POSS: PKU Omni-directional Smart Sensing

URL: www.poss.pku.edu.cn

Towards Omni-directional Sensing and Reasoning in Real World Environment

Huijing Zhao, zhaohj@cis.pku.edu.cn

Key Lab on Machine Perception (MOE)
School of Electronics Engineering and Computer Science

Peking University

Where is POSS Team?

Peking University

School of Electronics Engineering and Computer Science

(386)

Department of Machine Intelligence

(30)

$\subset$

Key Lab on Machine Perception (MOE)

(70)

3DVCR (3D Visual Computing and Robotics Group)

(9)

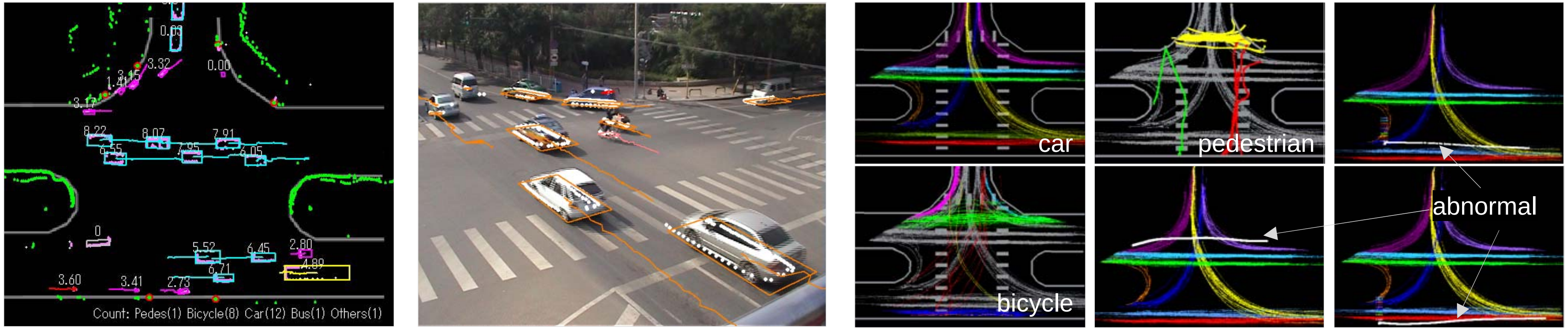
POSS (Intelligent Vehicle Team)

(1 faculty, 1 engineer, 6 PhD, 3 Master, 4 undergraduate students)

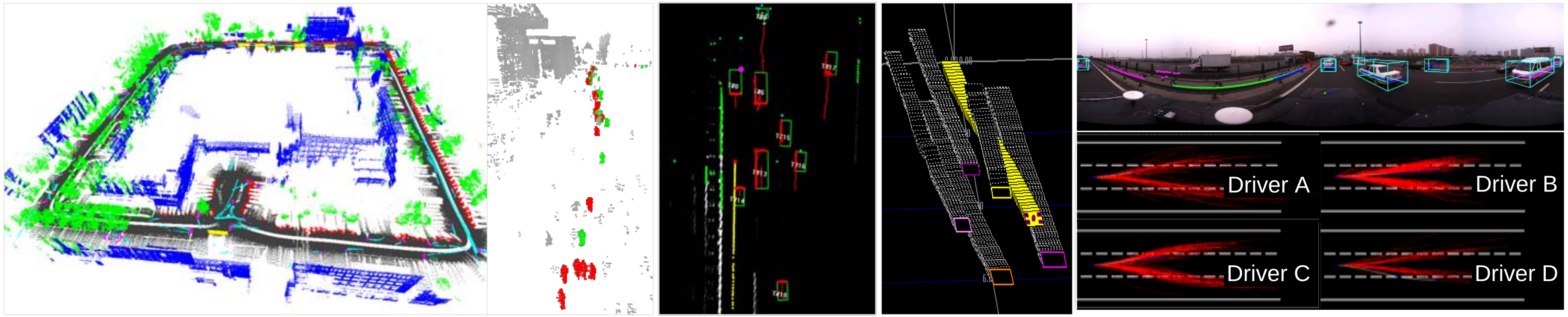
POSS Team

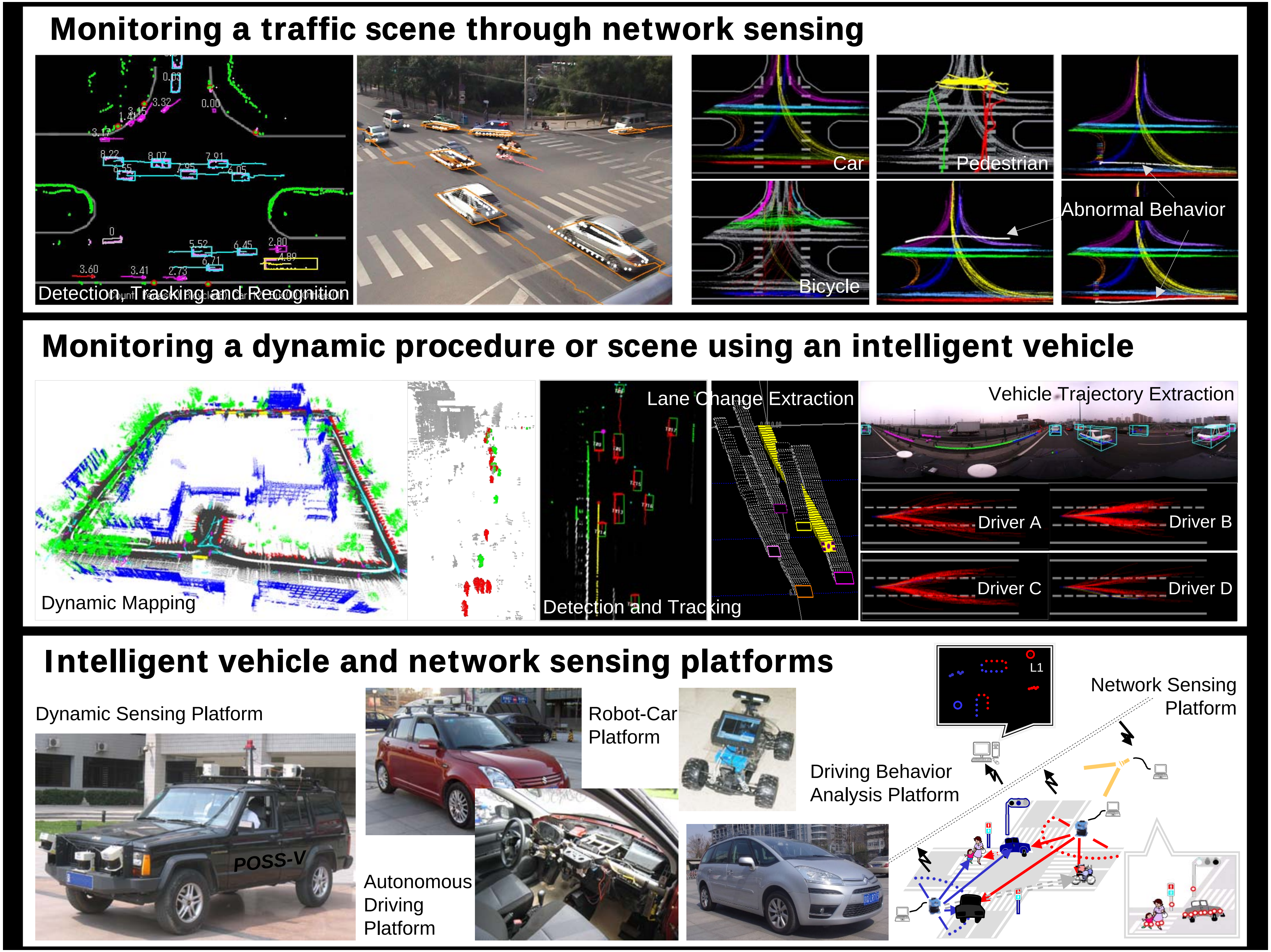
POSS Research

Monitoring a traffic scene through network sensing

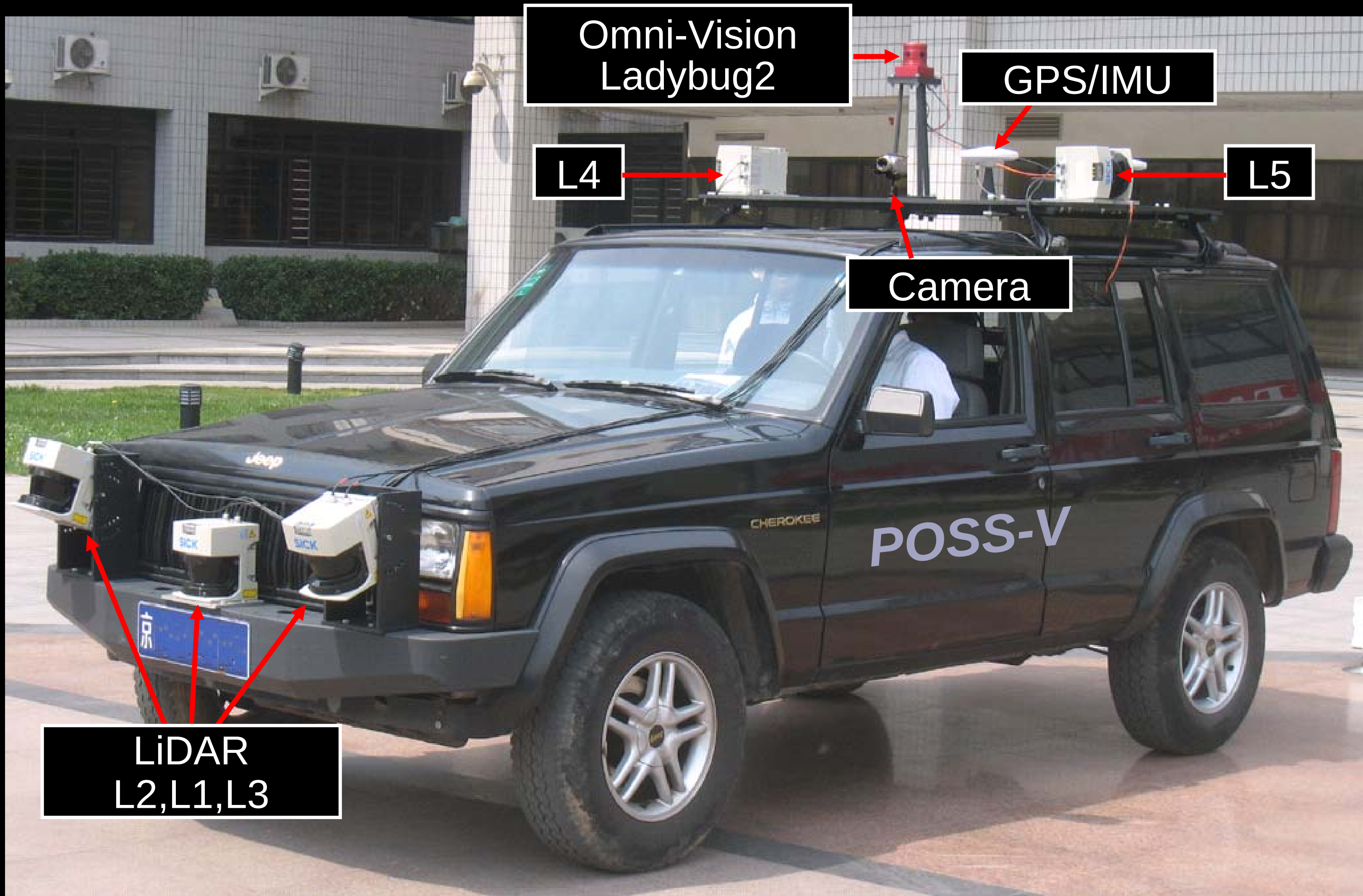


Monitoring a dynamic procedure or scene using an intelligent vehicle




Monitoring a traffic scene through network sensing
Detection, Tracking and Recognition
Car
Pedestrian
Bicycle
Abnormal Behavior
Monitoring a dynamic procedure or scene using an intelligent vehicle
Dynamic Mapping
Lane Change Extraction
Detection and Tracking
Vehicle Trajectory Extraction
Driver A
Driver B
Driver C
Driver D
Intelligent vehicle and network sensing platforms
Dynamic Sensing Platform
POSS-V
Robot-Car Platform
Autonomous Driving Platform
Driving Behavior Analysis Platform
Network Sensing Platform
L1

A Dynamic Sensing Platform



2008~

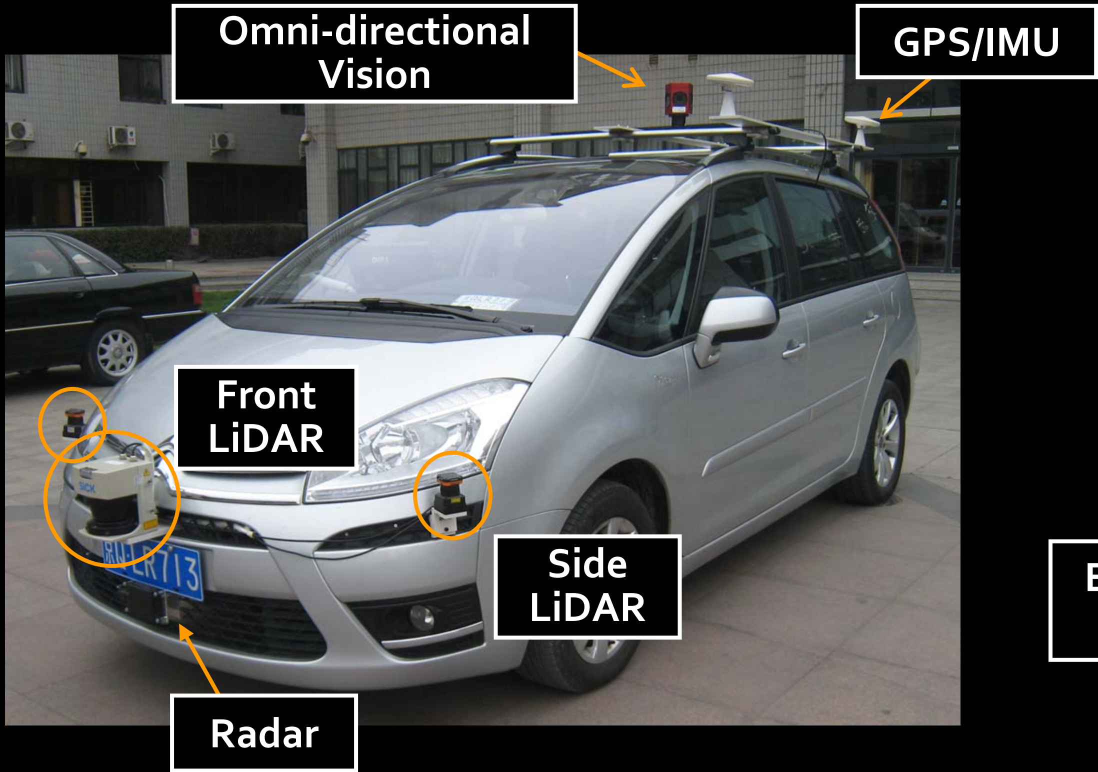
An Autonomous Driving Platform

2010~

demo1

demo2

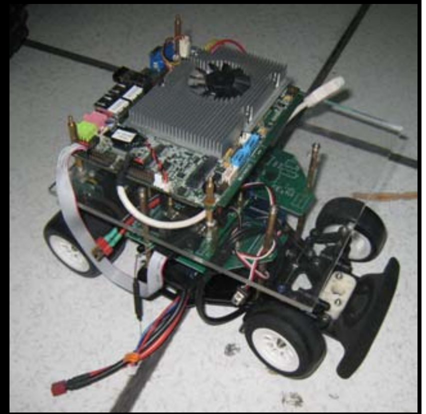
A PSA Platform



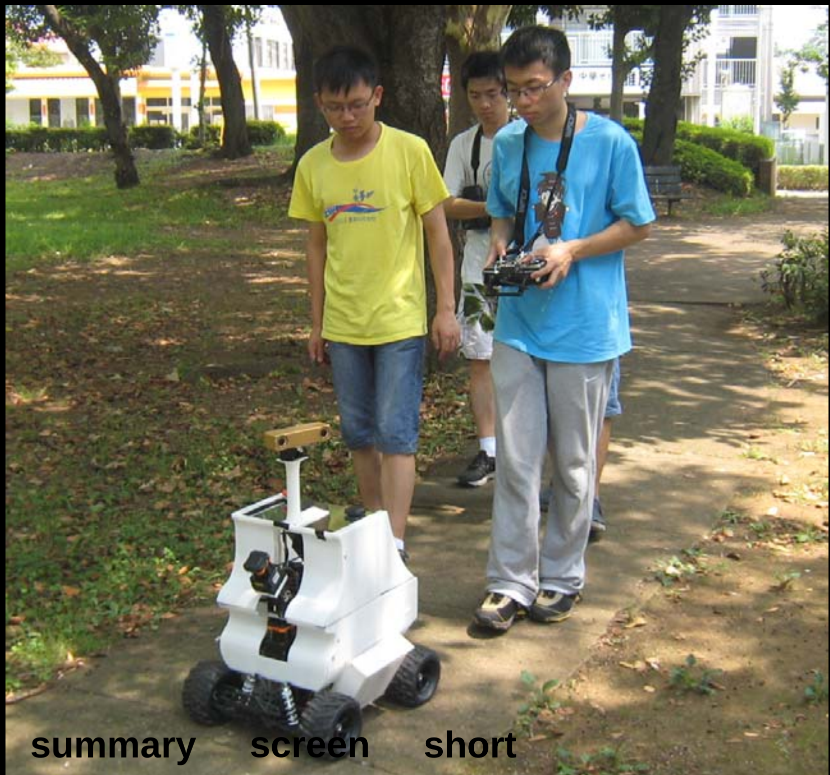
On-road data collection for driving behavior learning and reasoning

2013~

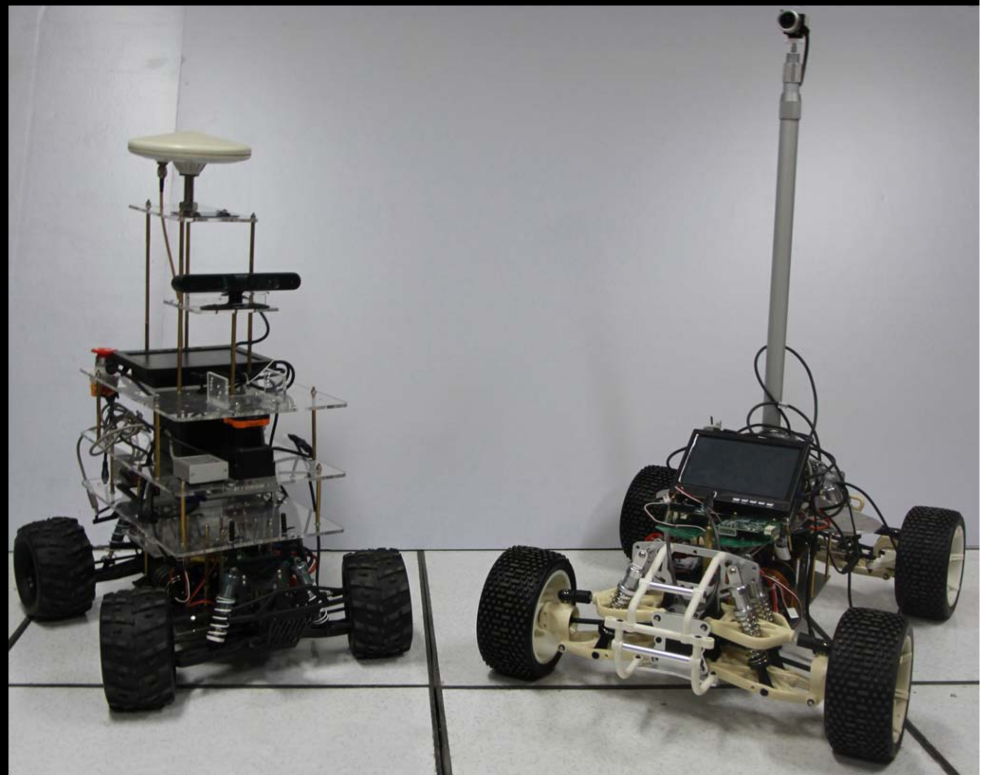
Robotic Platforms



miny1
miny2



Tsukuba Robotic Challenge 2013-14



POSS-MINYs 2013~

Our Goal

We focus on perception and reasoning techniques of intelligent vehicle.

We want to develop an intelligent vehicle of Omni-directional eyes perceiving an environment of both static and dynamic objects.

We want to reason based on real world sensing data, so that to aware situations and predict potential risks.

We want to map dynamic environments, which contains 3D geometry, semantics and scene dynamics.

We want to study potential applications in car navigation, traffic simulation, surveillance, etc.

Key Issues

- Sensor Alignment HMW
- Localization YYF
- 3D Mapping POI
- Mobile Object Detection and Tracking LIDAR Visual_IROS12 Visual_IROS14
- Scene Understanding FYK ICRA10
- Behavior Analysis YW DMARP LYB
- … XWD NXT

On-road Vehicle Trajectory Collection

Huijing Zhao, Chao Wang, Yubin Lin, Wen Yao,
Jinshi Cui, Hongbin Zha
Key Lab of Machine Perception, Peking University

Objective

- Developing an automated system to collect the synchronized motion trajectories that characterize the full course of driving maneuvers in real-world traffic scene.

Real driving data collection and analysis

Driver
(Driver's operations)

Vehicle
(Location, heading, speed etc.)

