) tracking [Chu et al. '13]
  - Main idea: 2+ kernels to describe an object
  - A kernel is defined by (spatially) weighted color histogram.
  - Multiple kernels are bound together under certain constraints **C(x)**.
- Problem formulation

$$\min_{\mathbf{x}} J(\mathbf{x}) = \sum_{\kappa=1}^{N_\kappa} w_\kappa J_\kappa(\mathbf{x})$$

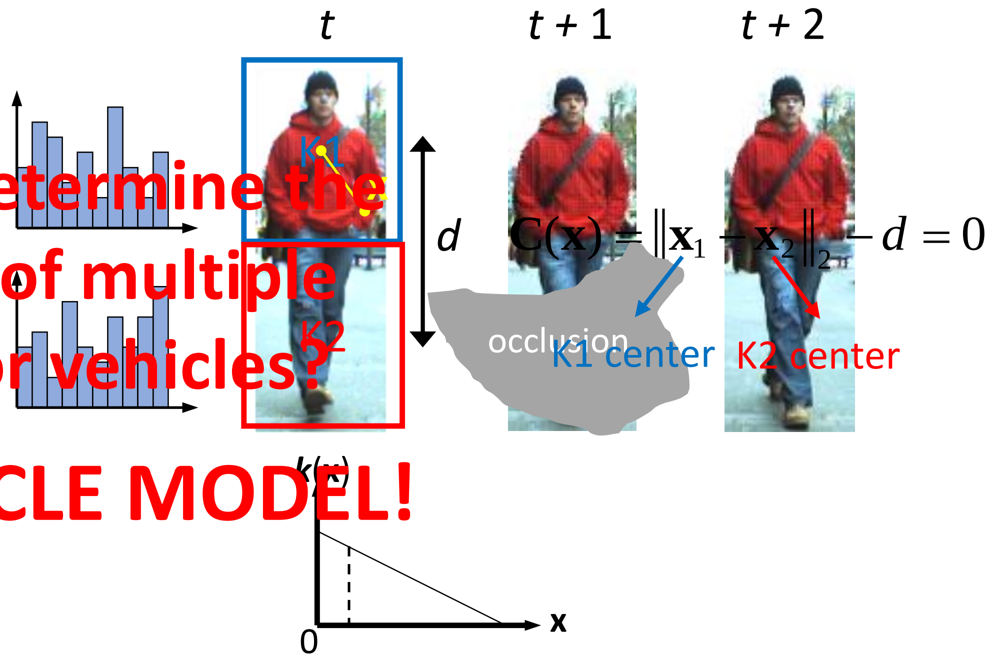
subject to $\mathbf{C}(\mathbf{x}) = \mathbf{0}$

for κ^{th} kernel,

$$J_\kappa(\mathbf{x}) \propto 1/\, simi_\kappa(\mathbf{x}),$$

$$w_\kappa = \gamma \times simi_\kappa(\mathbf{x})$$

$simi_\kappa(\boldsymbol{x})$: -(K-L dist.)



How to determine the locations of multiple kernels for vehicles?

-3D VEHICLE MODEL!

Introduction

- Other Challenges in object tracking [Leal-Taixe et al. ’15], [Milan et al. ’16]
  - Grouping of objects
  - Fast motion
  - Difference in viewpoint
  - Weather condition
  - Missing detection (tracking by detection)
  - False positives in detection (tracking by detection)
  - Initial occlusion (tracking by segmentation)
  - Object merging (tracking by segmentation)

Track 1 Approach:
SSD [Liu et al. ’16] + YOLO9000 [Redmon and Farhadi ’17]

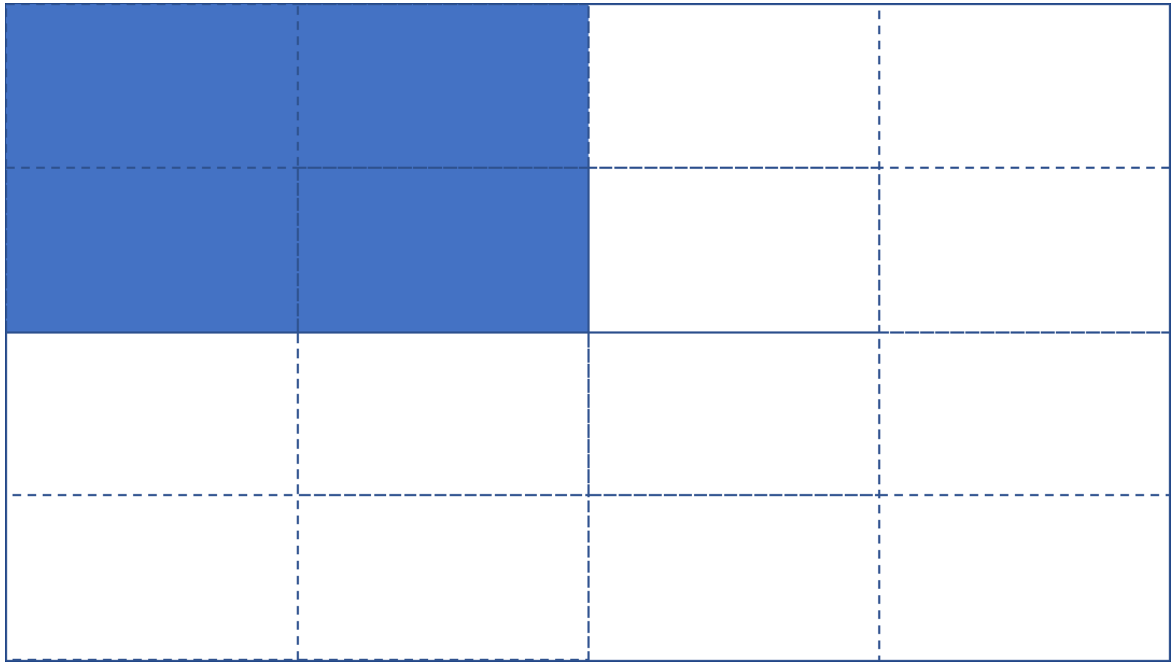
- SSD (**trained on aic480 and aic540**)
  - Multi-scale feature maps for detection
  - Different scales and aspect ratios for default boxes
  - More accurate
- YOLO (**pre-trained model on ImageNet and COCO datasets**)
  - Fast
  - Detect categories with very few objects, like Bus, Bicycle, Motorcycle and Pedestrian