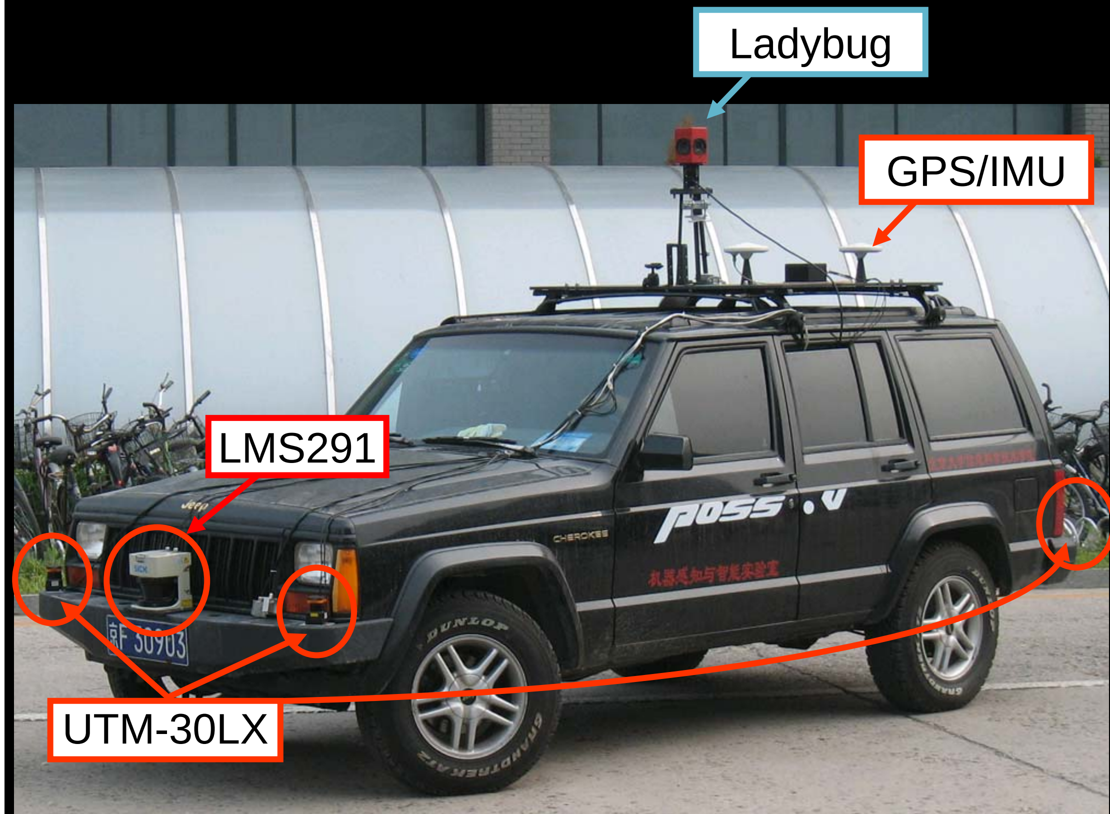
Driving behavior
(Lane change, overtaking, headway management etc.)

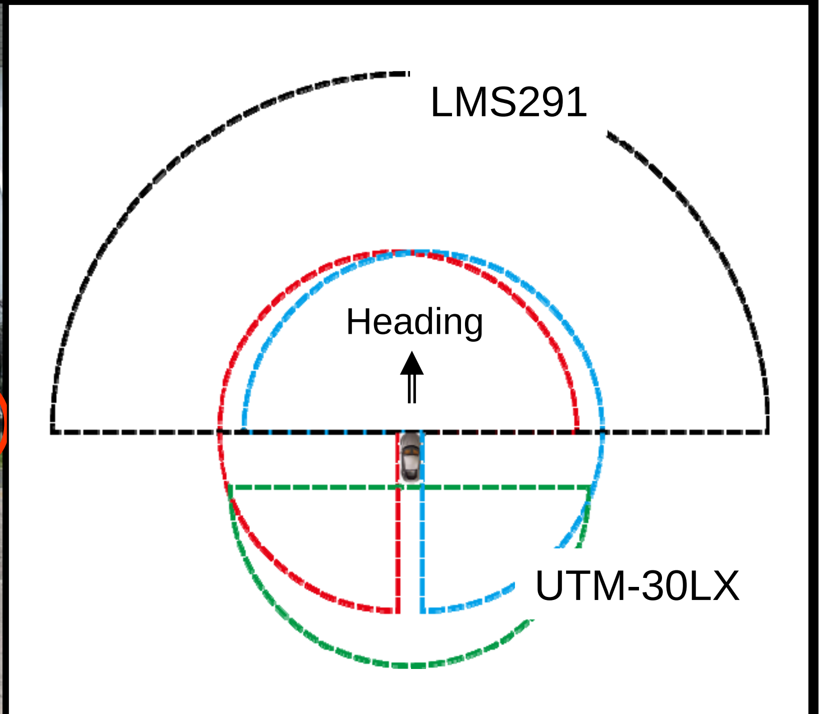
Driving Context
(Road environment and traffic participants)

Contributions

- A system is developed to collect the vehicle trajectories through on-road driving an instrumented vehicle with multiple 2D-LIDARs.
- A method of simultaneous mapping with vehicle detection and tracking (SMVDT) is developed to estimate the trajectories of environmental vehicles through multi-lidar data processing.

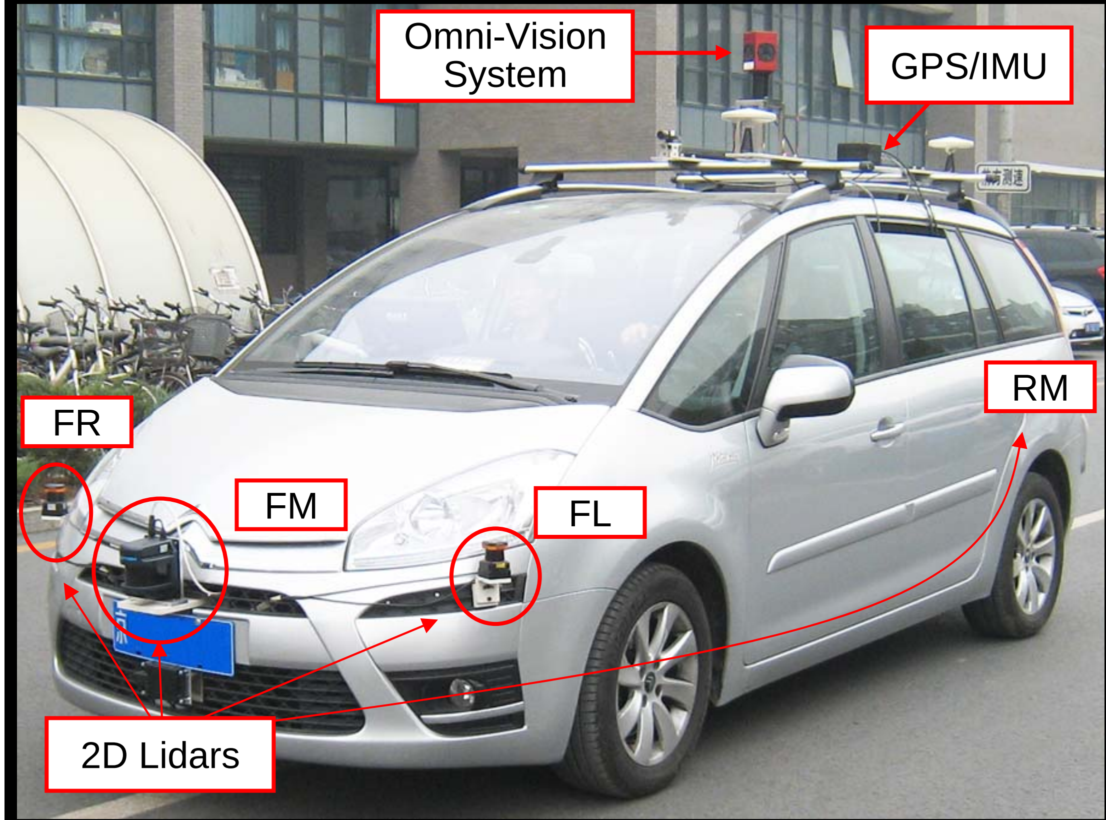
Sensor Setting

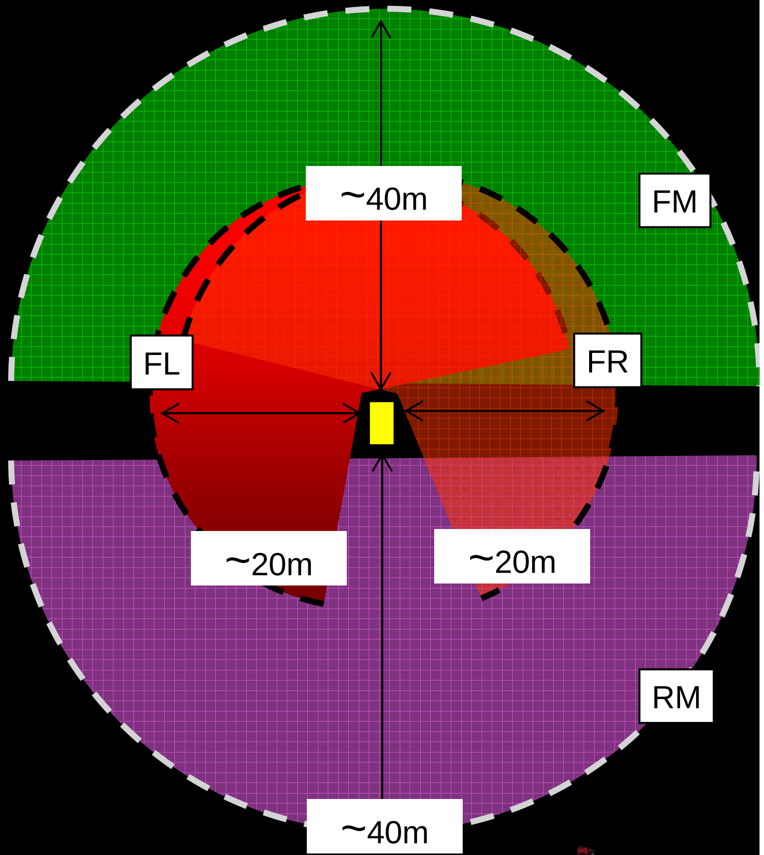




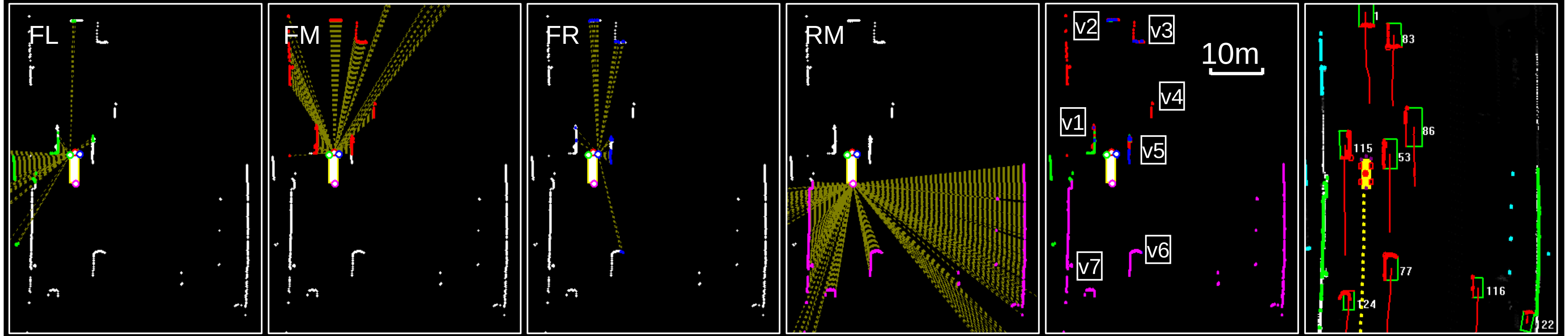
In this paper, the Ladybug is for visualization only.

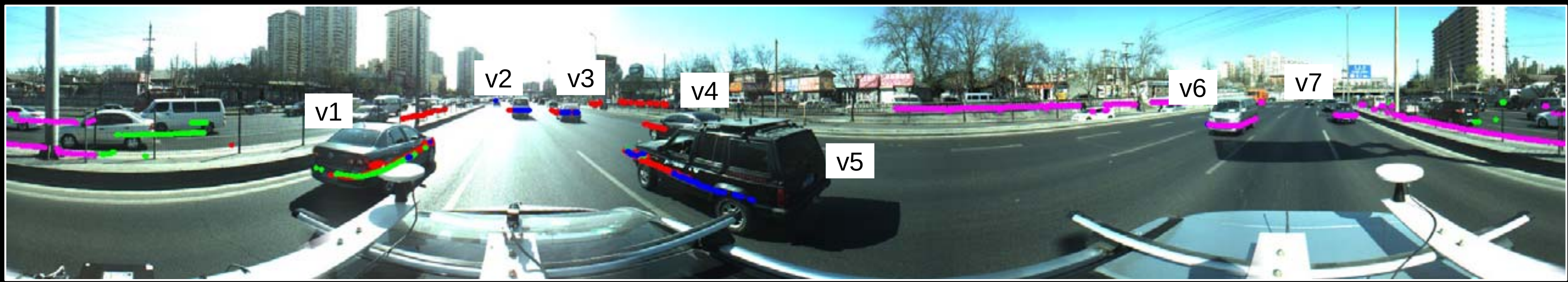
Sensor Setting

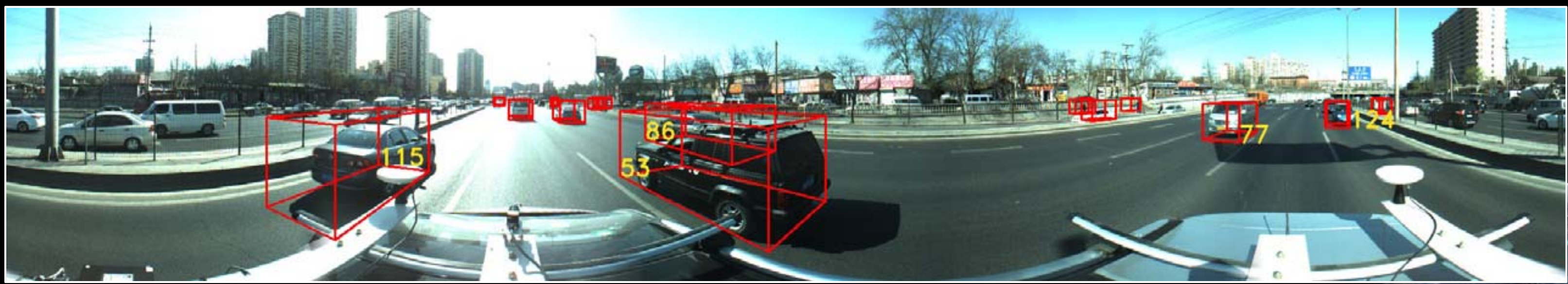




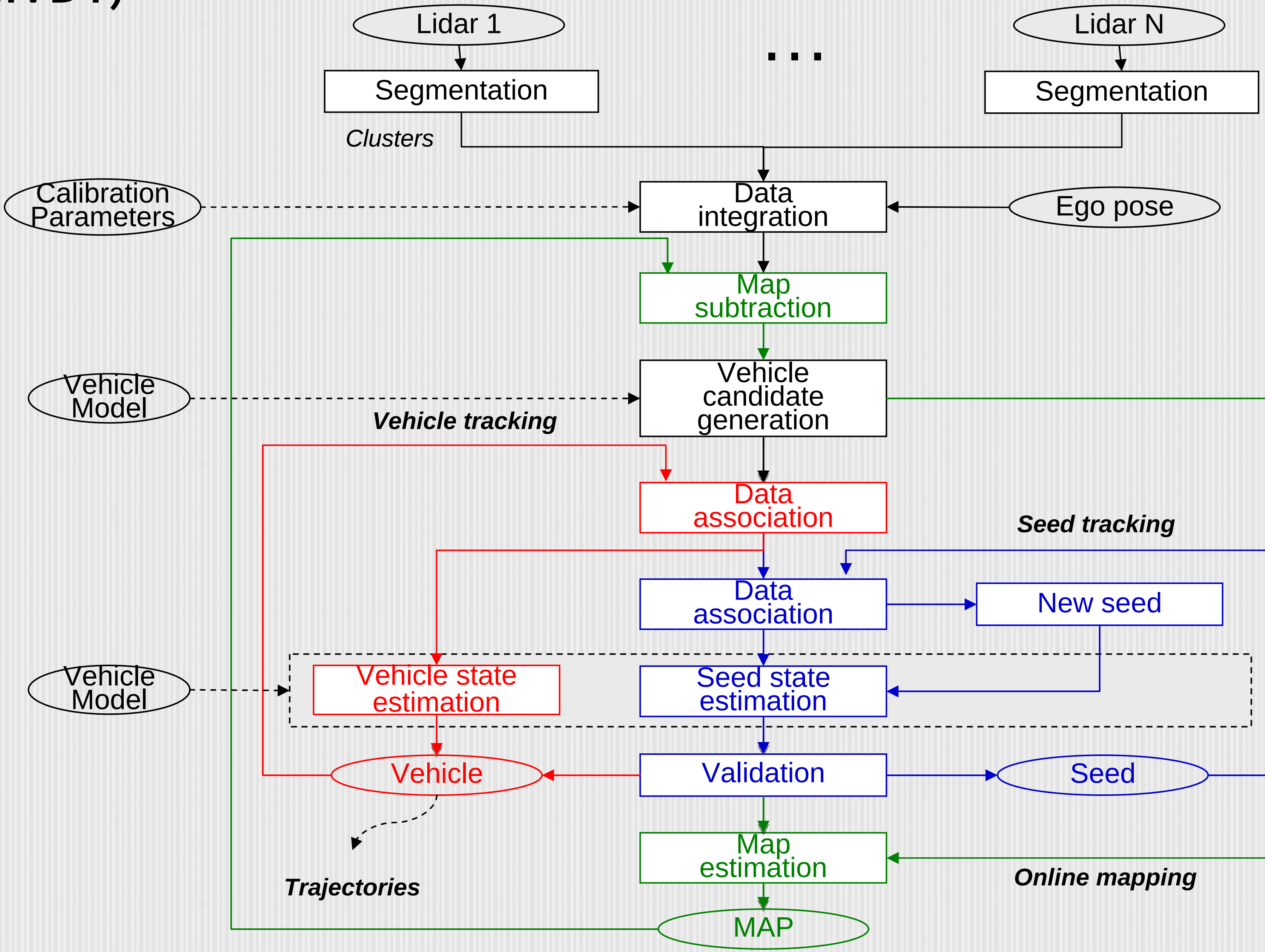
Fusion of Multi-Lidar Data







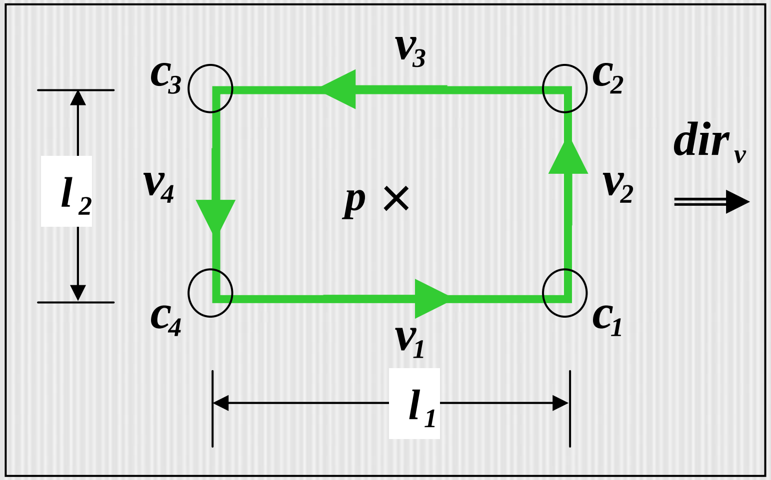
Simultaneous Mapping with Vehicle Detection and Tracking (SMVDT)



Vehicle Model vs Partial Observations

<u>Accounting partial observations:</u>

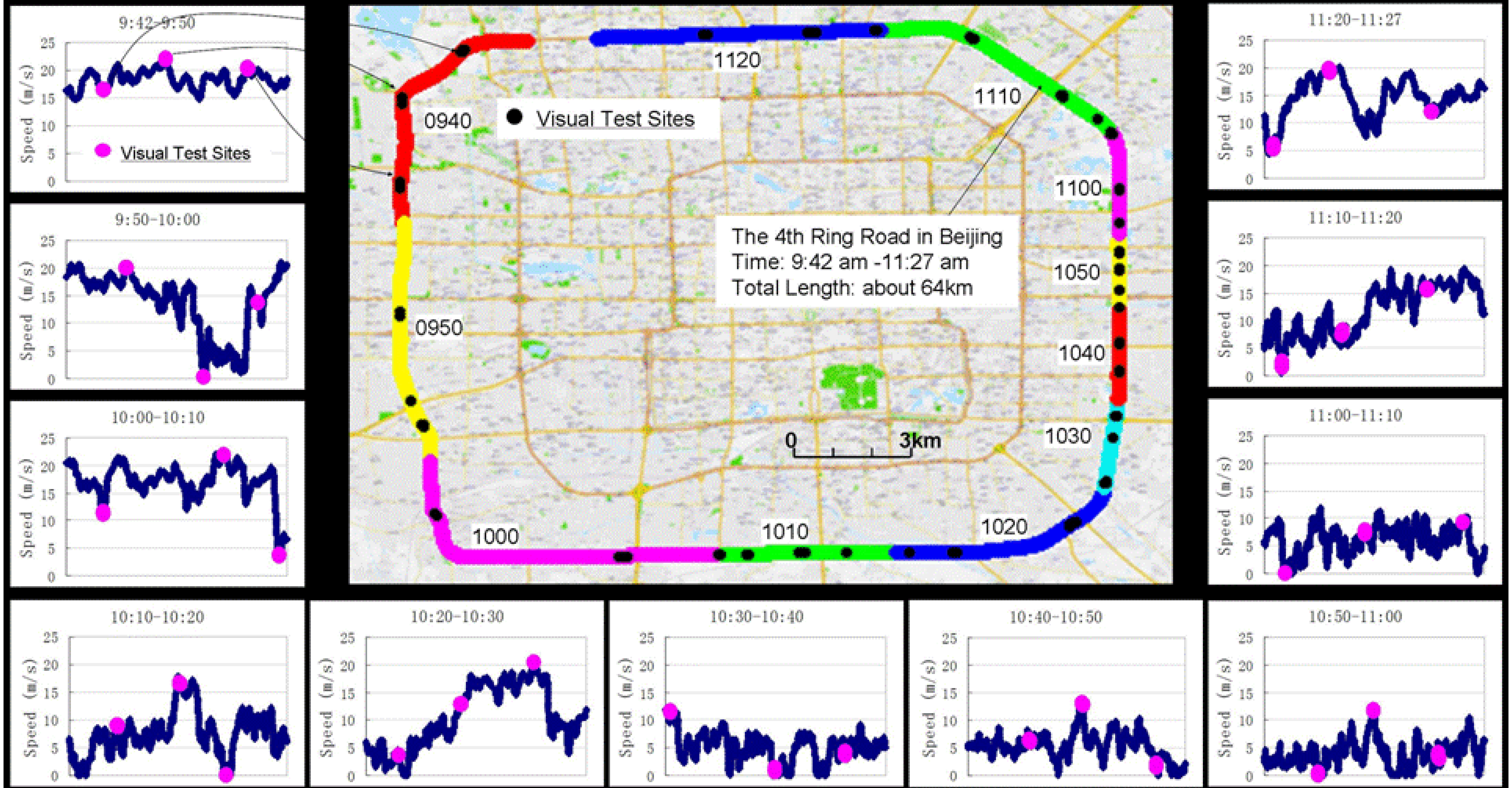
1) Reliability items are defined to denote whether the corresponding features are estimated on direct observations or inferred through the assumption on vehicle model.
2) Reliability items are accounted in data association and vehicle track estimation.