SSD Training

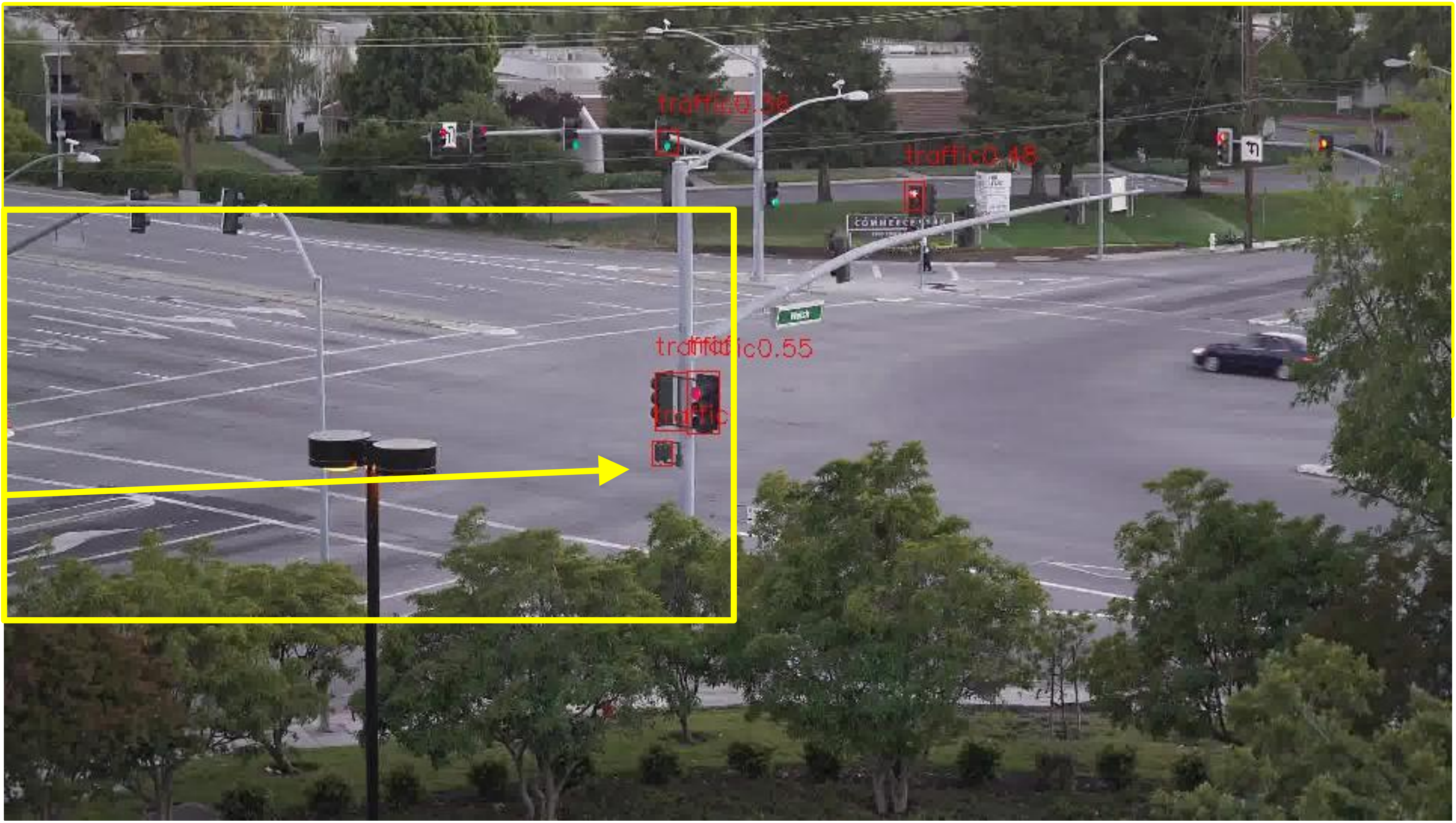
- **Training data**: aic480 and aic540
- Based on pre-trained model on ImageNet.
- **Model**: SSD_512 by vgg16
- **Parameters**: 200,000 iterations with batch_size = 16

YOLO with Multi-Scale Testing

- Divide each frame into 9 sub-regions
  - Advantage: Good for detecting small objects
  - Non-maximum suppression is used to combine results in overlapping areas.



YOLO with Multi-Scale Testing



Detect Categories with Few Objects

Pedestrian

Detect Categories with Few Objects

Bus

Detect Categories with Few Objects

Motorcycle

SSD + YOLO

- Ensemble Learning
  - Merge detected bboxes *B* from SSD (*y* = 1) and YOLO (*y* = -1) according to their confidence scores *s* and IOU ratios *r*.

Merge bounding boxes: $\hat{B} = w_1 B_1 + w_2 B_2,$ if the predictions are of the same class

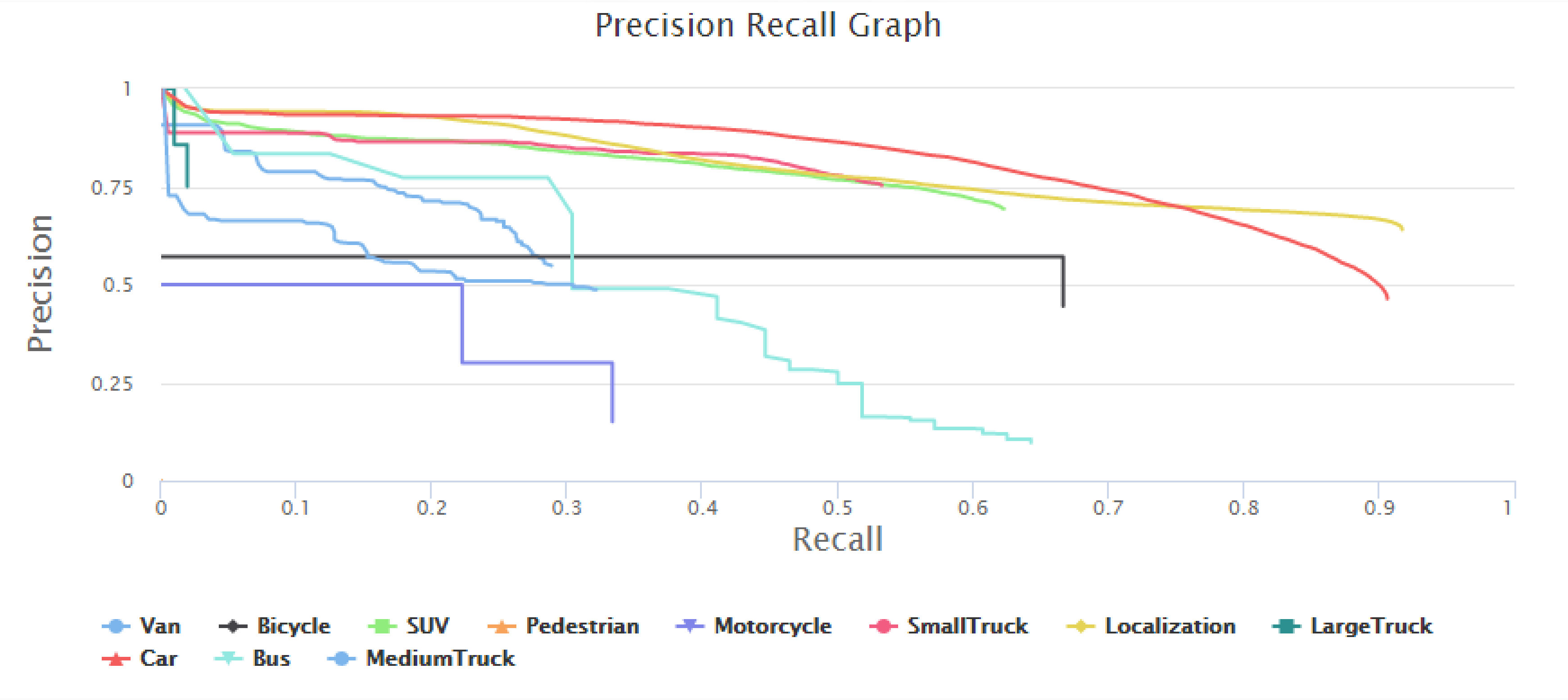
Choose prediction: $\hat{y} = w_3 s_1 + w_4 r_1 + w_5 s_2 + w_6 r_2,$ if the predictions are of different classes

$w_{1\text{-}6}$: Weights to be trained

- Advantages
  - **SSD** can detect Car, SUV, Trucks with high accuracy.
  - **YOLO** (w/ multi-scale testing) can help detect categories with very few objects, like Bus, Bicycle, Motorcycle and Pedestrian.