Vehicle Model

Item	Feature	Reliability
directional vectors ($i = 1, .., 4$)	v_i	rv_i
corner points ($i = 1, .., 4$)	c_i	rc_i
a center point	p	rp
lengths on two vertical edges ($i = 1, 2$)	l_i	rl_i

Experimental Setting

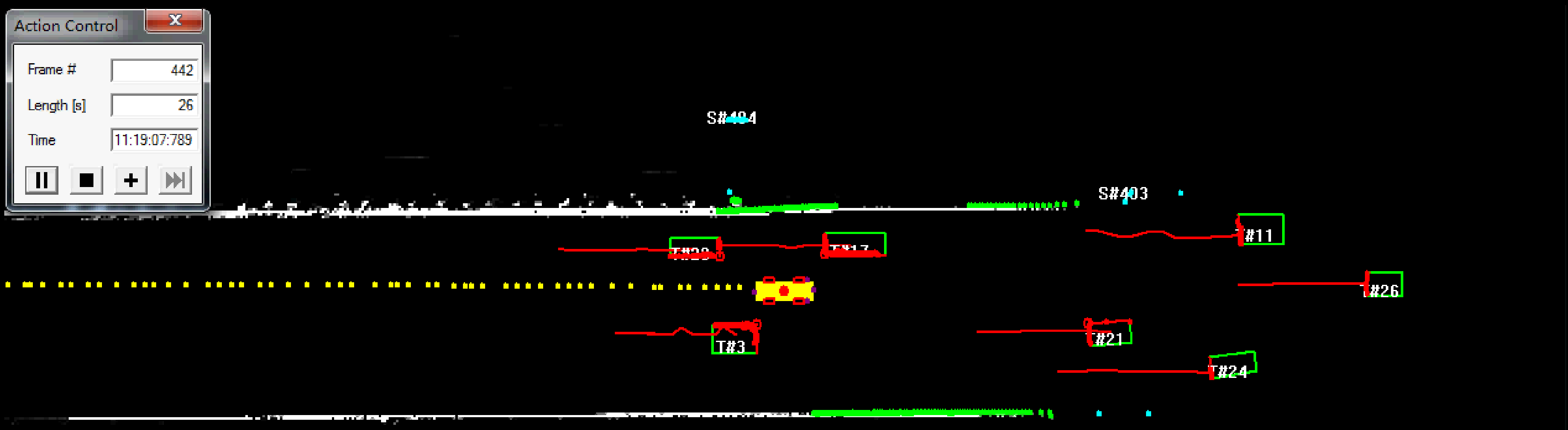


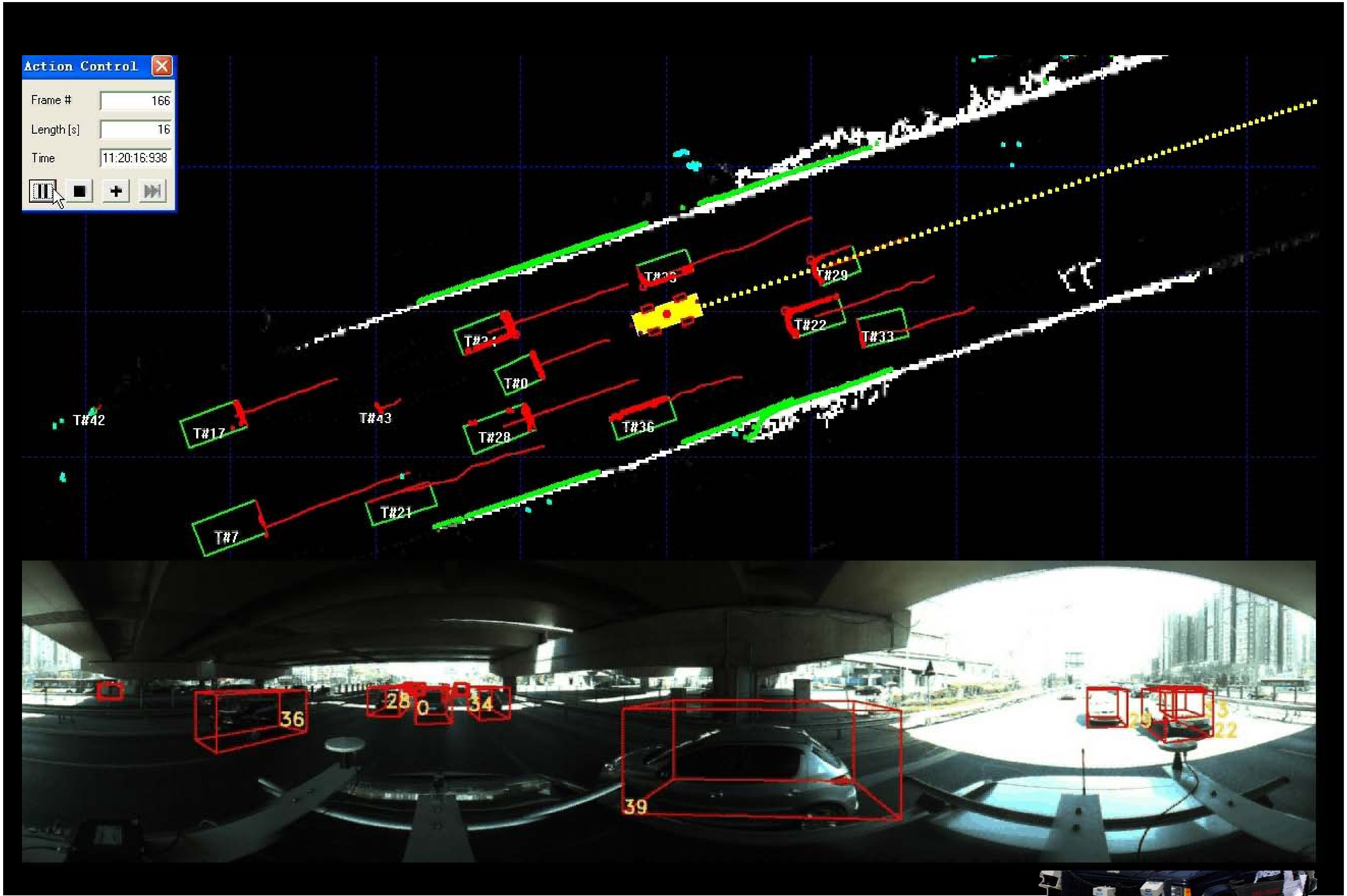
PEKING UNIVERSITY

POSS

PKU OMNI SMART SENSING

Online Multi-Lidar vs A Front Radar

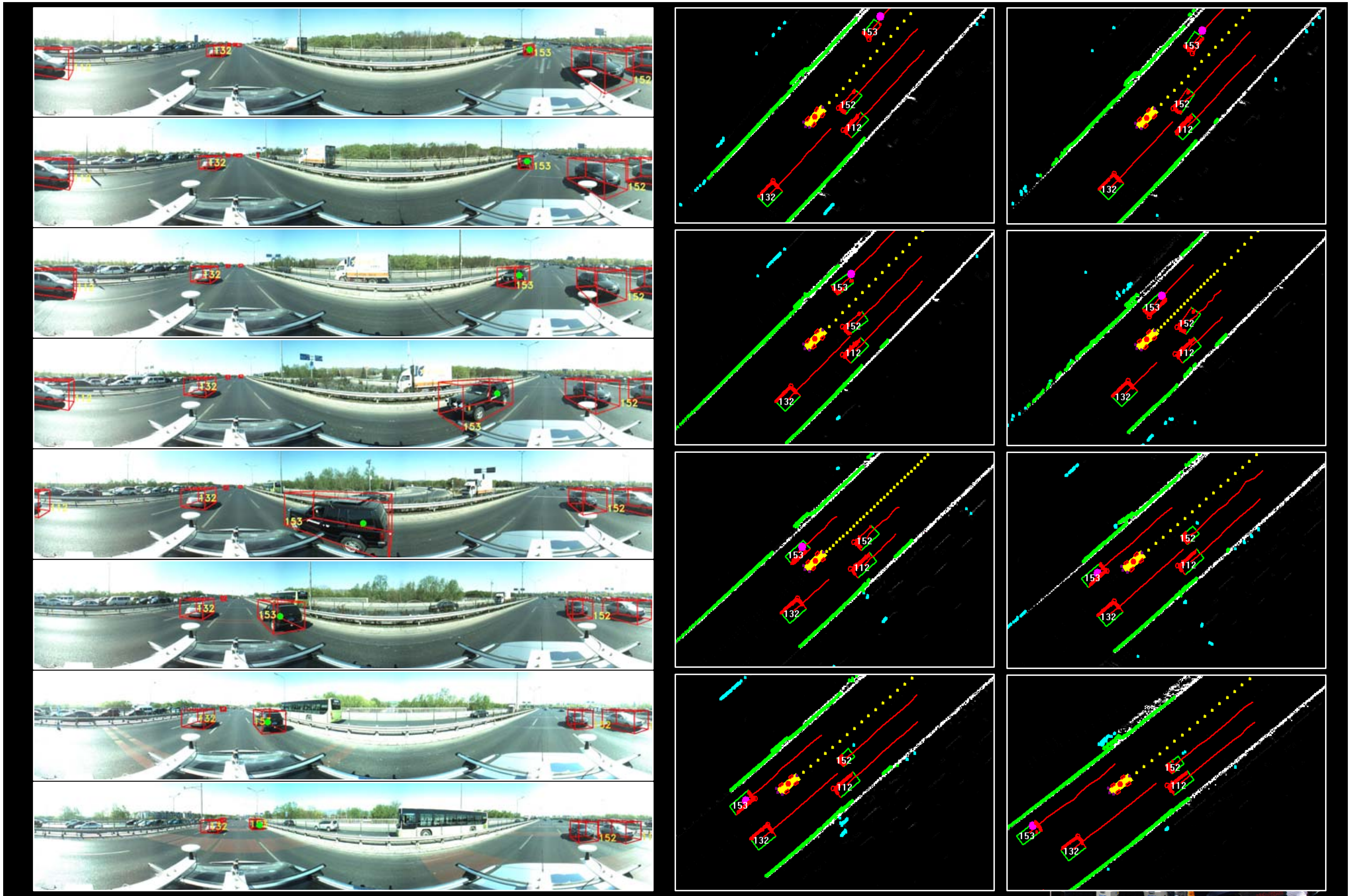




PEKING UNIVERSITY

POSS

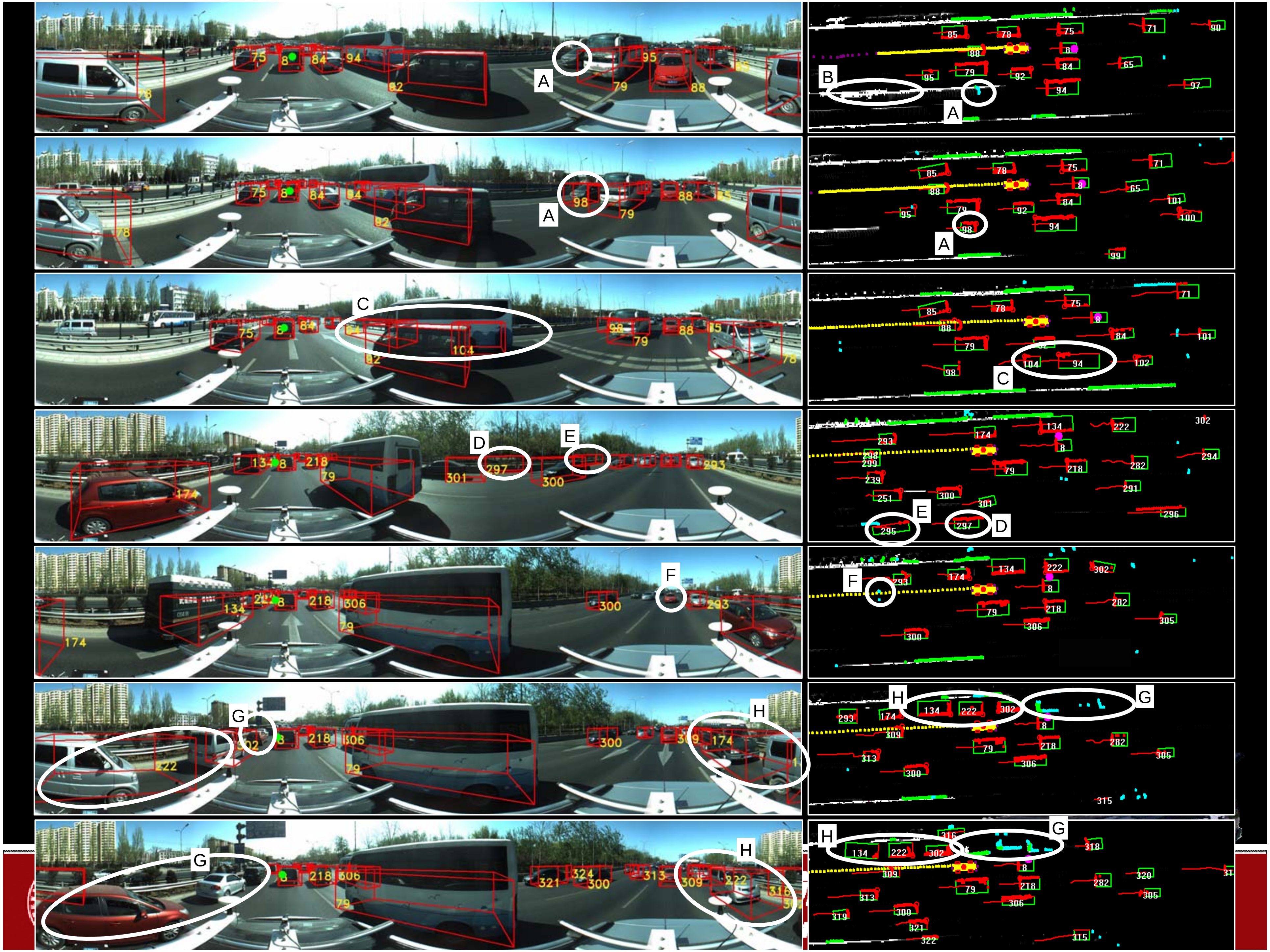
PKU OMNI SMART SENSING

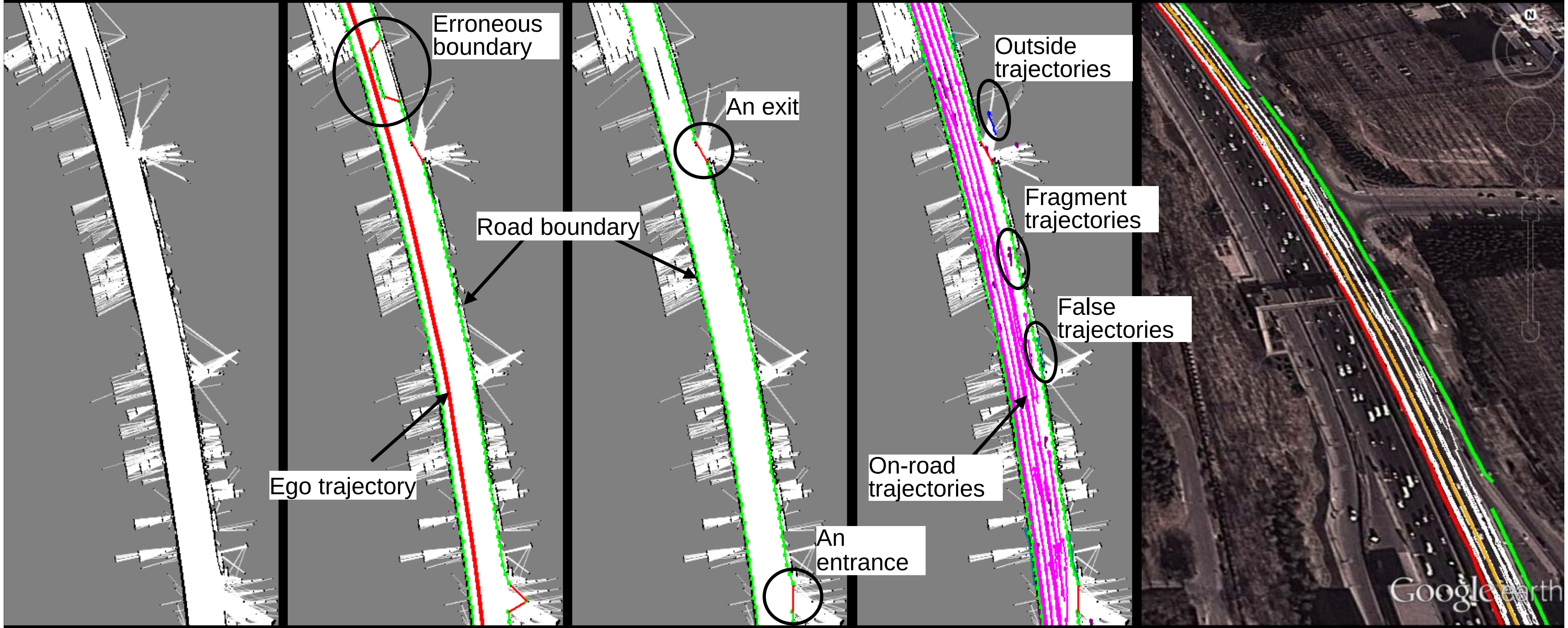


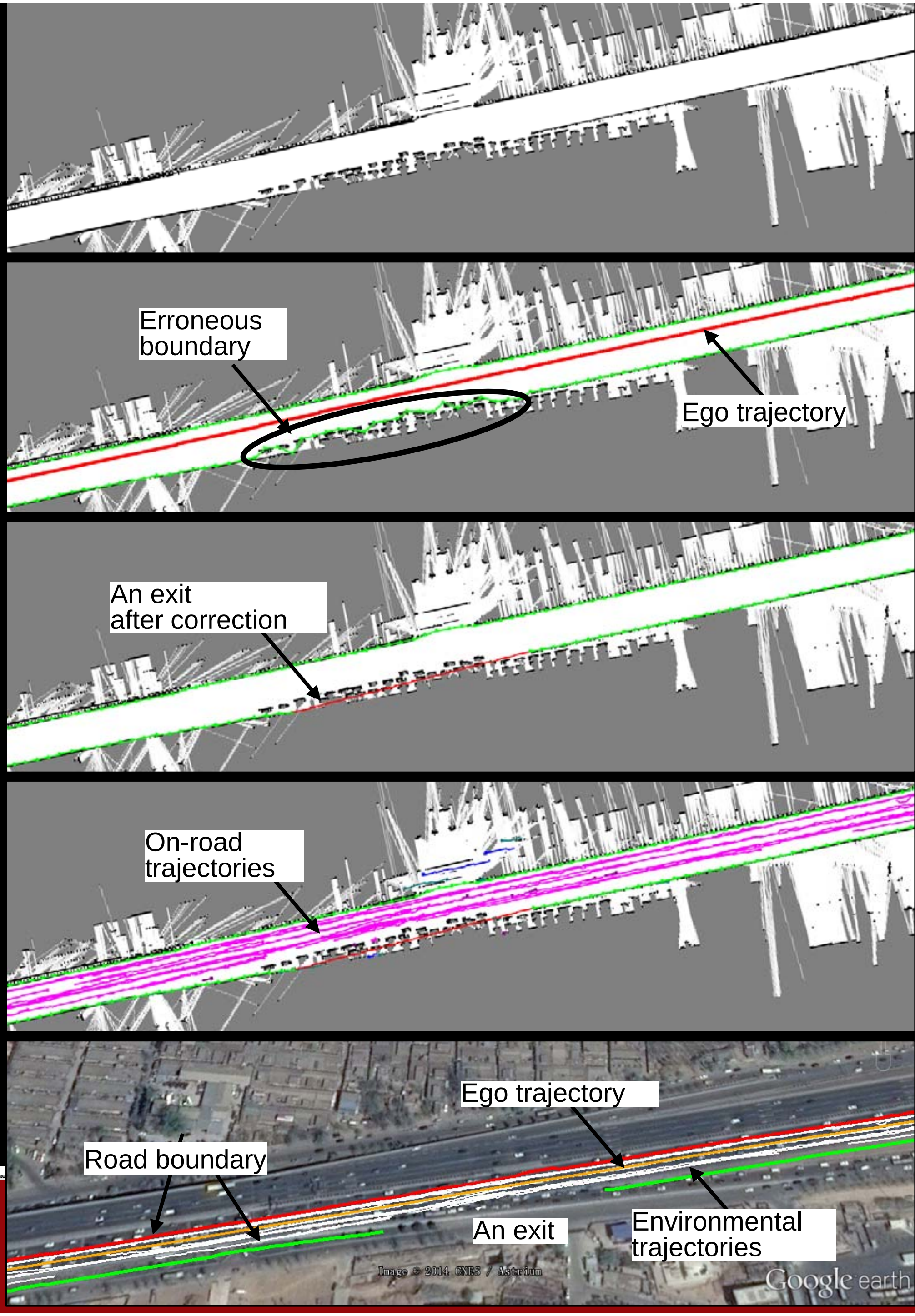
PEKING UNIVERSITY

POSS

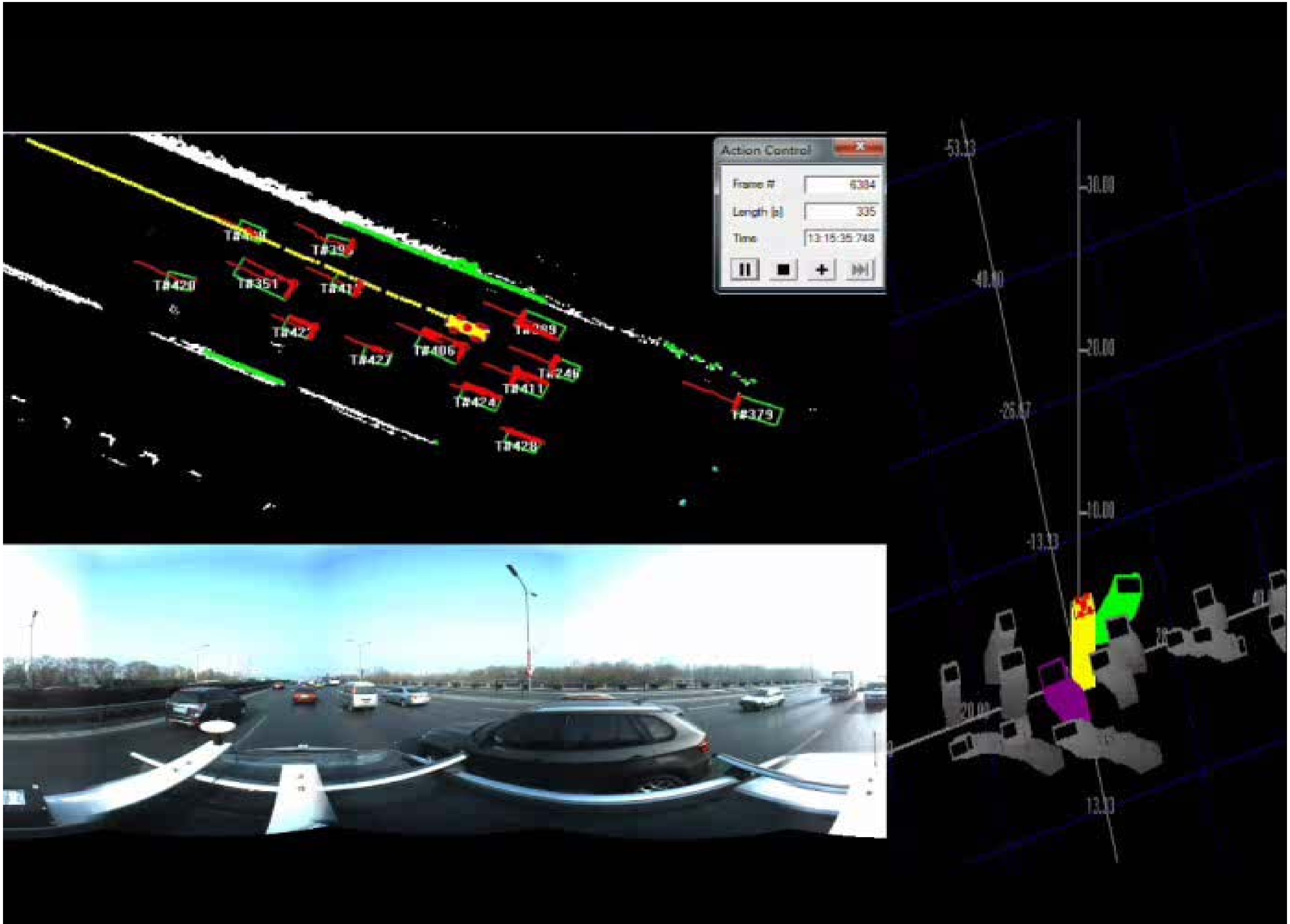
PKU OMNI SMART SENSING


A
B
C
D
E
F
G
H


Erroneous boundary
An exit
Outside trajectories
Fragment trajectories
Road boundary
False trajectories
Ego trajectory
On-road trajectories
An entrance
Google earth



PEKING UNIVERSITY


Action Control
Frame #
6384
Length (s)
335
Time
13:15:35:748
T#351
T#420
T#406
T#424
T#379
T#428
T#411
30.00
20.00
10.00
40.00
13.33

IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS) 2012