Track 1 Results: aic480

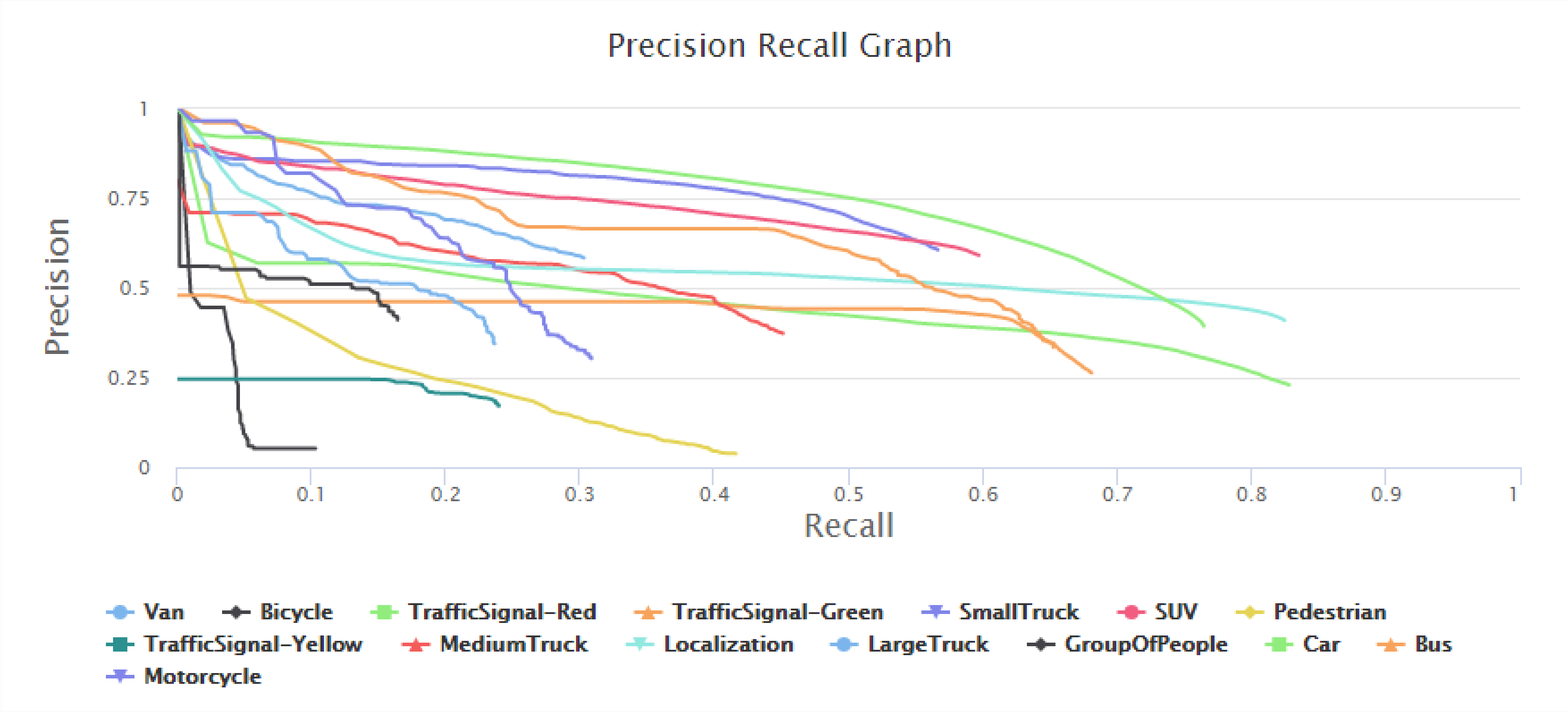
- mAP: 0.34



Class	AP	F1-score
Van	0.22	0.38
Bicycle	0.38	0.53
SUV	0.52	0.66
Pedestrian	0	0
Motorcycle	0.14	0.21
SmallTruck	0.45	0.62
Localization	0.74	0.75
LargeTruck	0.02	0.04
Car	0.75	0.61
Bus	0.35	0.17
MediumTruck	0.19	0.39

Track 1 Results: aic1080

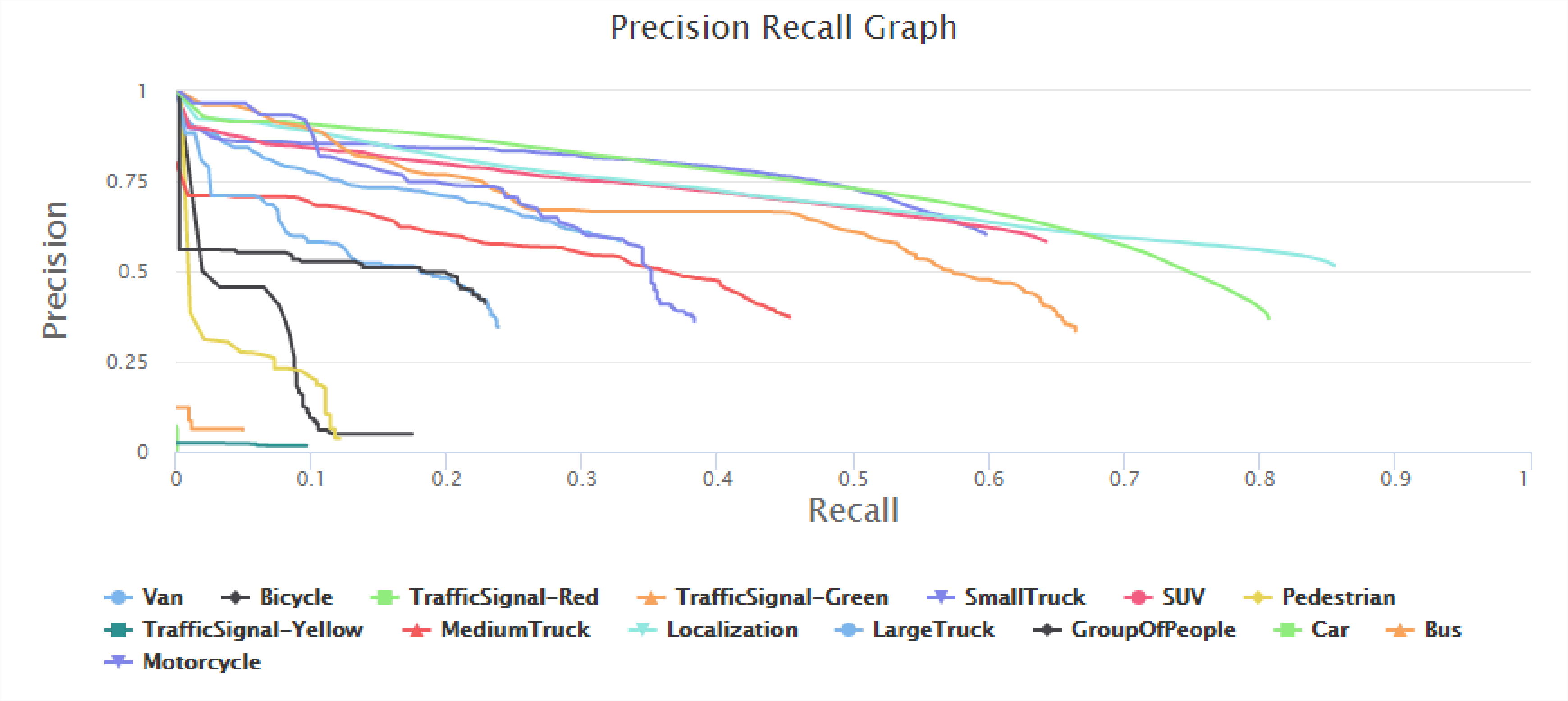
- mAP: 0.28



Class	AP	F1-score
Van	0.22	0.4
Bicycle	0.03	0.07
TrafficSignal-Red	0.37	0.36
TrafficSignal-Green	0.3	0.38
SmallTruck	0.45	0.59
SUV	0.45	0.59
Pedestrian	0.1	0.07
TrafficSignal-Yellow	0.06	0.2
MediumTruck	0.27	0.41
Localization	0.46	0.55
LargeTruck	0.14	0.28
GroupOfPeople	0.09	0.23
Car	0.59	0.52
Bus	0.45	0.44
Motorcycle	0.22	0.31