A System of Automated Training Sample Generation for Visual-based Car Detection

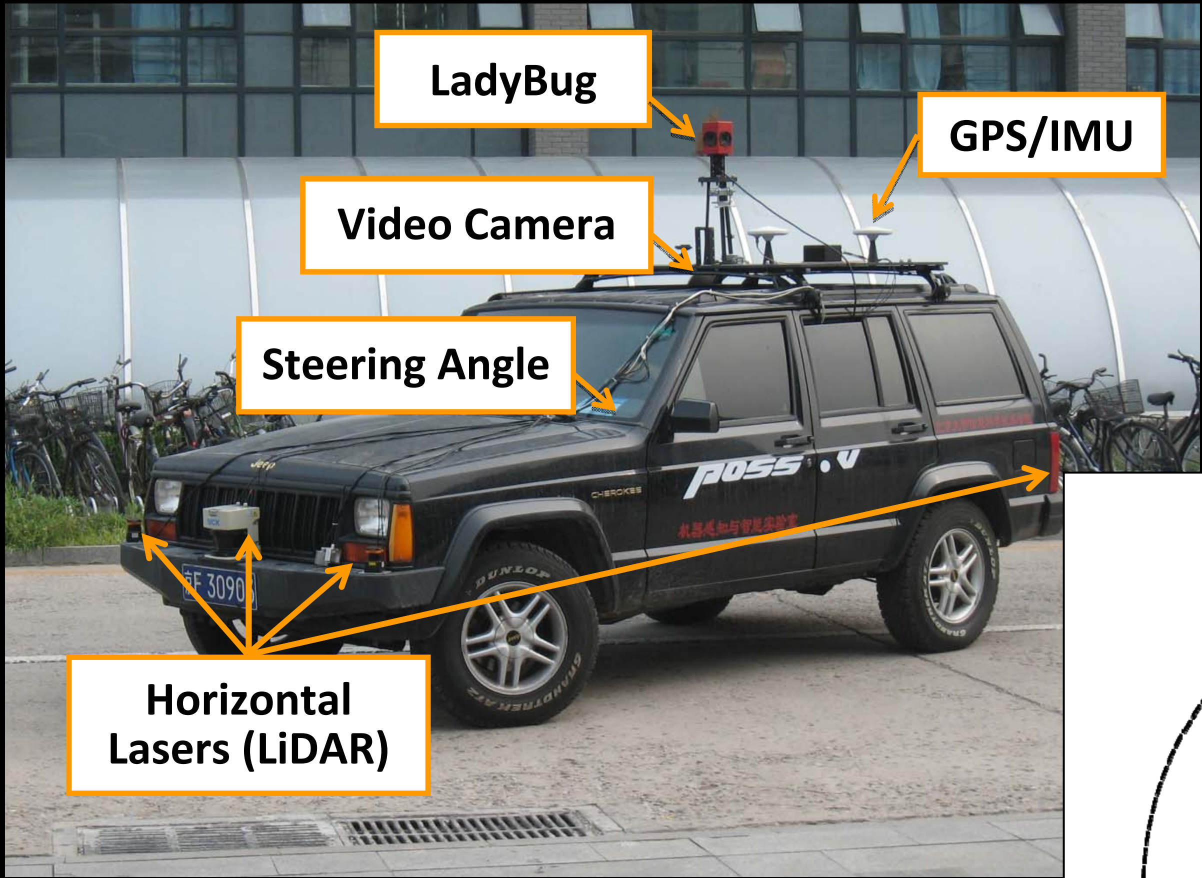
C. Wang, H. Zhao, F. Davoine, H. Zha

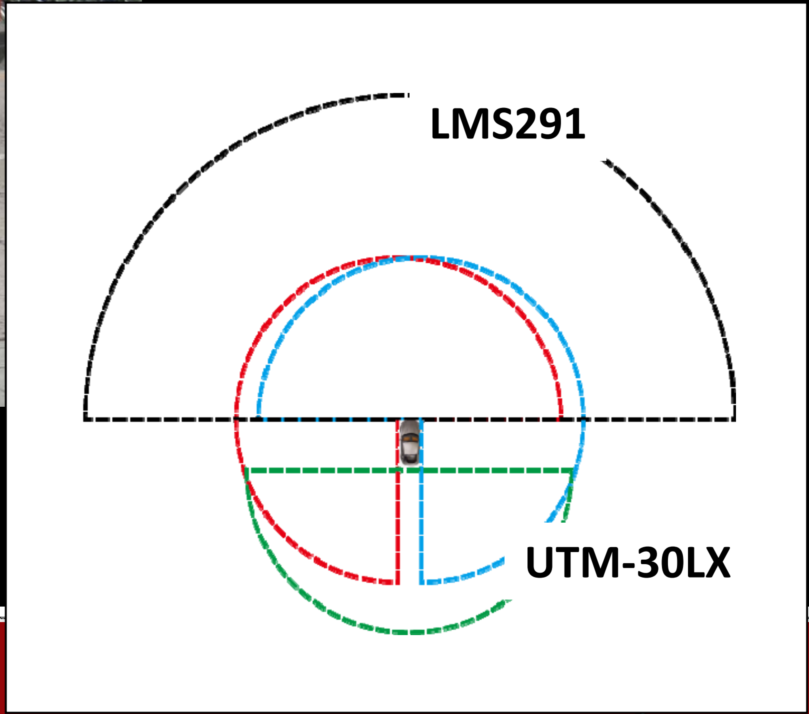
Key Lab of Machine Perception, Peking University
CNRS and LIAMA Sino French Laboratory

PEKING UNIVERSITY

POSS
PKU OMNI SMART SENSING

PKU-KLMP Platform





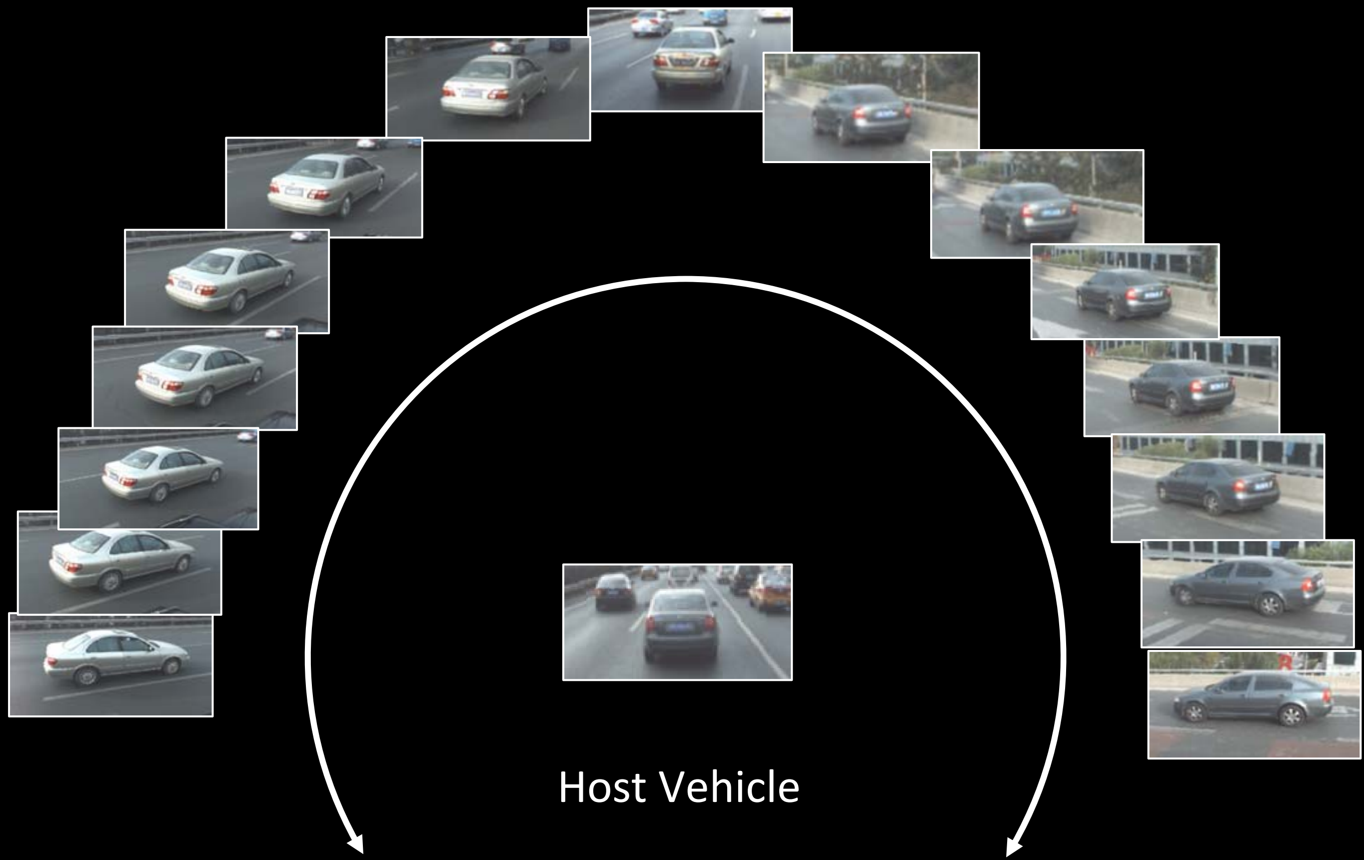
Training sample collection for visual-based car detector

Omni-Laser & GPS Data → Laser-based Car Detection and Tracking

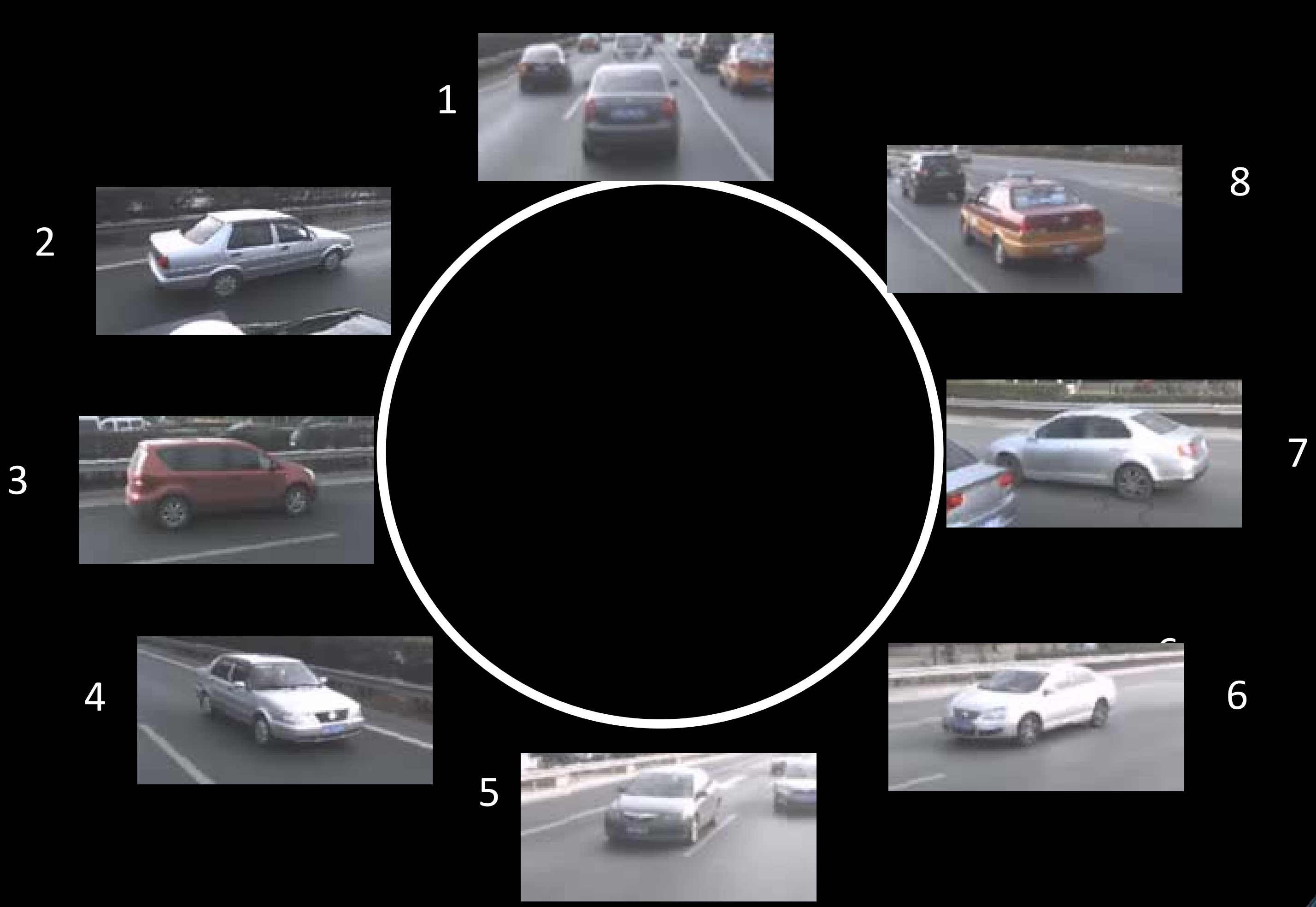
Omni-Image → Laser-visual Fusion

Laser-based Car Detection and Tracking → Laser-visual Fusion → Visual Sample Extraction → Visual Classifier Training

Duplicated Image Samples Removal



Orientation-based Sample Categorization



Results in Automatic Training Sample Extraction

IROS12

1 2 3 4 5 6 7 8

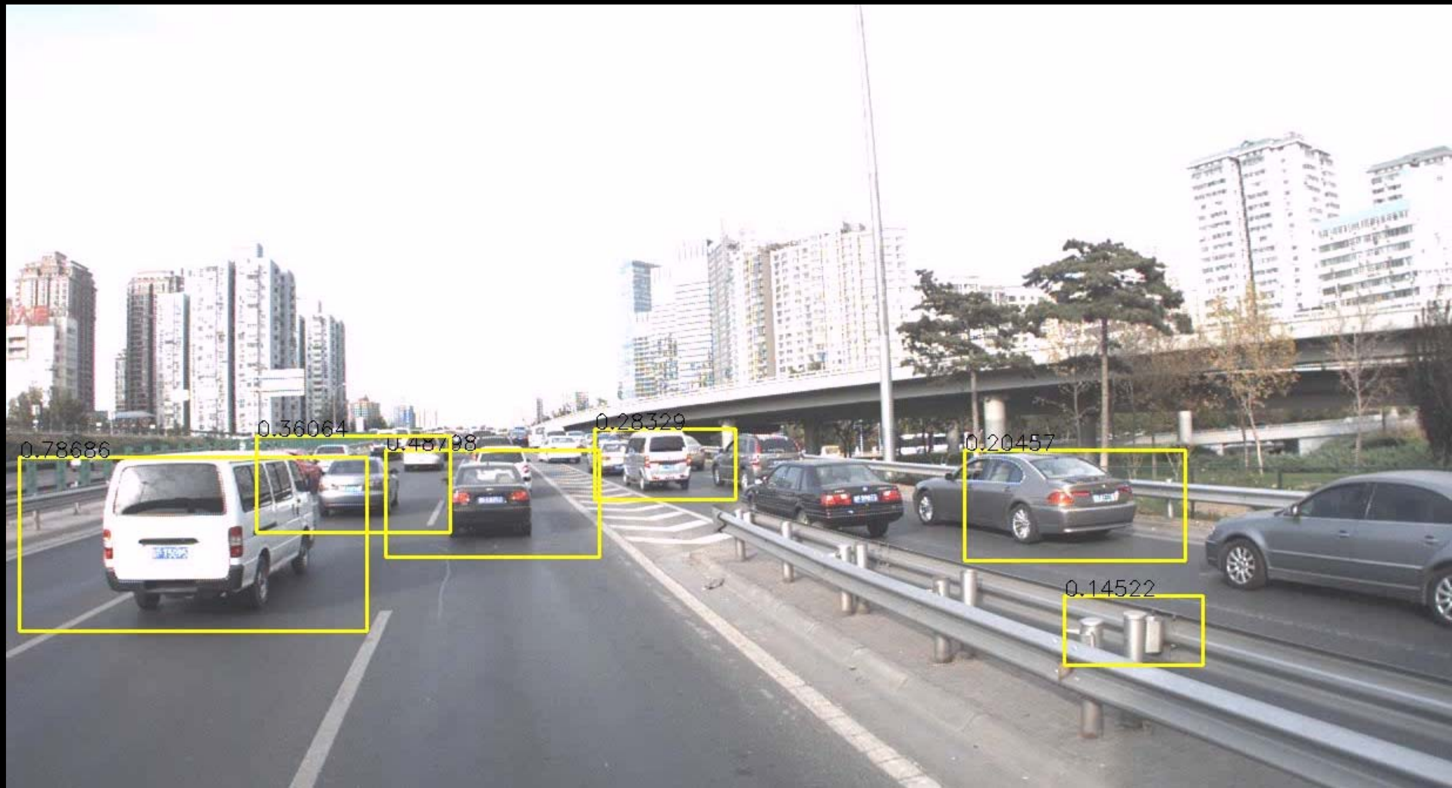
Results in Automatic Training Sample Extraction

- Vehicle runs 48.26km for 40 min
- 1526 trajectories， 5399 cars samples
- Sample details

Sub-categorization	1	2	3	4	5	6	7	8
Right samples	525	848	359	836	621	852	475	883
Wrong samples	10	21	13	28	11	33	21	19

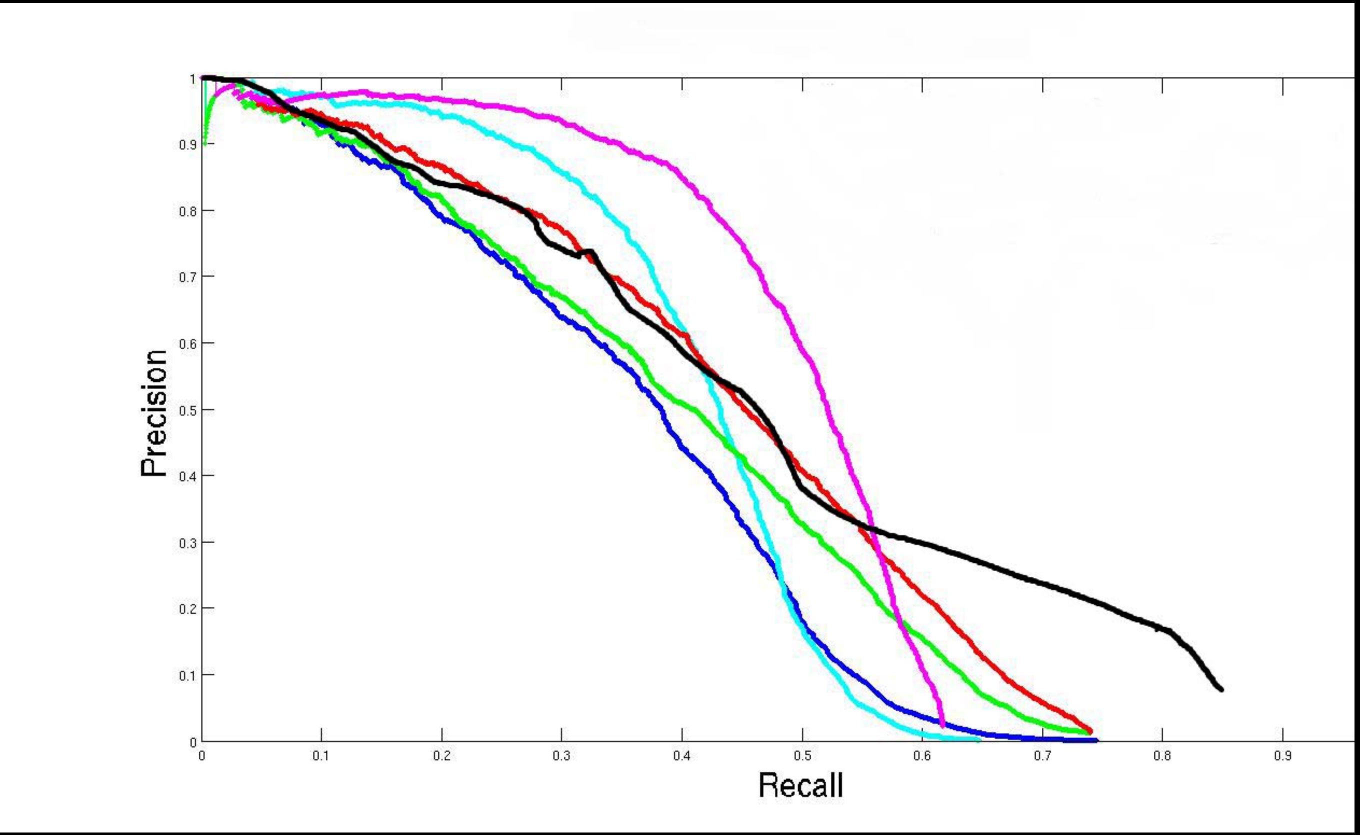
Training Samples Validation

- Ring Road training samples & Ladybug front camera



Training Samples Validation

- Ring Road training samples & Ladybug front camera



IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 20

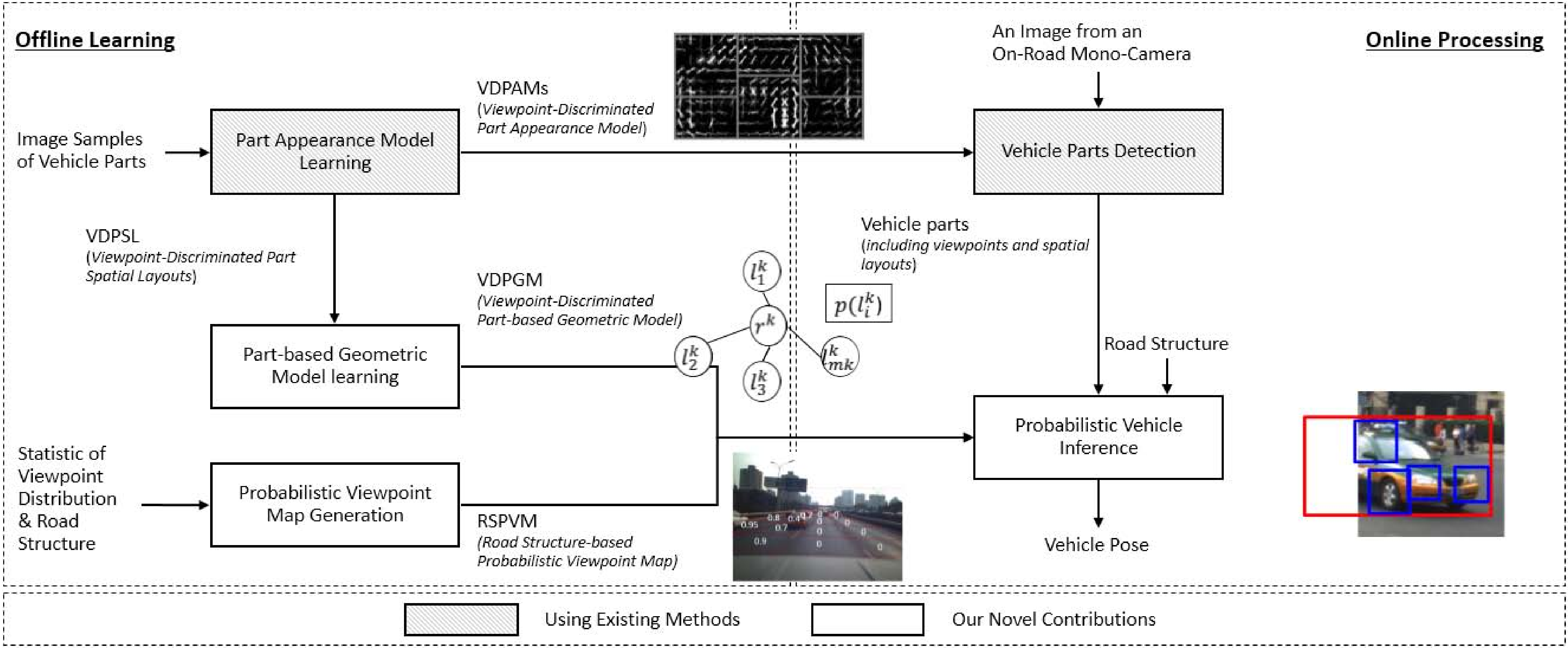
On-road Vehicle Detection through Part Model Learning and Probabilistic Inference

Chao Wang, Huijing Zhao, Chunzhao Guo, Seiichi Mita, Hongbin Zha

Motivation & Objective

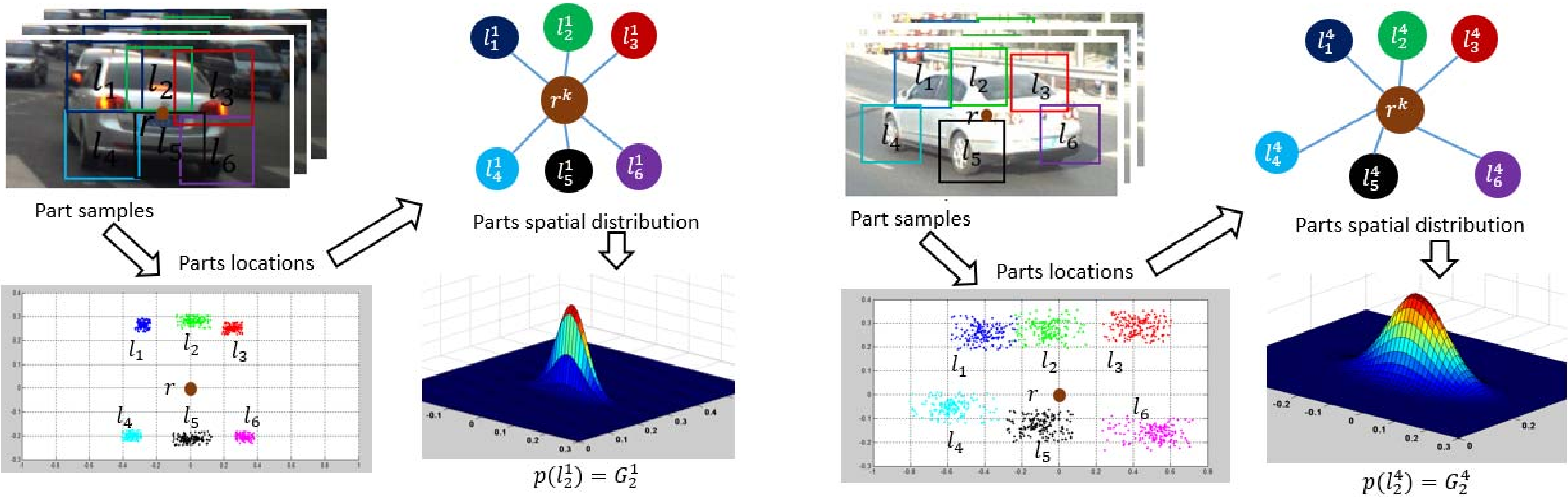
- Achieve robust on-road vehicle detection
  - Various appearance with different viewpoint
  - Occlusions

Framework



Offline Learning
Online Processing
Image Samples of Vehicle Parts
Part Appearance Model Learning
VDPAMs
(Viewpoint-Discriminated Part Appearance Model)
An Image from an On-Road Mono-Camera
Vehicle Parts Detection
VDPSL
(Viewpoint-Discriminated Part Spatial Layouts)
VDPGM
(Viewpoint-Discriminated Part-based Geometric Model)
Vehicle parts
(including viewpoints and spatial layouts)
$p(l_i^k)$
l_1^k
r^k
l_2^k
l_3^k
l_{mk}^k
Part-based Geometric Model learning
Road Structure
Probabilistic Vehicle Inference
Statistic of Viewpoint Distribution & Road Structure
Probabilistic Viewpoint Map Generation
RSPVM
(Road Structure-based Probabilistic Viewpoint Map)
Vehicle Pose
Using Existing Methods
Our Novel Contributions

VDPGM

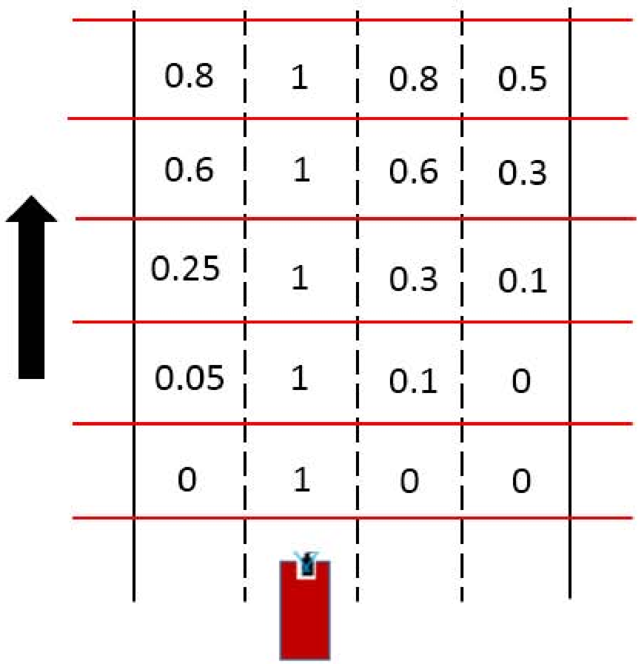


Geometric model of viewpoint 1

Geometric model of viewpoint 4

Probabilistic Viewpoint Map

- Example on Straight Road



world view

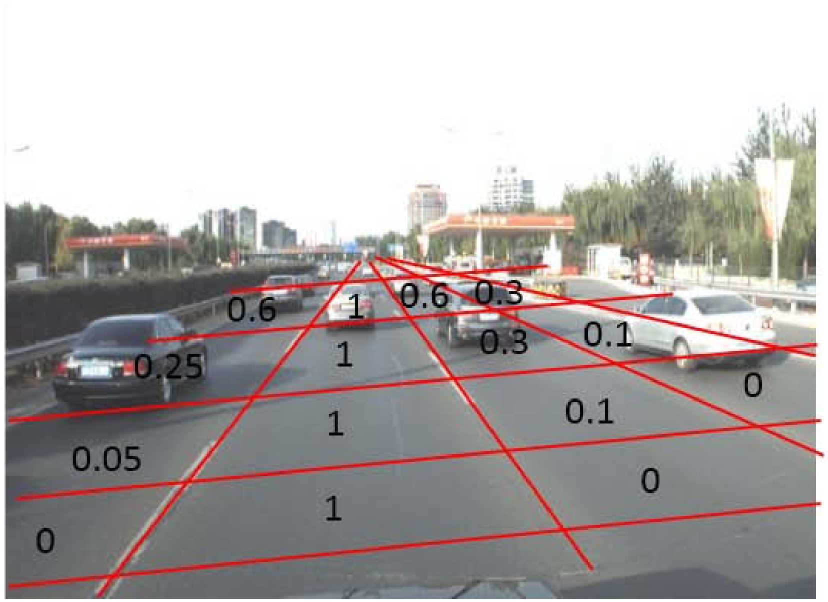
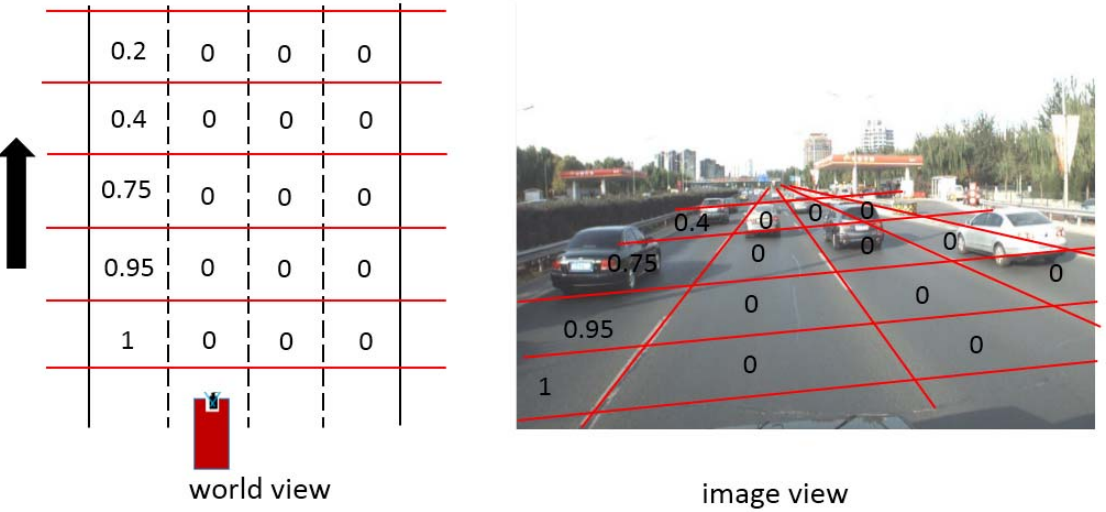


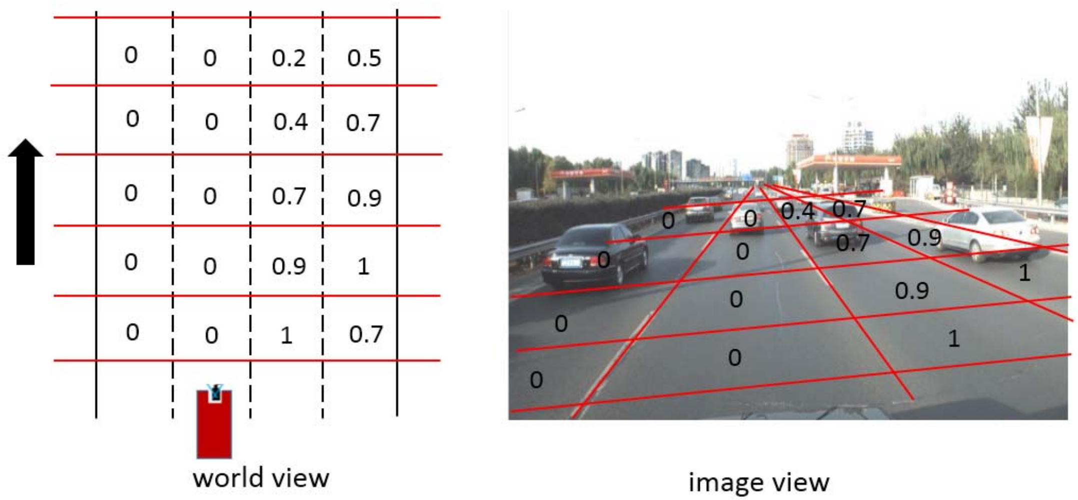
image view

RSPVM for viewpoint 1 on straight road

RSPVM



RSPVM for viewpoint 2 on straight road



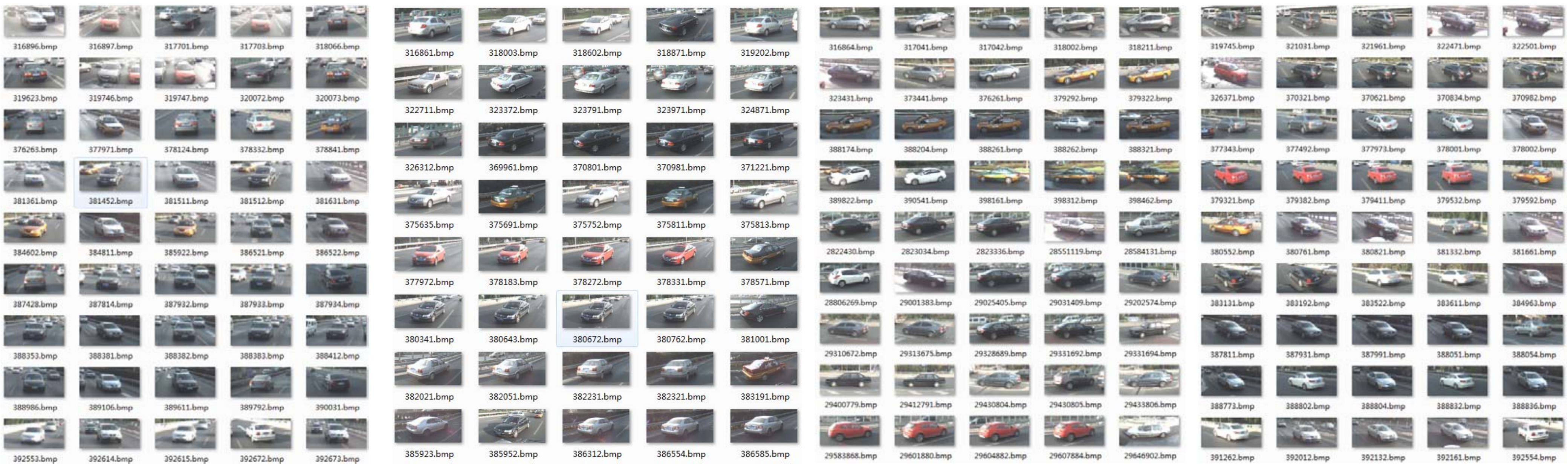
RSPVM for viewpoint 4 on straight road

Experiments

- Training and Test Data
  - **Reference:** C. Wang, H. Zhao, F. Davoine, H. Zha, A System of Automated Training Sample Generation for Visual-Based Car Detection, IROS2012.
  - Video images from a Ladybug3 camera

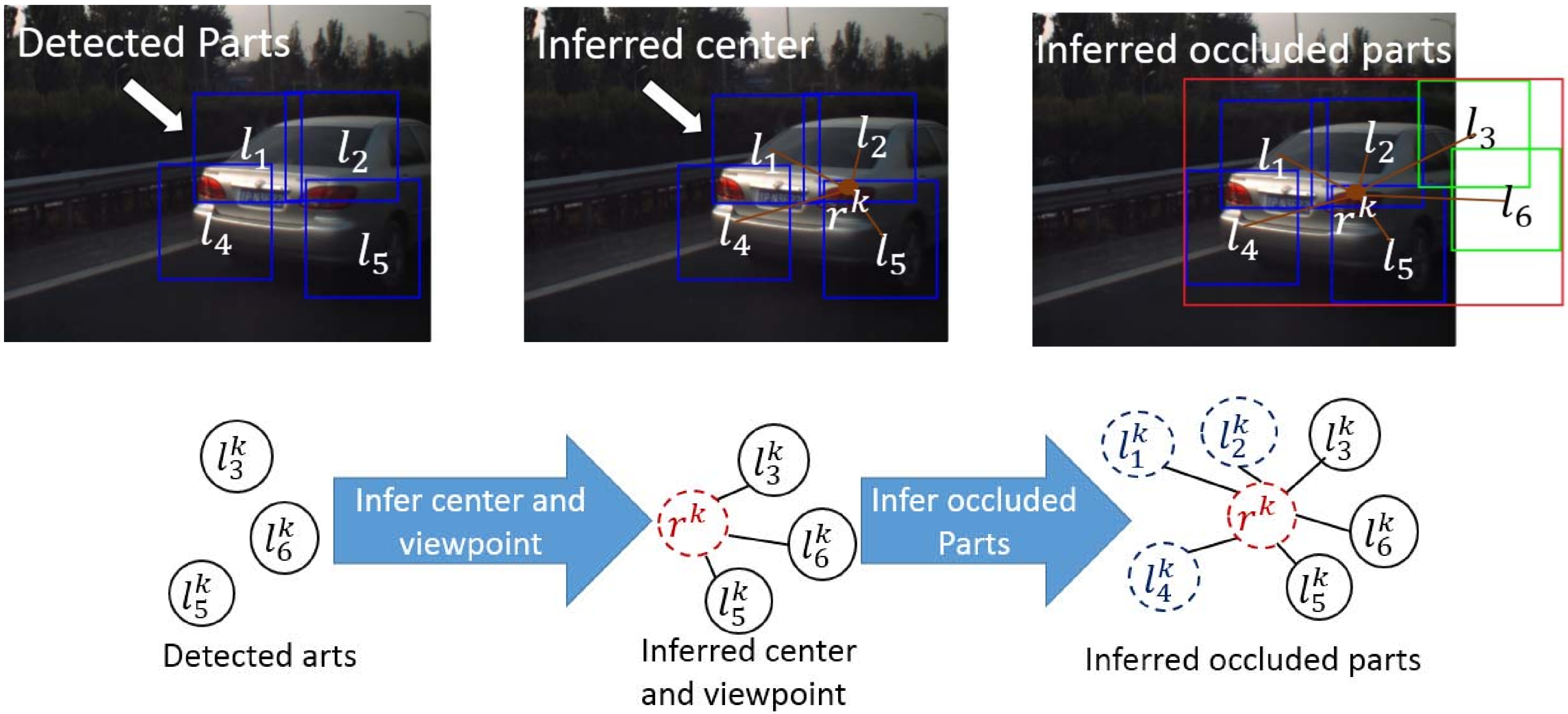
Experiments

- Training samples
    - Labeled with viewpoint class



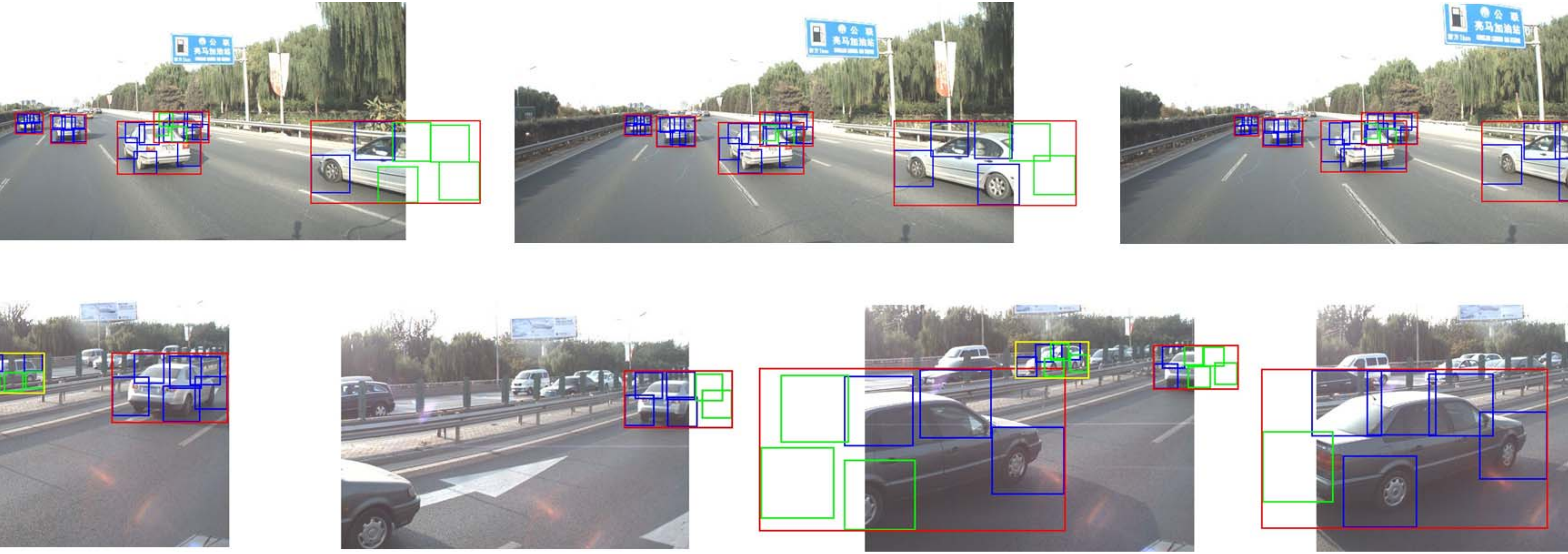
Viewpoint 1 Viewpoint 2 Viewpoint 3 Viewpoint 4