Track 1 Results: aic540

- mAP: 0.25

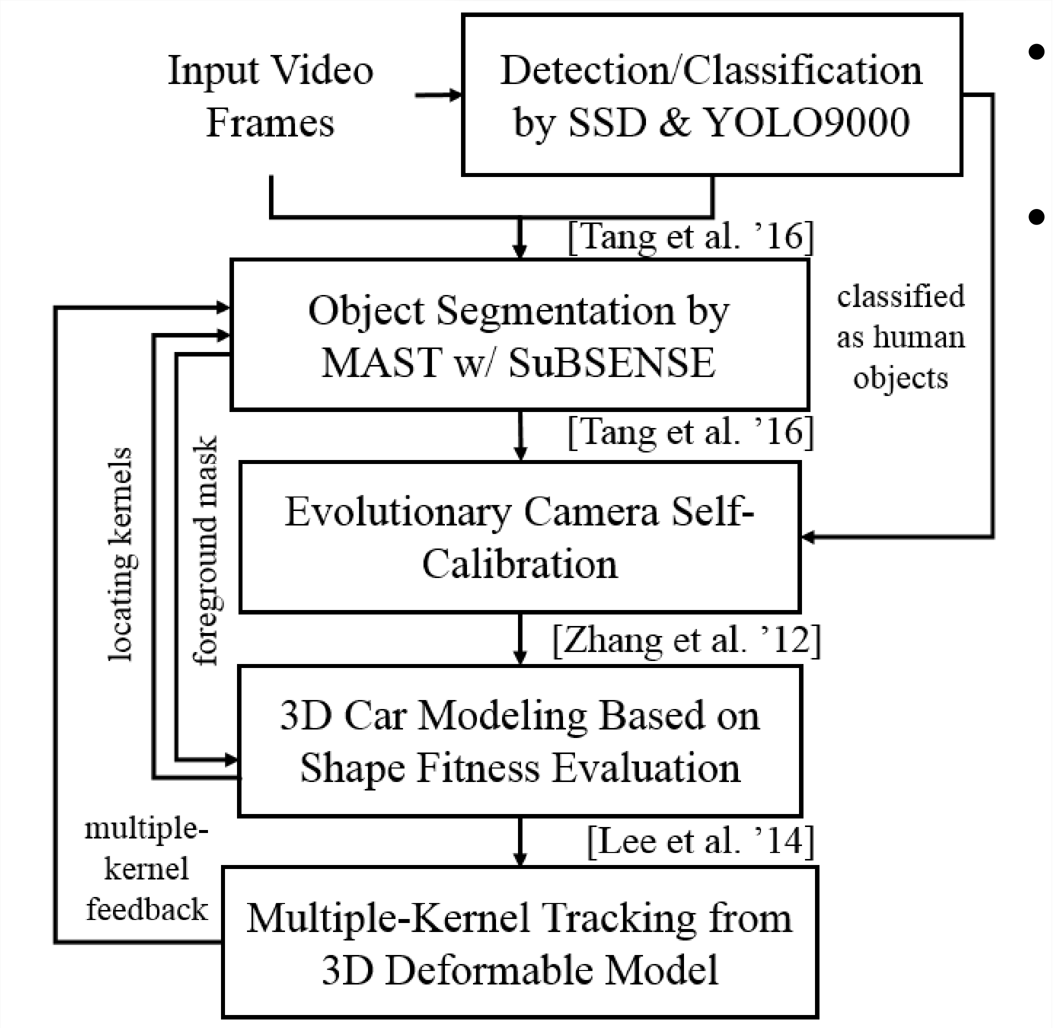


Class	AP	F1-score
Van	0.24	0.42
Bicycle	0.05	0.08
TrafficSignal-Red	0	0
TrafficSignal-Green	0	0.05
SmallTruck	0.48	0.6
SUV	0.48	0.61
Pedestrian	0.03	0.06
TrafficSignal-Yellow	0	0.03
MediumTruck	0.27	0.41
Localization	0.61	0.64
LargeTruck	0.14	0.28
GroupOfPeople	0.12	0.29
Car	0.61	0.51
Bus	0.46	0.44
Motorcycle	0.29	0.37

Track 1 Demo

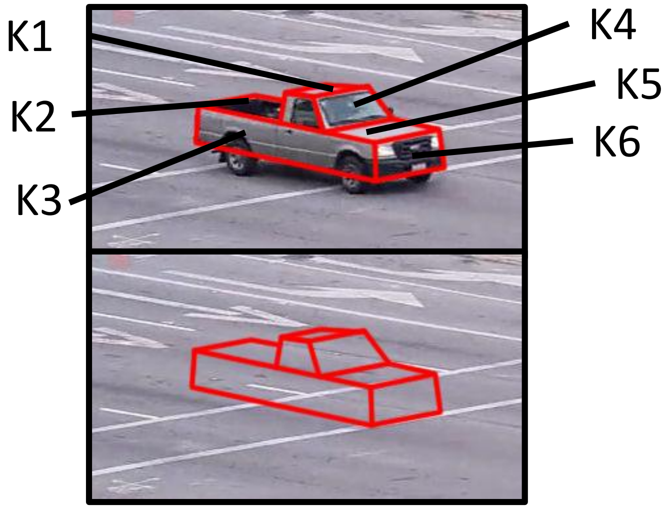

TrafficSignal-Red 0.99
TrafficSignal-Red 0.977
SUV 0.833
SUV 0.807
Car 0.975
Car 0.839
Car 0.988

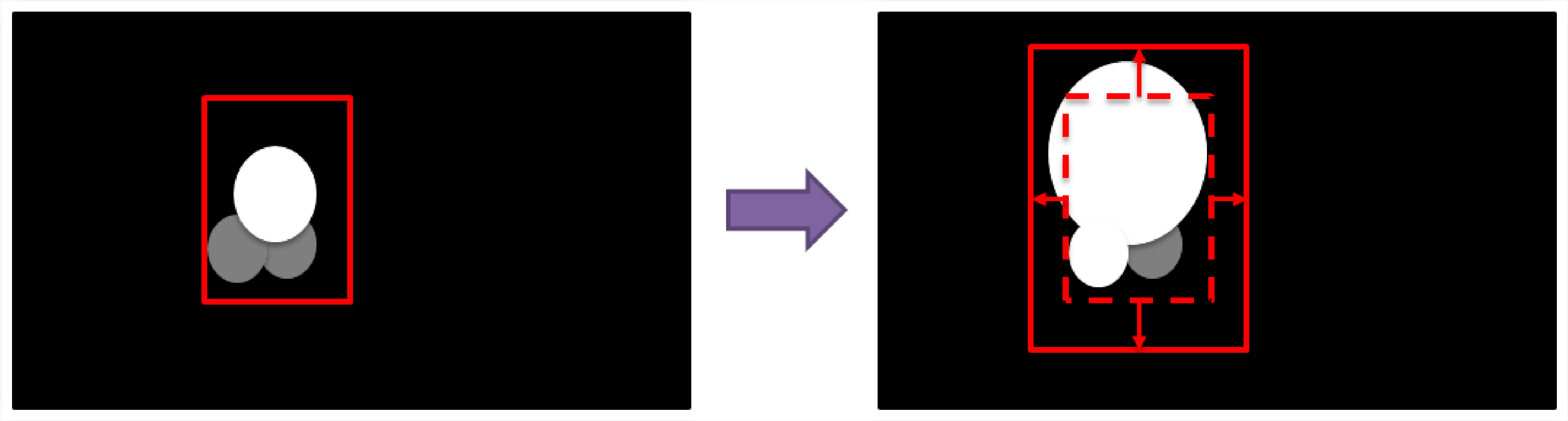
Track 2 Approach: CMK Tracking + 3D Car Modeling + Self-Calibration + Segmentation