Results



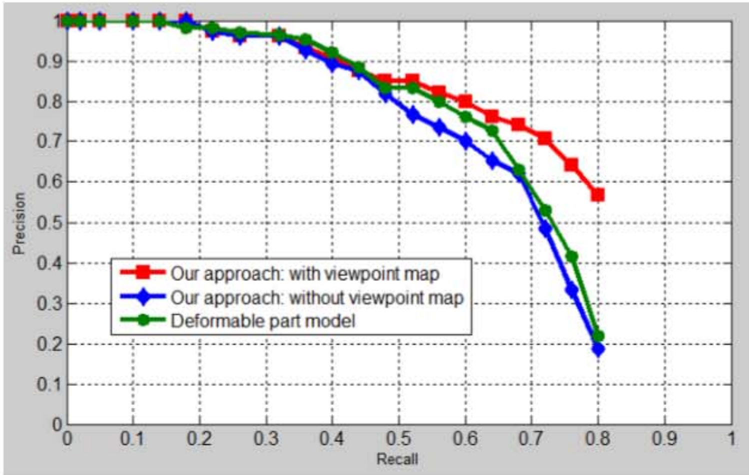
Detected Parts
Inferred center
Inferred occluded parts
l_1 l_2 l_3 l_4 l_5 l_6 r^k
l_3^k l_6^k l_5^k
Infer center and viewpoint
r^k l_3^k l_6^k l_5^k
Infer occluded Parts
l_1^k l_2^k l_3^k r^k l_6^k l_4^k l_5^k
Detected arts
Inferred center and viewpoint
Inferred occluded parts

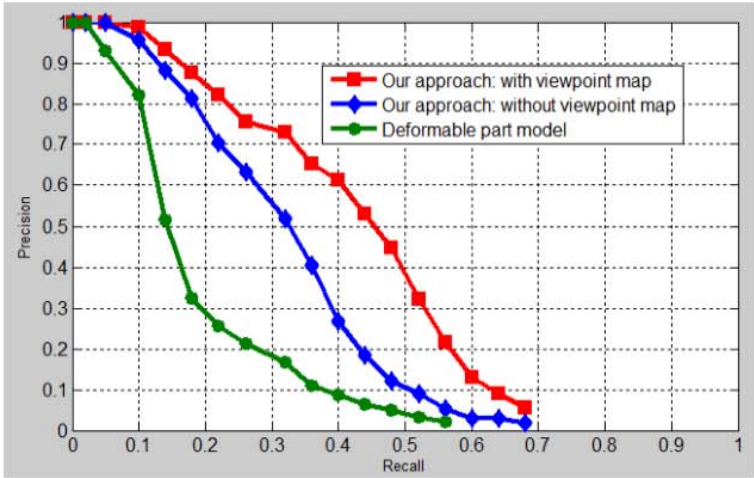
Results



Results

- Numerical experimental results analysis
  - Detection results in precision/recall curve compare with DPM
  - Viewpoint estimation accuracy (with/without RSPVM).





Viewpoint 1	0.88/0.76	0.04/0.10	0.0/0.0	0.08/0.14
Viewpoint 2	0.15/0.20	0.75/0.66	0.10/0.12	0.0/0.02
Viewpoint 3	0.0/0.0	0.07/0.12	0.83/0.74	0.10/0.14
Viewpoint 4	0.14/0.22	0.0/0.0	0.08/0.13	0.78/0.65
	V. 1	V. 2	V. 3	V. 4

(a) Detection results on full visible vehicles

Viewpoint 1	0.80/0.60	0.05/0.13	0.0/0.0	0.15/0.27
Viewpoint 2	0.21/0.30	0.68/0.55	0.11/0.13	0.0/0.02
Viewpoint 3	0.0/0.05	0.14/0.16	0.69/0.60	0.17/0.19
Viewpoint 4	0.22/0.22	0.0/0.04	0.13/0.16	0.65/0.58
	V. 1	V. 2	V. 3	V. 4

(b) Detection results on occluded vehicles

Conclusions

- This research proposed a framework for on-road vehicle detection with its focus on vehicle pose inference based on detected part instances by addressing both partial observation and varying viewpoints in one probabilistic framework.
- Geometric models describing the configuration of vehicle parts as well as their spatial relations are learned for each dominant viewpoint.
- Viewpoint maps are generated on each typical road structure for probabilistic prediction of vehicle at each location.
- Experiments have been conducted using Beijing ringroad data, results demonstrated efficiency of the proposed work on on-road vehicle detection, especially for the partially observed vehicles on varying viewpoints.

Wen Yao, Yubin Lin, Chao Wang, Huijing Zhao, Hongbin Zha

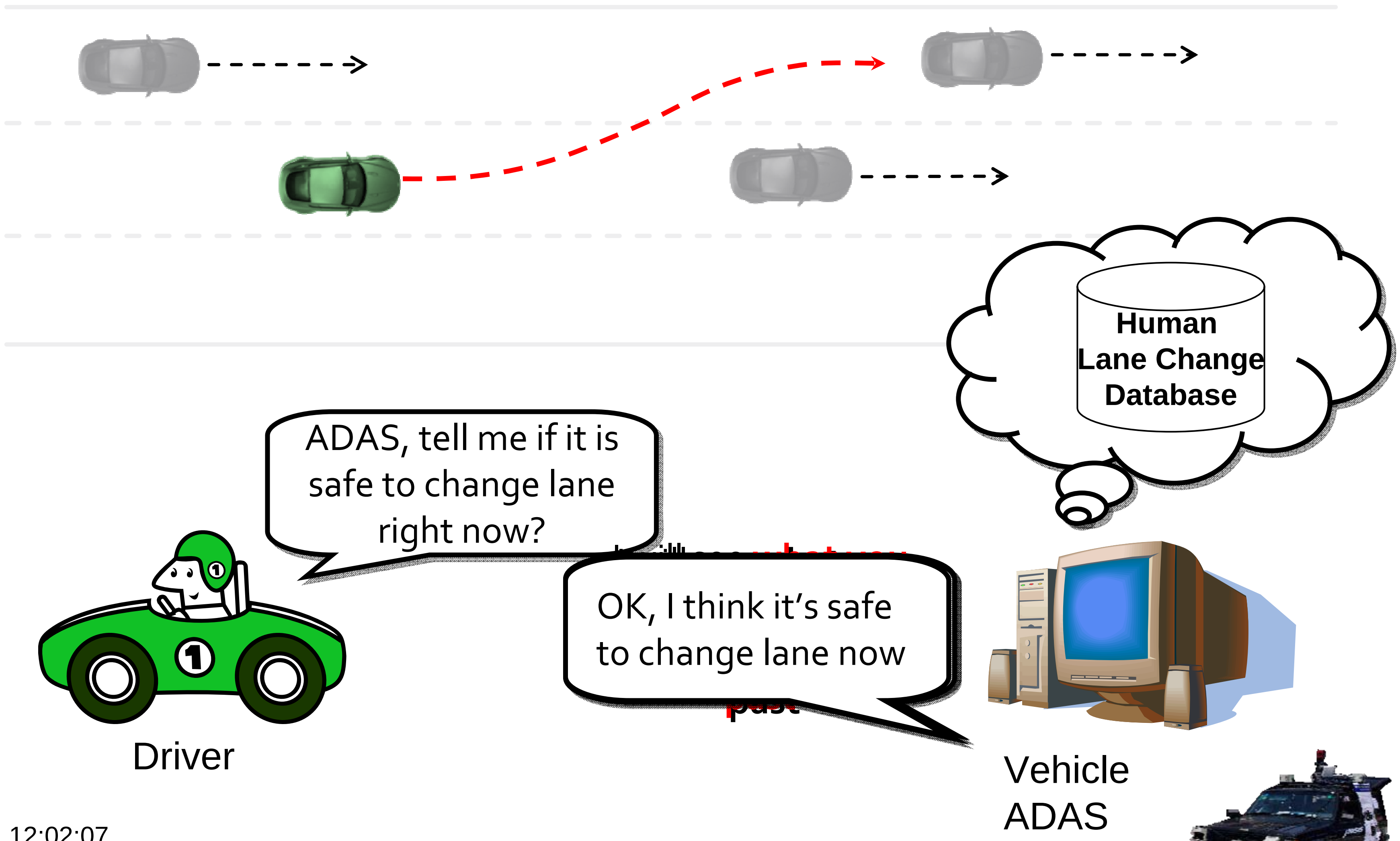
Learning from ones' history lane change driving data

Abstract

- Motivations:
  - Recent Advanced Driver Assistant System (ADAS) research tries to improve system intelligence to produce more naturalistic driving assistance by learning from human driving behaviors. This requires large amount of labeled real driving data as a prerequisite.
  - Extract data segments for specific driving behavior (e.g. lane change behavior) modeling from large car data sequence is time consuming for manual work and needs efficient and automatic extraction method.
  - Large amount of lane change behavior data can be recorded to form a database for further data analysis/ behavior modeling/ trajectory prediction.
- Contribution:
  1) An on-roar driving data acquisition system is set up.
  2) An efficient automatic lane change data segments extraction algorithm
  3) Real lane change data are recorded in a database and used for further behavior modeling related research

12:02:07

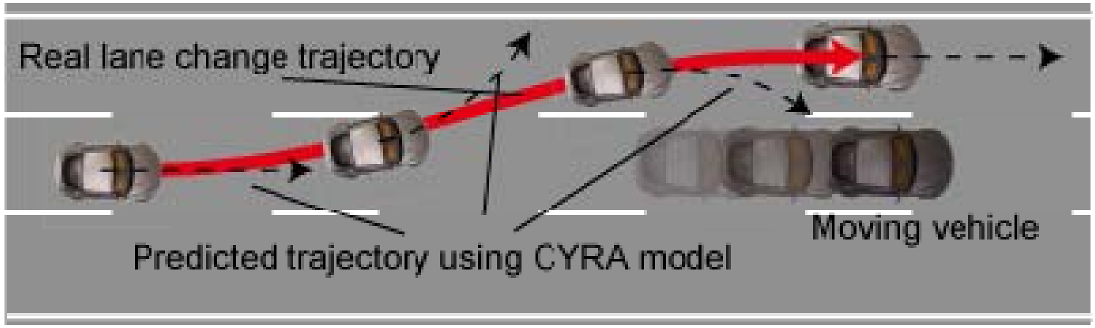
Introduction



12:02:07

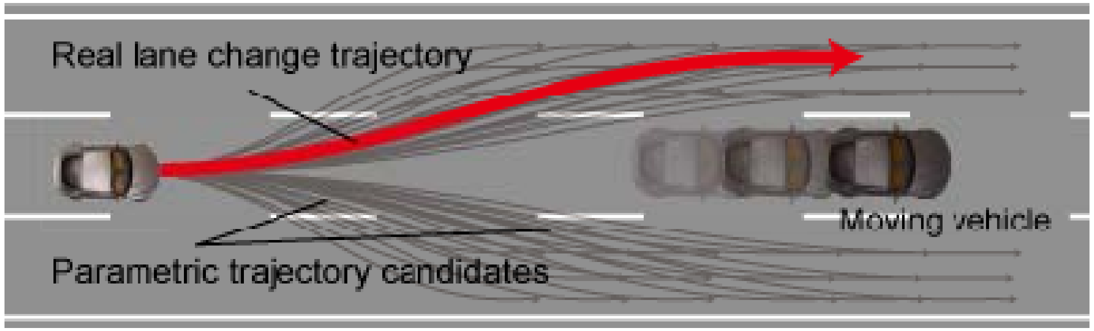
Introduction

- Why do we need to model human lane change behavior from real data



Lane change prediction using CYRA model

Require behavior level model



Trajectory selection from parametric data

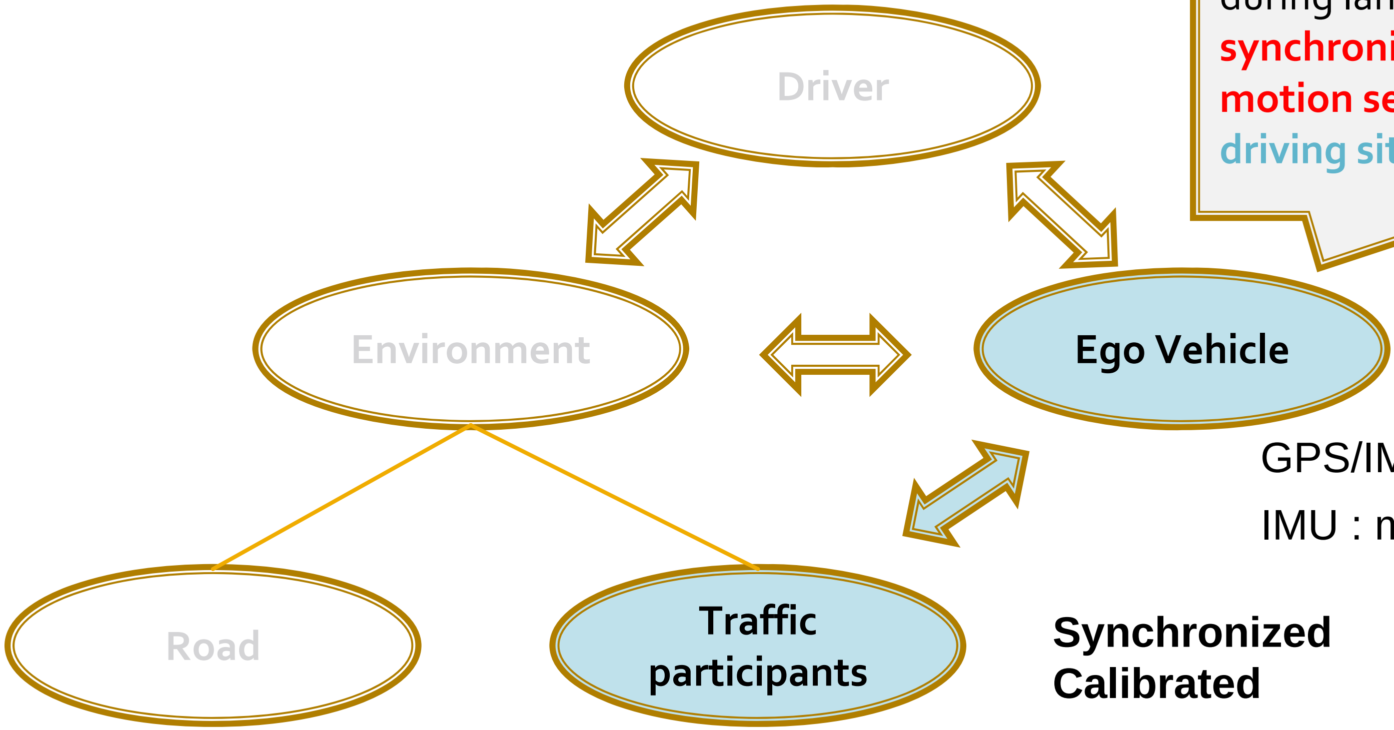
Non-naturalistic behavior model is not sufficient

Introduction

- In order to build naturalistic driving behavior models, we need:
  - Large amount of real driving data
    - → **Data collection platform**
  - Data segment extraction for target behavior from raw data
    - → **Driving behavior segment extraction**
  - A method to teach the ADAS system to give naturalistic driving assistance by learning from real driving behavior data
    - → **Application based on human driving behavior learning** (lane change trajectory prediction in this work)

Introduction

In order to study the interaction of ego and surrounding vehicles during lane change maneuvers, **synchronized acquisition** of the **motion sequences** in **natural driving situations** is required.



GPS/IMU : trajectory

IMU : motion of steering handle

Synchronized
Calibrated

LiDARs : motion trajectory of surrounding cars

Camera : lane marker

12:02:07

Research flow

An instrumented vehicle for on-road data acquisition by human drivers

1. On-road Driving Data Acquisition

GPS/IMU Vehicle Parameters Lidar Video

2. Lane Change Data Extraction

Ego vehicle Traffic participants Road Driver

Offline data processing

Synchronized trajectories of ego and environmental vehicles during lane change maneuver

Human Lane Change Database

A lane change intention is detected

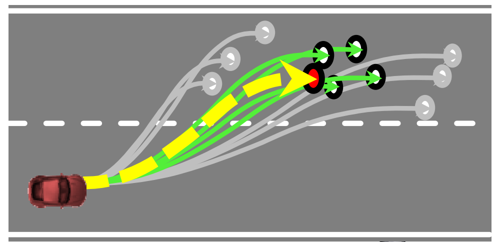
Situation-based Trajectory Searching

Find a set of lane change data at similar initial situation

Lane Change Trajectory Prediction

Parametric trajectory generation on end state prediction

Online risk assessment



12:02:07

Research flow

An instrumented vehicle for on-road data acquisition by human drivers

1. On-road Driving Data Acquisition

GPS/IMU *Vehicle Parameters* *Lidar* *Video*

2. Lane Change Data Extraction

Ego vehicle *Traffic participants* *Road* *Driver*

Offline data processing

Synchronized trajectories of ego and environmental vehicles during lane change maneuver

Human Lane Change Database

A lane change intention is detected

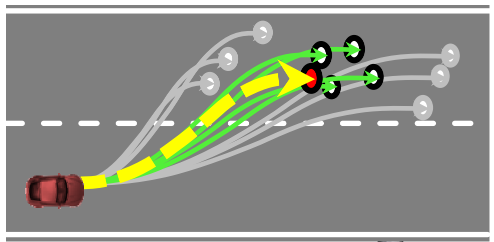
Situation-based Trajectory Searching

Find a set of lane change data at similar initial situation

Lane Change Trajectory Prediction

Parametric trajectory generation on end state prediction

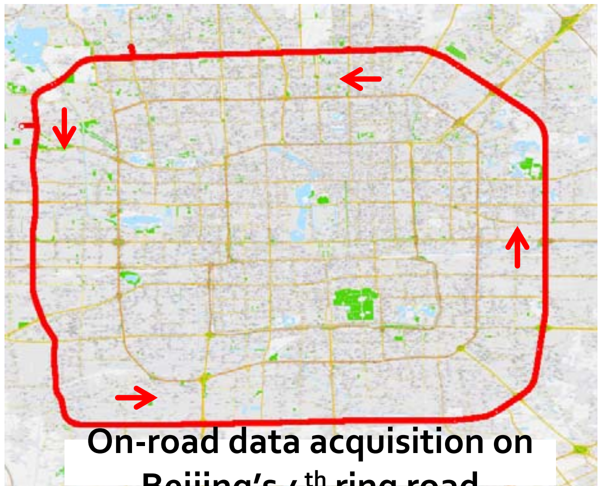
Online risk assessment



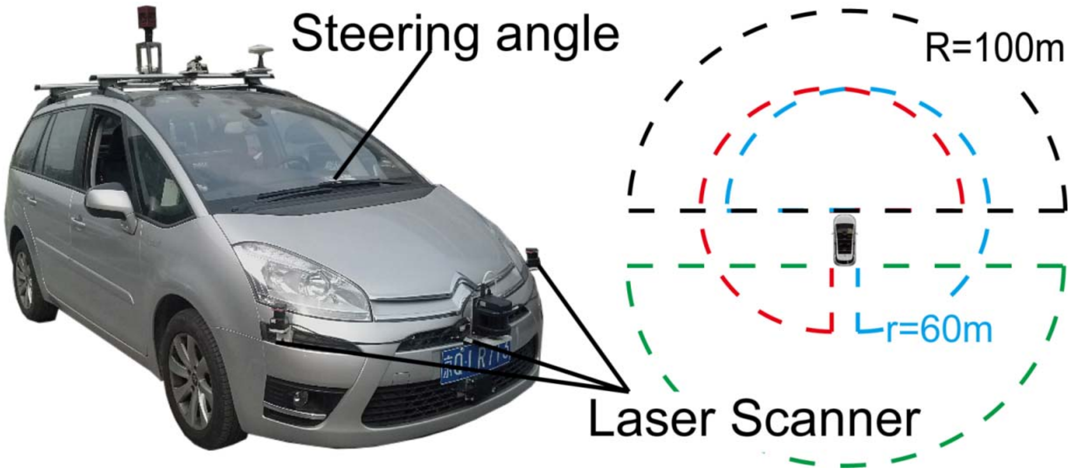
12:02:07

On road real driving data acquisition

- Experimental scenario:
  - Urban high speed road, mainly straight road
  - Fluent but heavy traffic so that there is great chance for lane change behavior.
- Data acquisition using an instrumented vehicle mainly recording:
  - Steering wheel angle
  - Wheel speed
  - Range data from LIDAR



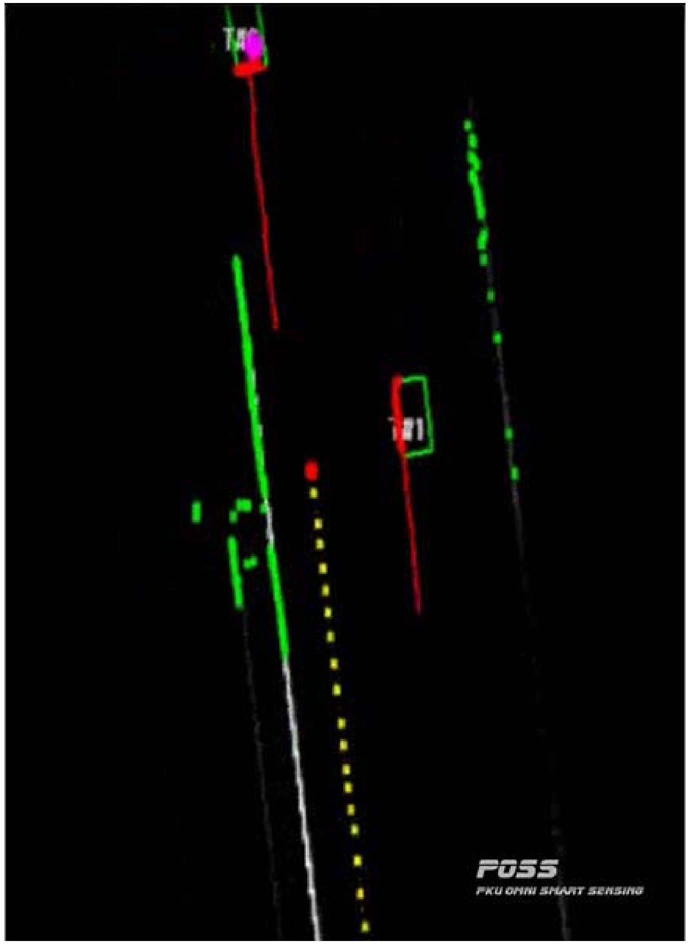
On-road data acquisition on Beijing's 4th ring road



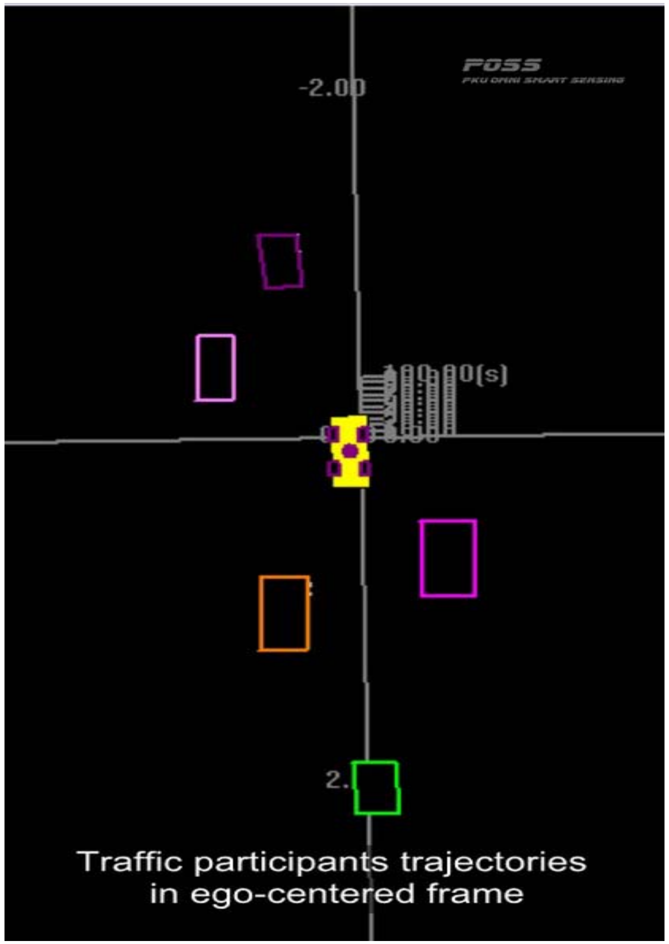
12:02:07

On road real driving data acquisition

- Synchronized trajectory collection of the ego and surrounding vehicles



On-road data visualization
from MODT module



An extracted lane change
behavior segment in ego-frame

12:02:07

Research flow

An instrumented vehicle for on-road data acquisition by human drivers

1. On-road Driving Data Acquisition

GPS/IMU *Vehicle Parameters* *Lidar* *Video*

2. Lane Change Data Extraction

Ego vehicle *Traffic participants* *Road* *Driver*

Offline data processing

Synchronized trajectories of ego and environmental vehicles during lane change maneuver

Human Lane Change Database

A lane change intention is detected

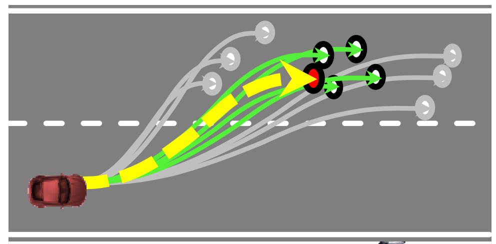
Situation-based Trajectory Searching

Find a set of lane change data at similar initial situation

Lane Change Trajectory Prediction

Parametric trajectory generation on end state prediction

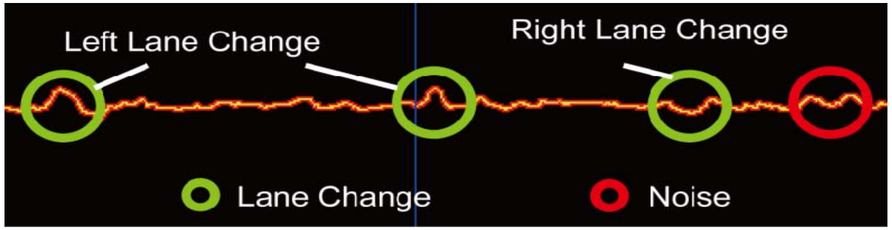
Online risk assessment



12:02:07

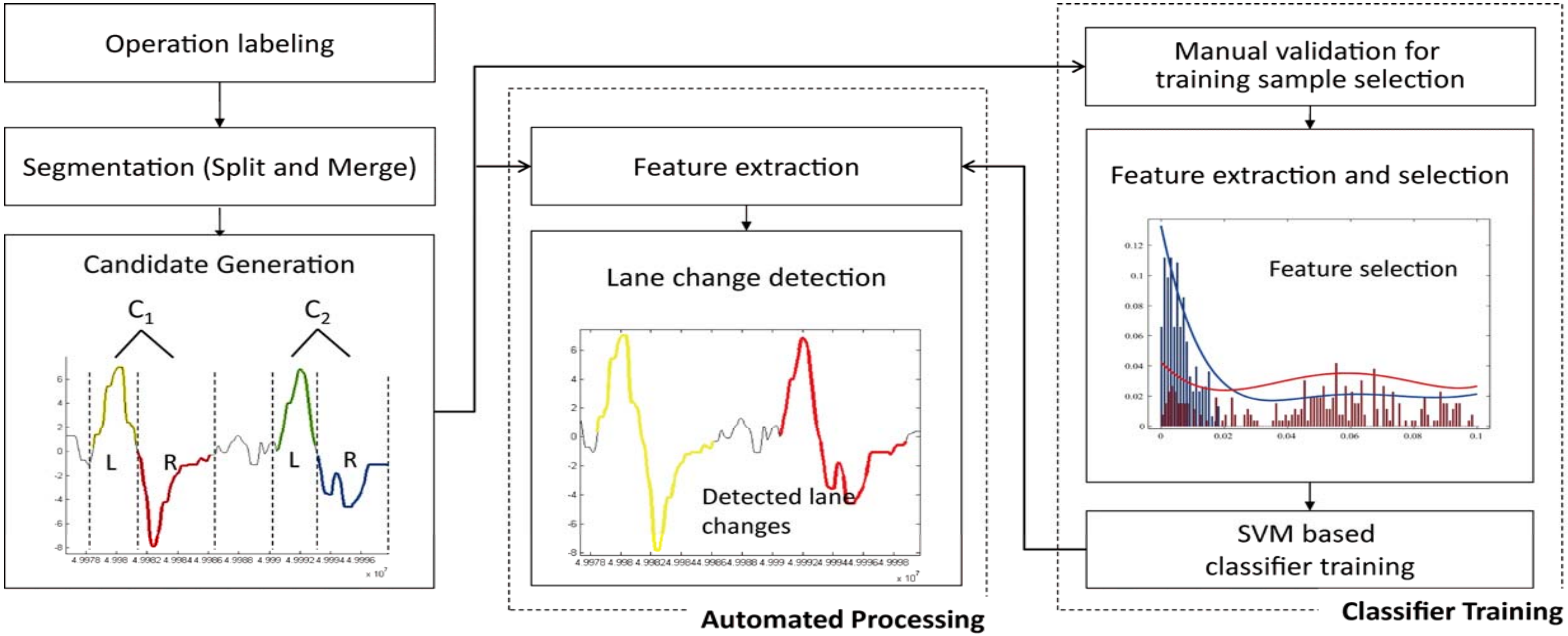
Automatic lane change data extraction algorithm

- Real driving data is noisy and requires efficient lane change behavior data extraction algorithm to generate large amount of samples for behavior modeling



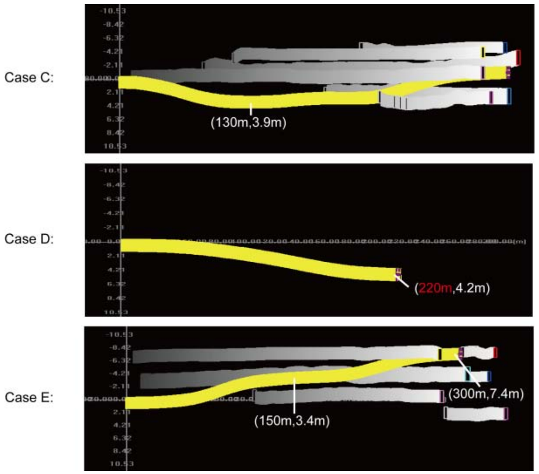
Steering wheel data

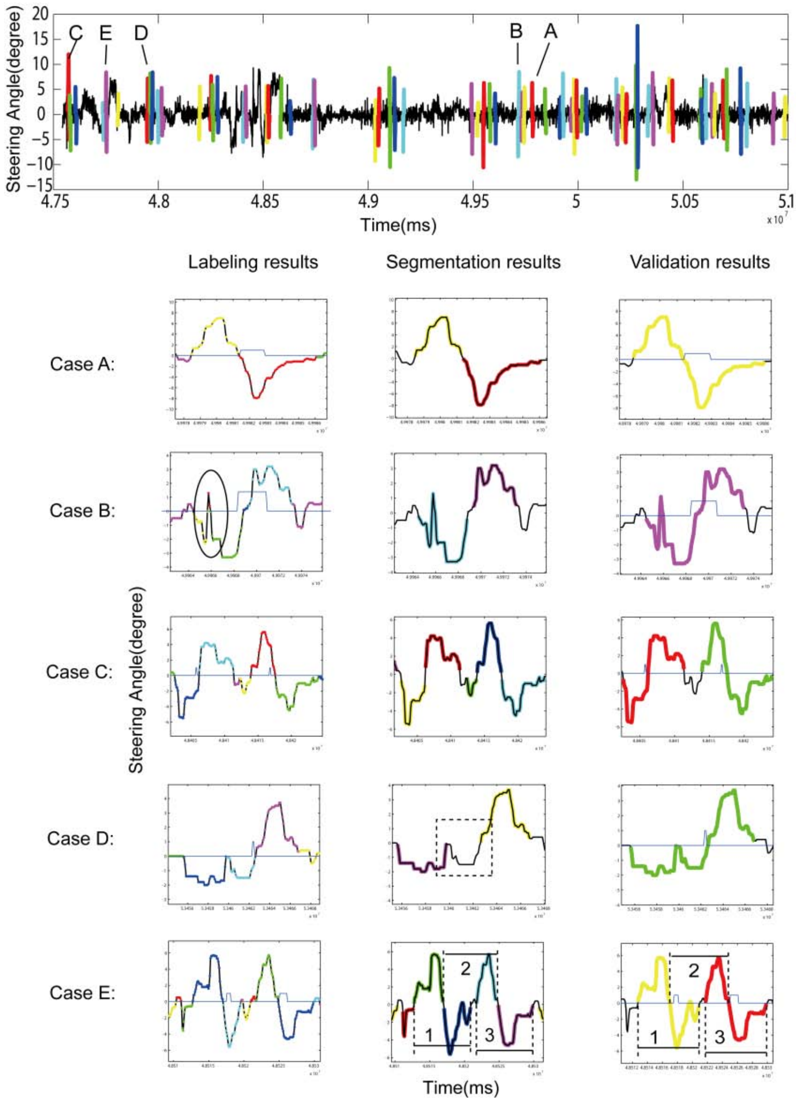
- Work flow of automatic lane change behavior data extraction



Automatic lane change data extraction algorithm

- Extraction steps and results of each step
- We can handle multiple cases which might be confused in real driving situation

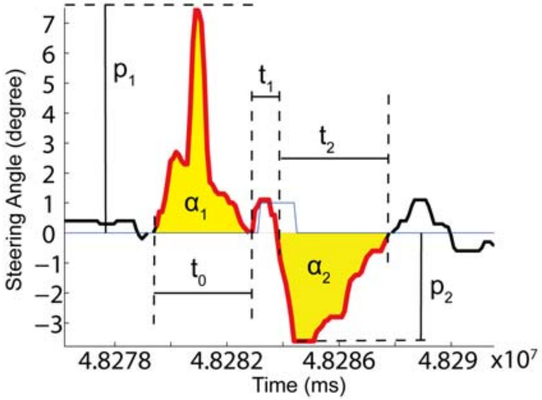




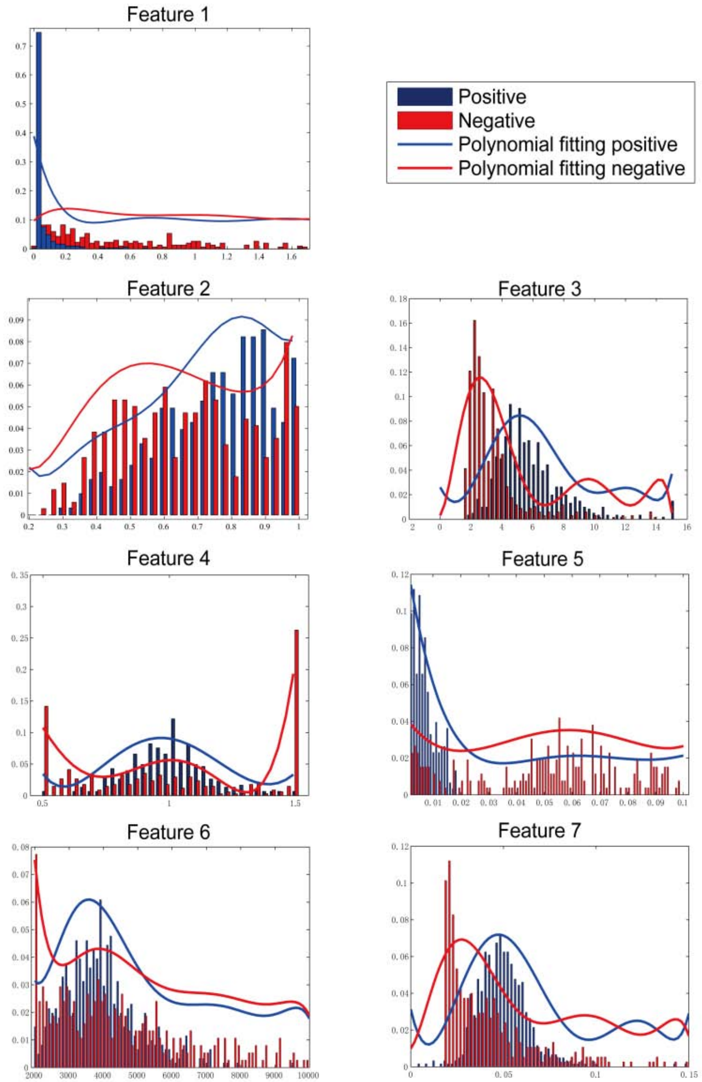
12:02:07

Automatic lane change data extraction algorithm

- Feature definition and selection

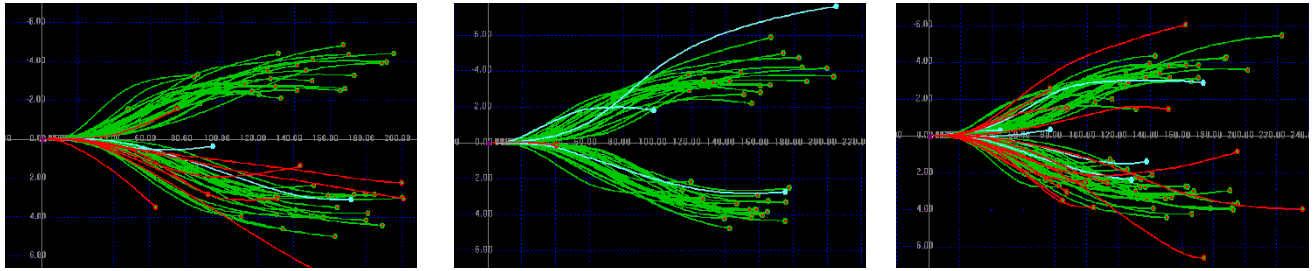


Feature	Definition
$\frac{2t_1}{(t_0+t_2)}$	Gap time / Ratio time
$\frac{\min(t_0,t_2)}{\max(t_0,t_2)}$	Min maneuver time / Max maneuver time
p_1, p_2	Peak steering value for each turning maneuver
$\frac{\alpha_1}{\alpha_2}$	Yaw difference of first turning / Yaw difference of second turning
$\lvert\alpha_1+\alpha_2\rvert$	Yaw difference produced by a candidate maneuver pair
t_1, t_2	Time of each turning maneuver
α_1, α_2	Yaw difference produced by each turning maneuver



Automatic lane change data extraction algorithm

- Extract ego-vehicles' lane change trajectories



Day	Round	Label	T	F	TP Rate	TN Rate	Accuracy
D1	R1	P	48/50	2/50	96%	97.8%	97.6%
		N	356/364	8/364			
	R2	P	38/42	4V42	90.5%	97.6%	96.8%
		N	361/370	9/370			
D2	R1	P	42/45	3/45	93.3%	99.7%	99%
		N	372/373	1/373			
	R2	P	59/64	5/64	92.2%	97.7%	97.1%
		N	503/515	12/515			
D3	R1	P	47/50	3/50	94%	98.8%	98.2%
		N	397/402	5/402			
	R2	P	53/57	4/57	93%	97.4%	97%
		N	665/683	18/683			

12:02:07

Research flow

An instrumented vehicle for on-road data acquisition by human drivers

1. On-road Driving Data Acquisition

GPS/IMU *Vehicle Parameters* *Lidar* *Video*

2. Lane Change Data Extraction

Ego vehicle *Traffic participants* *Road* *Driver*

Offline data processing

Synchronized trajectories of ego and environmental vehicles during lane change maneuver

Human Lane Change Database

A lane change intention is detected

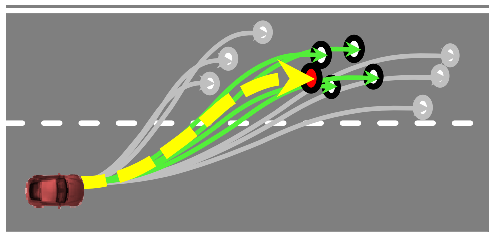
Situation-based Trajectory Searching

Find a set of lane change data at similar initial situation

Lane Change Trajectory Prediction

Parametric trajectory generation on end state prediction

Online risk assessment



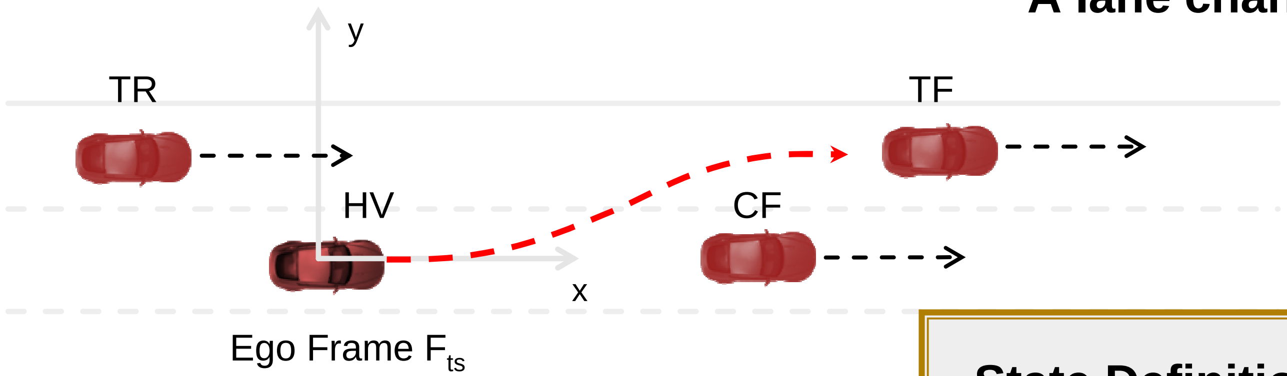
12:02:07

Lane change trajectory prediction

- Lane departure warning application
  - Predict lane change behavior according to lateral offset, yaw angle, etc.
  - Ego-vehicle states → lane changing or not
- Driver intention prediction
  - Predict lane change behavior according to driver operation / driver gaze direction
  - Driver-states → driver is going to change lane or not
- Our trajectory prediction for risk assessment
  - Judging if it is the right time to change lane according to the lane change habit of this driver
  - Ego-vehicle states + surrounding vehicle states + driver history lane change database → a trajectory which the driver is most probably to execute if he changes lane right now