- **Goal**: Tracking & understanding vehicle attributes at the same time!
- **Novelty / Contribution**
  - **Fully unsupervised** 2D/3D vehicle tracking, modeling and camera calibration
  - **Extension of CMK tracking** based on 3D vehicle model to handle occlusion
  - **Adaptive re-initialization** of 3D vehicle model to create better fitting
  - **Evolutionary camera self-calibration** to automatically infer 3D from 2D
  - **Adaptive object segmentation** facilitated by multiple-kernel feedback from tracking

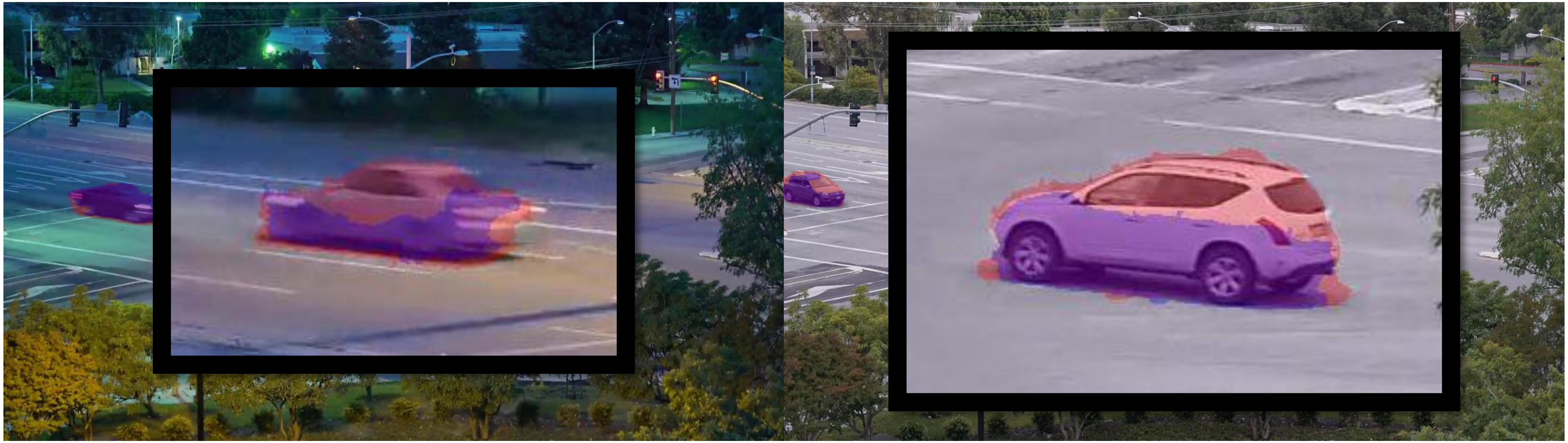
Multiple-kernel Adaptive Segmentation and Tracking (MAST)





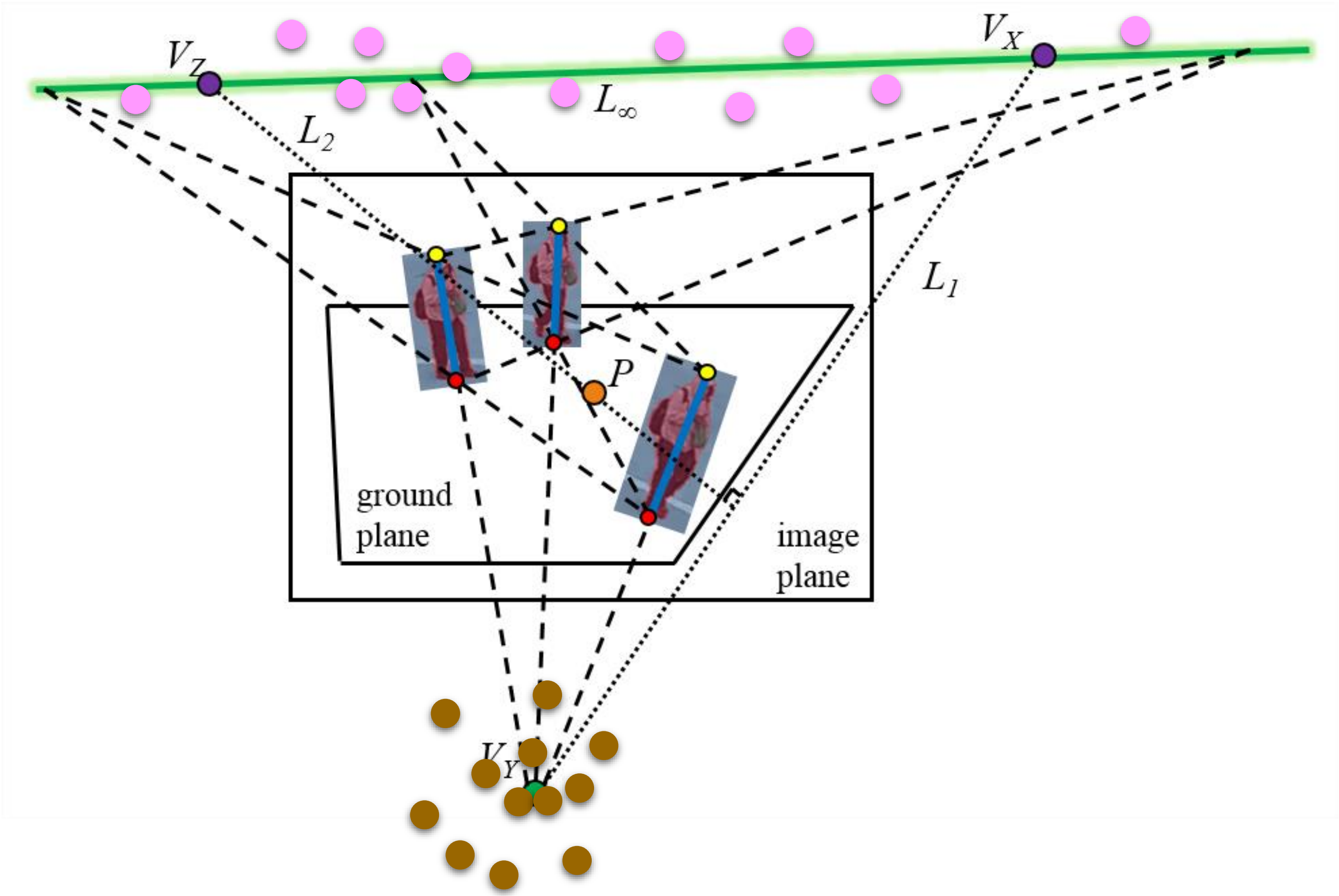
- w_{pen}: Penalty weight ∝ $simi_{\text{color}}$ / $simi_{\text{chrom}}$ base on a fuzzy Gaussian function
- Distance thresholds in background subtraction and/or the chromaticity thresholds in shadow detection is penalized by multiplying $\left(1 - w_{\text{pen}}\right)$.
- The kernel region to be re-segmented is expanded by a factor of $w_{\text{pen}}/2$.

Multiple-kernel Adaptive Segmentation and Tracking (MAST)



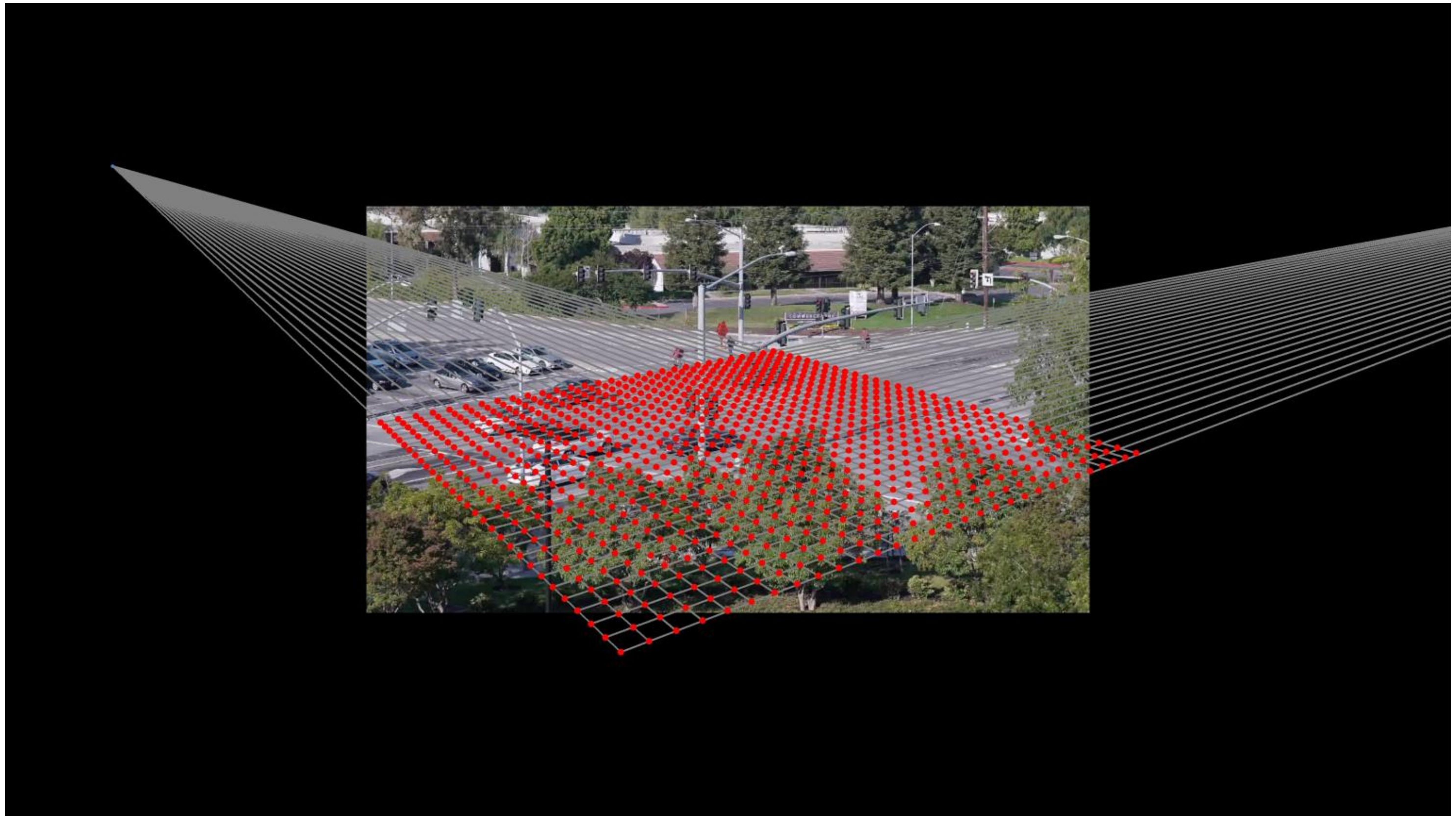
Blue: preliminary segmentation from SuBSENSE with shadow detection
Red: segmentation after applying multiple-kernel feedback from tracking

Evolutionary Camera Self-calibration



- Noise removal in V_Y estimation by **mean shift clustering**
- Noise removal in L_∞ estimation by **Laplace linear regression**
- **Evolutionary algorithm-based optimization** for vanishing points locations and camera parameters
- Convergence with only **~100 tracking positions** required

Evolutionary Camera Self-calibration



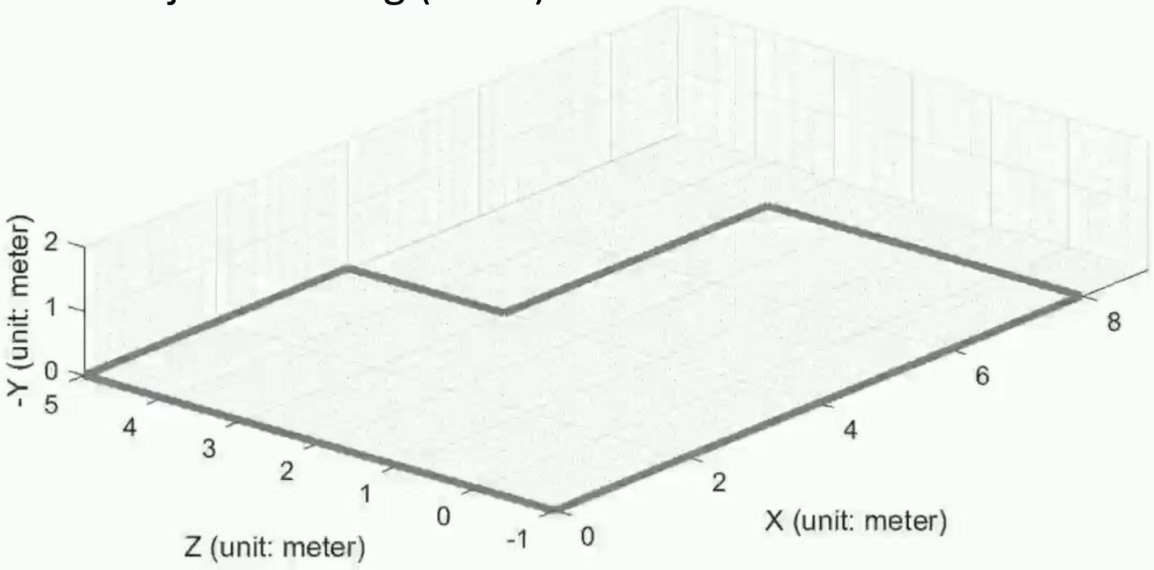
Visualization of estimated ground plane: The red dots form a (30 m * 30 m) 3D grid on the ground plane projected to 2D space

Inferring 3D from 2D

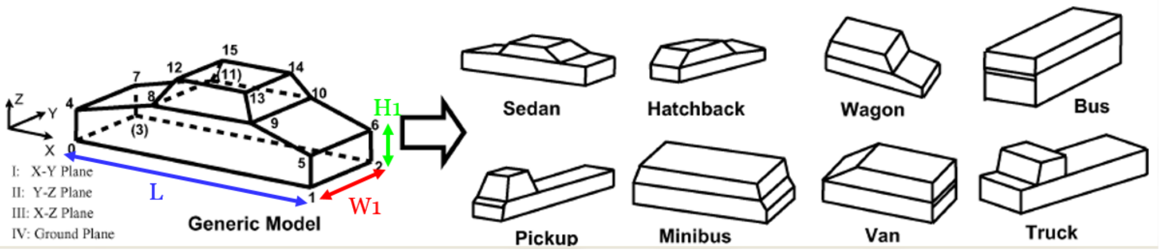
Object tracking (in 2D)

Object segmentation (w/ region of interest, i.e., ROI)

Object tracking (in 3D) via camera self-calibration



3D Vehicle Modeling



Pose Initialization → **Iterative Optimization** (EMNA$_{global}$): **Fitness Evaluation Score (FES)** ← **15 parameters** → **3-D vehicle model**

12 shape parameters

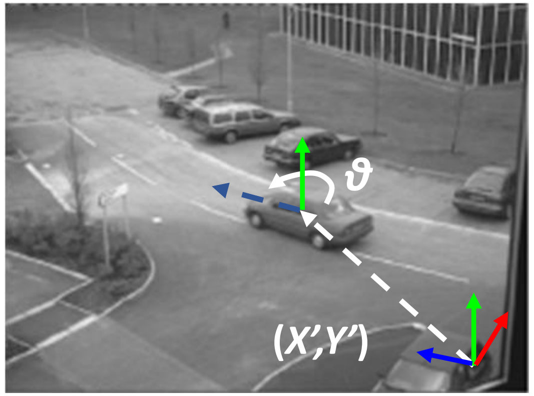
Parameters	Descriptions
$W1$	Distance from 1 to 2
$H1$	Distance from 1 to 5
$H2$	Distance from 0 to 4
L	Distance from 0 to 1
$H3$	Distance from 8 to I
$X1$	Distance from 8 to II
$X2$	Distance from 9 to II
$X3$	Distance from 12 to II
$X4$	Distance from 13 to II
$W2$	Distance from 13 to 14
$H4$	Distance from 13 to I
Δ	Distance from I to IV

15 parameters +

3 pose parameters

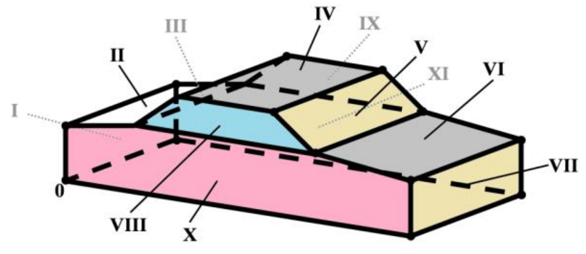
orientation: ϑ
translation: X', Y'

iteration



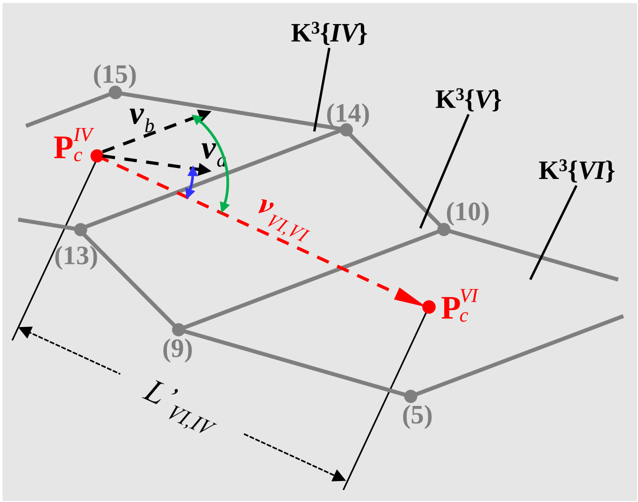
3D CMK Vehicle Tracking

- Regard each patch of the 3D vehicle model as a kernel.



K{•}	Vertices	Description
I	0, 3, 4, 7	rear-side
II	4, 7, 8, 11	boot cover
III	8, 11, 12, 15	rear window
IV	12, 13, 14, 15	roof
V	9, 10, 13, 14	windshield
VI	5, 6, 9, 10	engine hood
VII	1, 2, 5, 6	front-side
VIII	8, 9, 12, 13	right window
IX	10, 11, 14, 15	left window
X	0, 1, 4, 5, 8, 9	right-side
XI	2, 3, 6, 7, 10, 11	left-side

- Constraints in 3D space



1. $\left\| \mathbf{P}_c^{\kappa} - \mathbf{P}_c^{\kappa^*} \right\|^2 = (L'_{\kappa,\kappa^*})^2$

2. $$\begin{cases} \dfrac{v_a \cdot v_{\kappa,\kappa^*}}{\| v_a \| \| v_{\kappa,\kappa^*} \|} = \cos\left(\phi_{\kappa,\kappa^*}\right) \\ \dfrac{v_b \cdot v_{\kappa,\kappa^*}}{\| v_b \| \| v_{\kappa,\kappa^*} \|} = \cos\left(\varsigma_{\kappa,\kappa^*}\right) \end{cases},$$

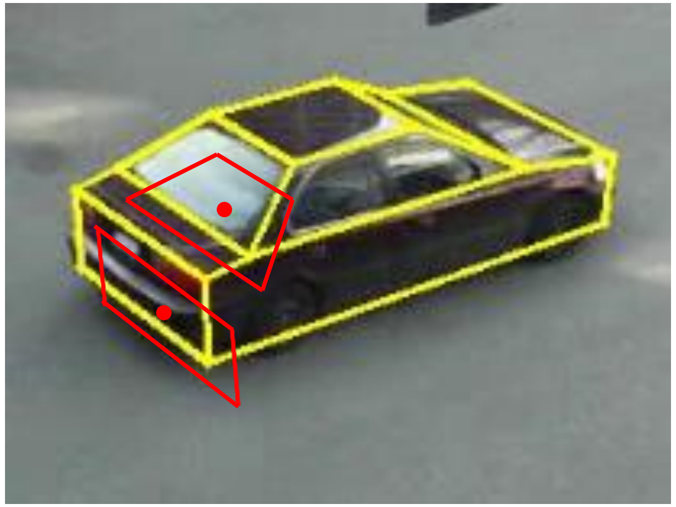
for any visible $K^3\{\kappa \mid \kappa \neq \kappa^*\}$

- New Cost function

$$J(\mathbf{x}) = \sum_{\kappa=1}^{N_k} w_\kappa \left(\underbrace{J_\kappa^s(\mathbf{x})}_{\text{similarity term}} + \underbrace{J_\kappa^f(\mathbf{x})}_{\text{fitness term}} \right)$$

$$J_\kappa^f(\mathbf{x}) = \frac{\sum_{i=1}^{n} k\left(\left\| \dfrac{\mathbf{P}^\kappa - \tilde{\mathbf{P}}_i^\kappa}{h'} \right\|^2 \right) \boxed{E_\kappa(\mathbf{p}_i^\kappa)}}{\sum_{i=1}^{n} k\left(\left\| \dfrac{\mathbf{P}^\kappa - \tilde{\mathbf{P}}_i^\kappa}{h'} \right\|^2 \right)}$$

— FES



Track 2 Results

- **Experimental data:**
  - Two videos from "walsh_santomas"
- **Hand-labeled ground truth**: 1,356 frames, 32 objects, 1,760 tracking locations
- **Methods to compare with:**
  - mast [Tang et al. '16] (tracking by segmentation): Proposed segmentation w/ CMK tracking, state-of-the-art on NLPR_MCT benchmark (http://mct.idealtest.org/)
  - kalman [Chu et al. '11] (tracking by segmentation): Kalman-filtering tracking from foreground segmentation w/o multiple-kernel feedback
  - rnn [Milan et al. '17] (tracking by detection): First deep learning-based MOT method, state-of-the-art on MOT Challenge (https://motchallenge.net/)
  - sort [Bewley et al. '16] (tracking by detection): Fast online MOT based on rudimentary data association and state estimation techniques

Track 2 Results

1st rank labeled in red, 2nd rank labeled in blue

Methods	MOTA%	MOTP%	FAF	FP	FN	ID Sw.
cmk3d	82.0	99.5	0.23	7	310	0
mast	79.8	91.9	0.26	118	214	23
kalman	64.2	86.4	0.46	197	404	29
rnn	69.0	96.3	0.40	53	484	8
sort	61.8	99.1	0.50	13	629	30

- Standard metrics used in MOT Challenge benchmark:

MOTA (↑): Multiple Object Tracking **Accuracy**. This measure combines three error sources: false positives, missed targets and identity switches.
MOTP (↑): Multiple Object Tracking **Precision**. The misalignment between the annotated and the predicted bounding boxes.

FAF (↓): The average number of **false alarms** per frame.
FP (↓): The total number of **false positives**.
FN (↓): The total number of **false negatives** (missed targets).
ID Sw. (↓): The total number of **identity switches**.

Track 2 Demo

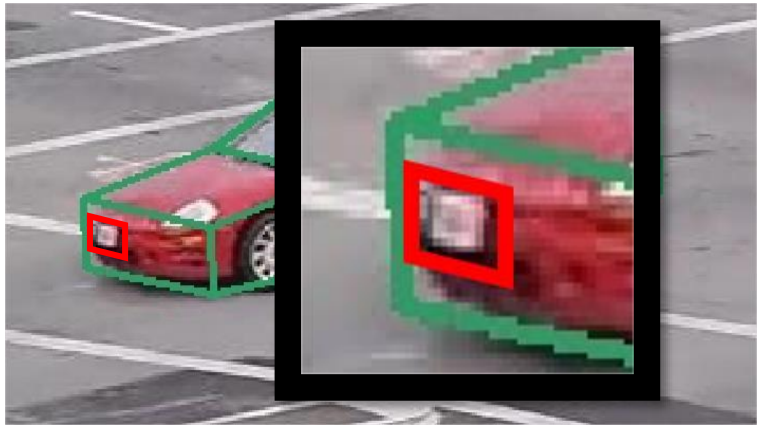
Track 2 Demo: Vehicle Orientation

Track 2 Demo: Mutual Occlusion

Track 2 Demo: AVSS2007 Benchmark

Conclusion

- Track 1
  - SSD + YOLO w/ multi-scale testing to improve detection of small objects
  - mAPs on aic480, aic1080 and aic540 are 0.34, 0.28 and 0.25 respectively.
- Track 2
  - Fully unsupervised 3D vehicle tracking and modeling assisted by camera self-calibration
  - Capable of overcoming strong occlusion
  - Outperforms both state-of-the-art of tracking by segmentation and tracking by detection
- Future work / other proposals
  - Feedback of vehicle types from 3D car modeling to object detection/classification
  - Extension to tracking/re-identification across multiple cameras
  - License plate identification based on 3D vehicle model



Future Work: Tracking across Cameras

Cam1

Cam2

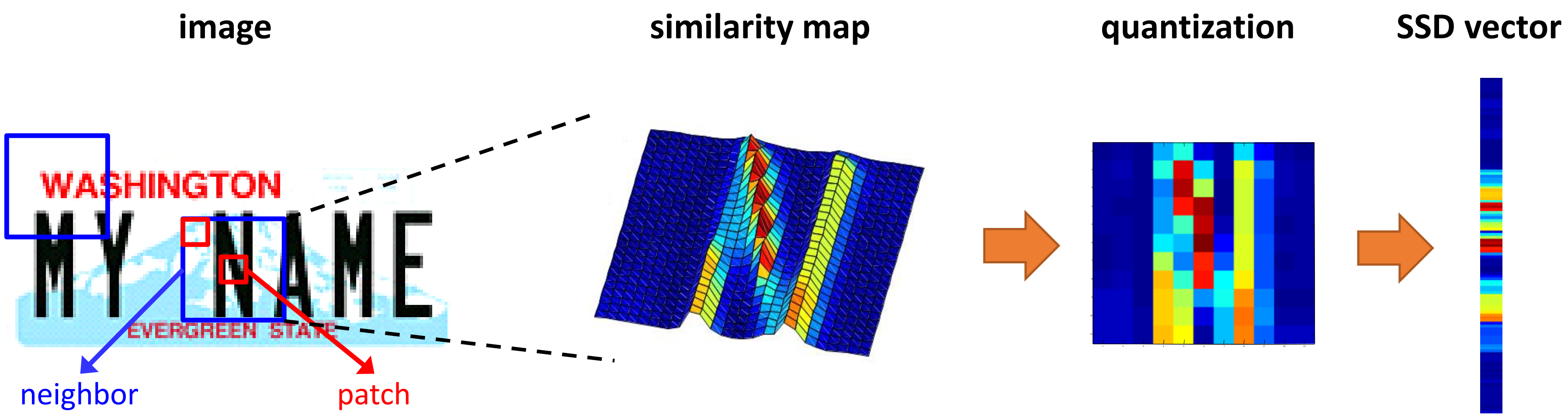
Cam3

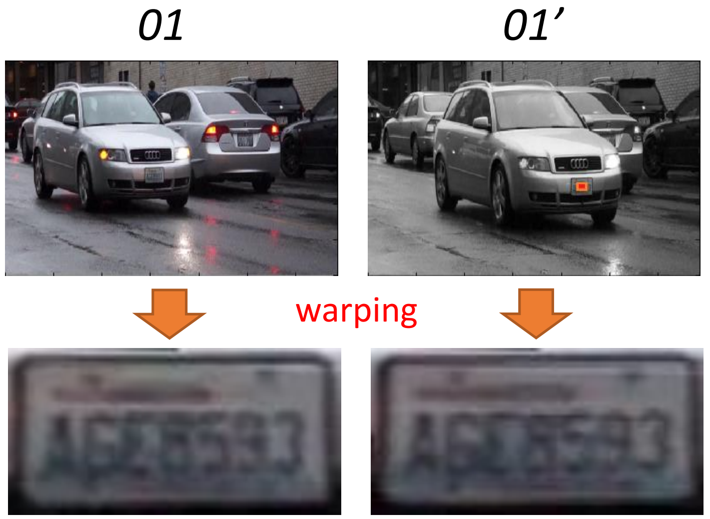
Future Work: License Plate Identification

- License Plate in surveillance camera
  - Not very clear, even hard to recognize
  - Conventional OCR can not perform well
    - color, edge, intensity, gradient, etc

- Self-Similarity Descriptor[Shechtman *et al.*, 2007]
  - Based on similarity layout between neighbors
    - Robust to color change, deformation & translation.

Self-similarity Descriptor





Performance Experiment:
10 datasets, each has a pair of extracted license plate.

TABLE I. SIMILARITY SCORE OF THE COMPARISON

	01	02	03	04	05	06	07	08	09	10
01'	0.7610	0.5736	0.5043	0.5568	0.5008	0.5171	0.5910	0.4938	0.5646	0.5636
02'	0.5862	0.7706	0.4839	0.4963	0.4841	0.5052	0.5365	0.4901	0.5341	0.5104
03'	0.4871	0.4580	0.7557	0.5070	0.5363	0.5133	0.5126	0.4336	0.4873	0.4990
04'	0.5949	0.5238	0.5818	0.7719	0.5530	0.5287	0.5994	0.5014	0.5446	0.56657
05'	0.5333	0.5279	0.5600	0.5400	0.7707	0.5519	0.5852	0.4910	0.5361	0.5165
06'	0.5039	0.4890	0.5385	0.4696	0.5544	0.8534	0.5527	0.4834	0.5398	0.5592
07'	0.5910	0.5147	0.5150	0.5569	0.5615	0.5718	0.7606	0.5292	0.5408	0.5271
08'	0.5052	0.4784	0.4617	0.5086	0.4990	0.5087	0.5420	0.7600	0.4994	0.4929
09'	0.5845	0.5235	0.4861	0.4762	0.5007	0.5382	0.5666	0.4730	0.8018	0.5613
10'	0.5410	0.4990	0.5022	0.5083	0.4977	0.5603	0.5362	0.4579	0.5805	0.8415