State space definition for lane change prediction

A lane change maneuver (t_s,t_e)

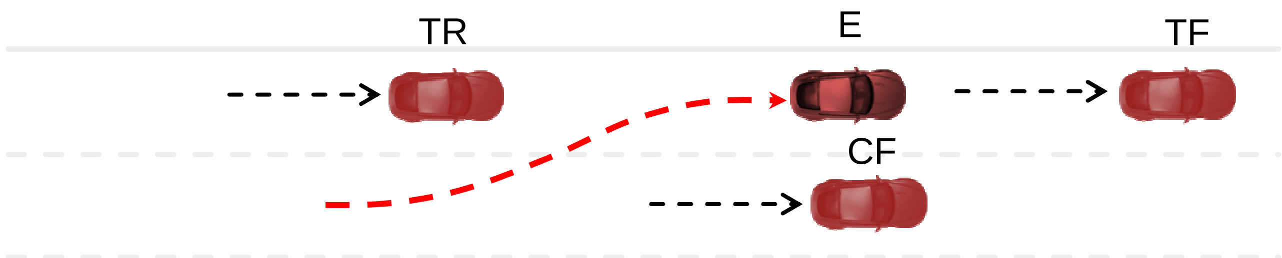


Initial state S at $\mathbf{t_s}$

State Definition

S, S* = (s_{HV}, s_{CF}, s_{TF}, s_{TR})

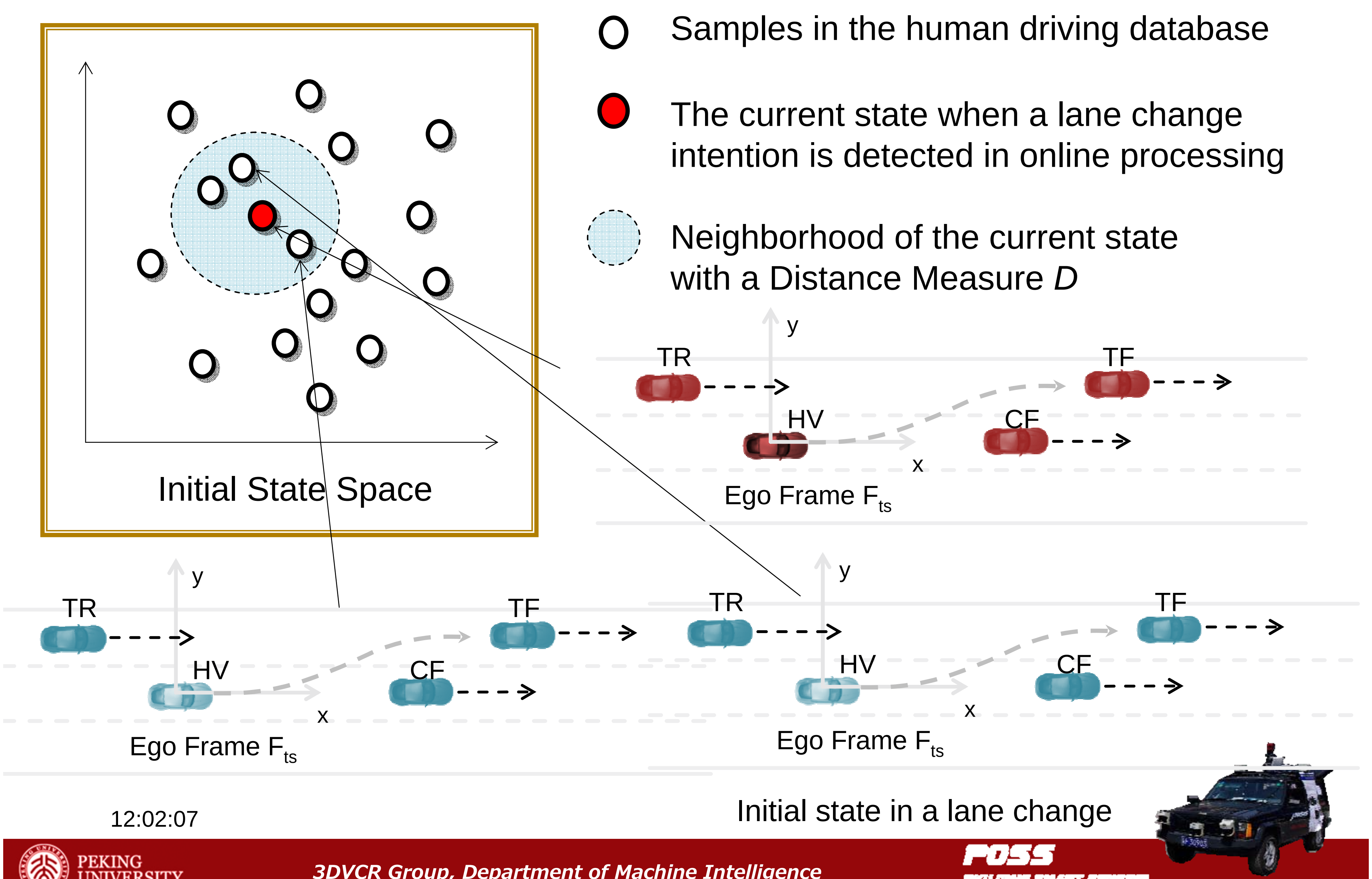
$s_{HV,CF,TF,TR}$ = (Pos, $Spe.$, $Acc.$) at $F_{ts,te}$



End state S* at $\mathbf{t_e}$

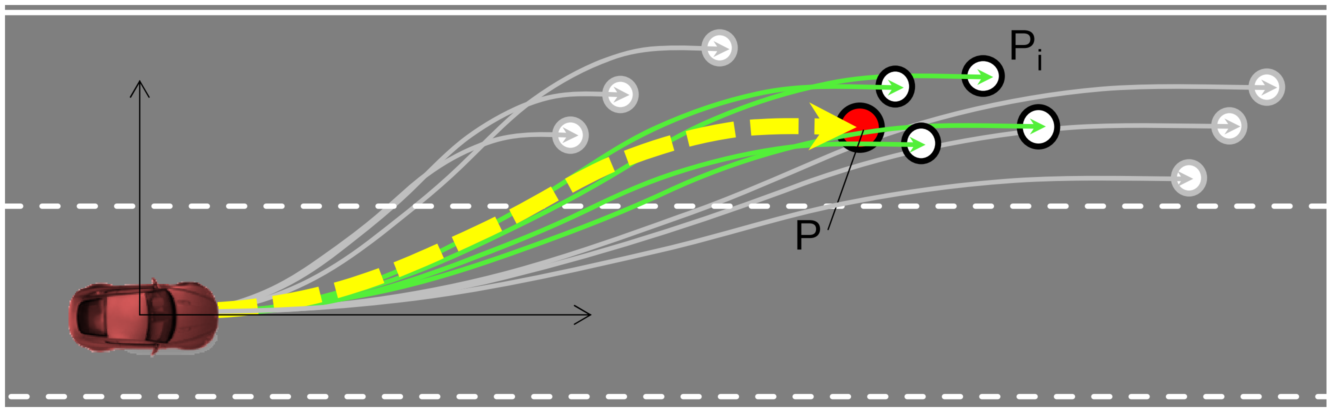
12:02:07

Lane change trajectory prediction

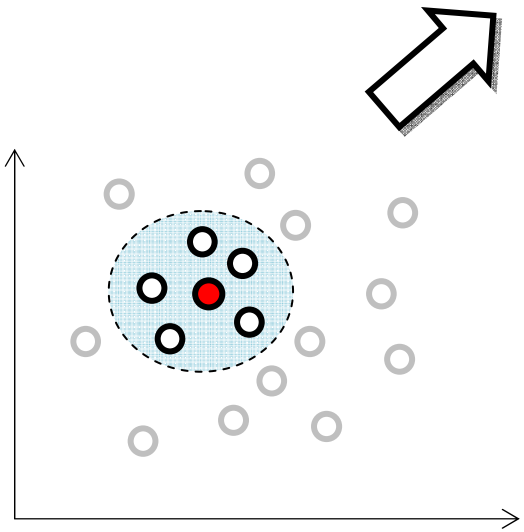


12:02:07

Lane change trajectory prediction



End State Space



Initial State Space

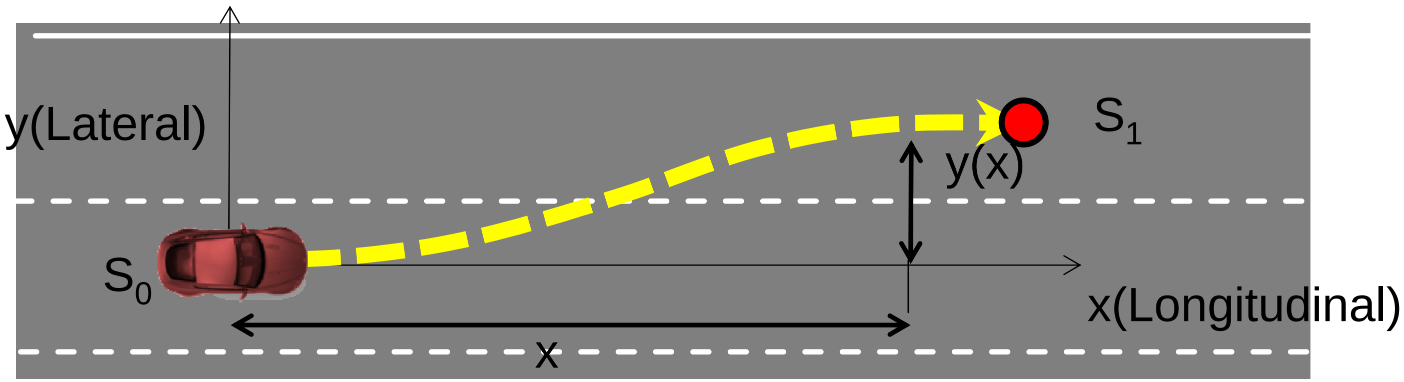
12:02:07

- End states of the neighborhood samples in the human driving database
- A predicted end state
- A predicted lane change trajectory

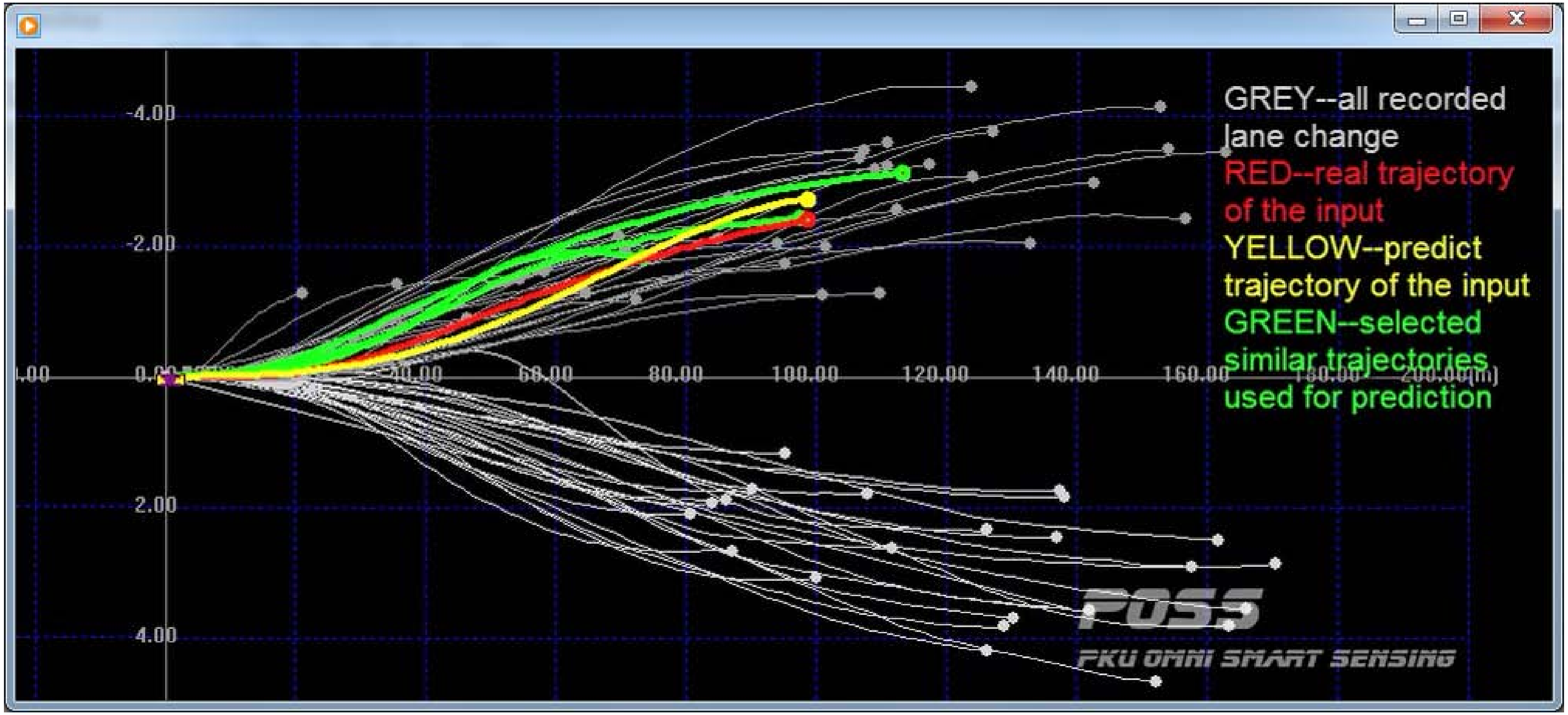
Lane change trajectory prediction

- Quintic polynomial model
  - Given the initial state the predicted end state $S_1 = (x(t_1), y(t_1), y(\dot{t}_1), y(\ddot{t}_1))$ of a lane change behavior, a quintic polynomial model is used to generate a smooth trajectory as the estimation:
  - $y(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5$

satisfying $y(x_1) = y_1; y(0) = y'(0) = y'(x_1) = y''(0) = y''(x_1) = 0$



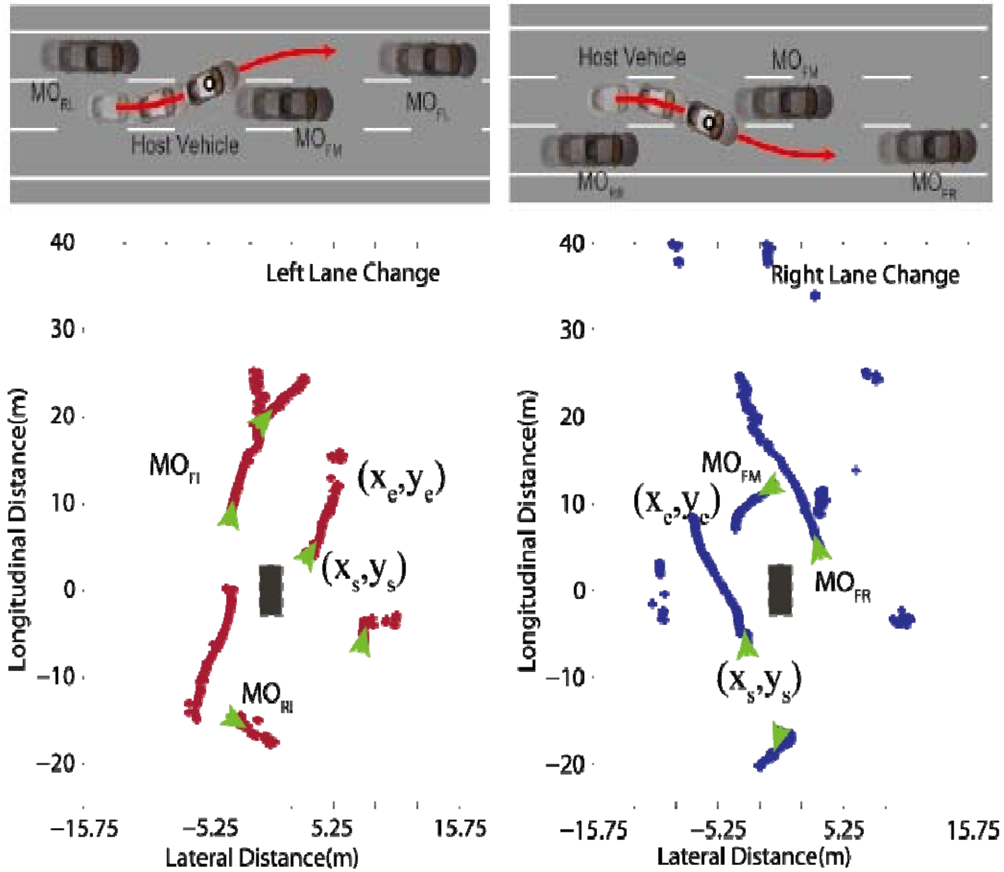
A result of lane change trajectory prediction



12:02:07

Lane change data analysis

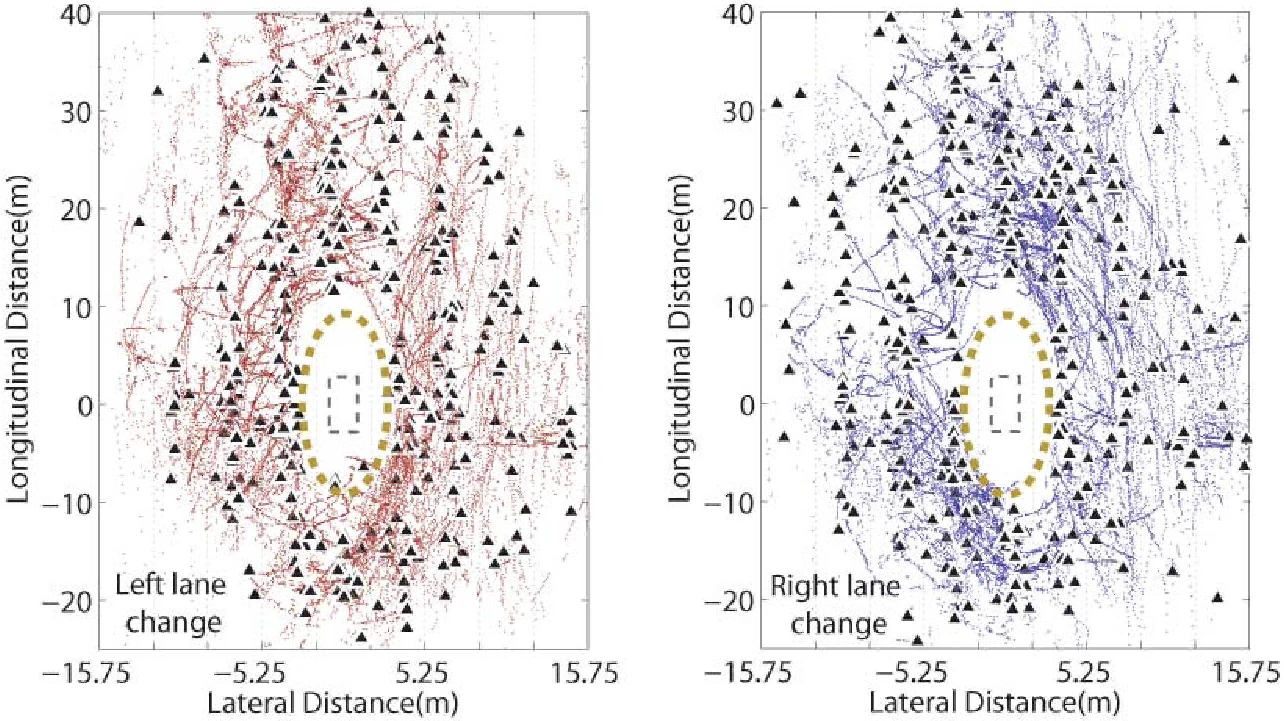
- Traffic participants' trajectories in ego-vehicle local frame



12:02:07

Lane change data analysis

- Traffic participants around ego-vehicle
  - Safety zone
    - With large amount of collected trajectories of adjacent traffic participants, we find an area which implies the driver's personal evaluation of safety during lane change



12:02:07

IEEE Intelligent Vehicle Symposium 2014

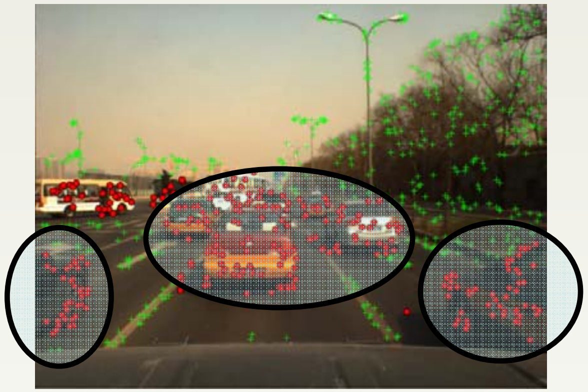
Monocular Visual Localization using Road Structural Features

Yufeng Yu [1], Huijing Zhao [1], Franck Davoine [2], Jinshi Cui [1], Hongbin Zha [1]

[1] Key Lab of Machine Perception, Peking University, Beijing, China
[2] CNRS, LIAMA Sino-French Laboratory, Beijing, China

Motivation

Point based localization



moving objects

Lane based localization

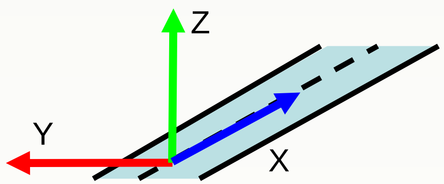
unclear lane

Verticle line based localization

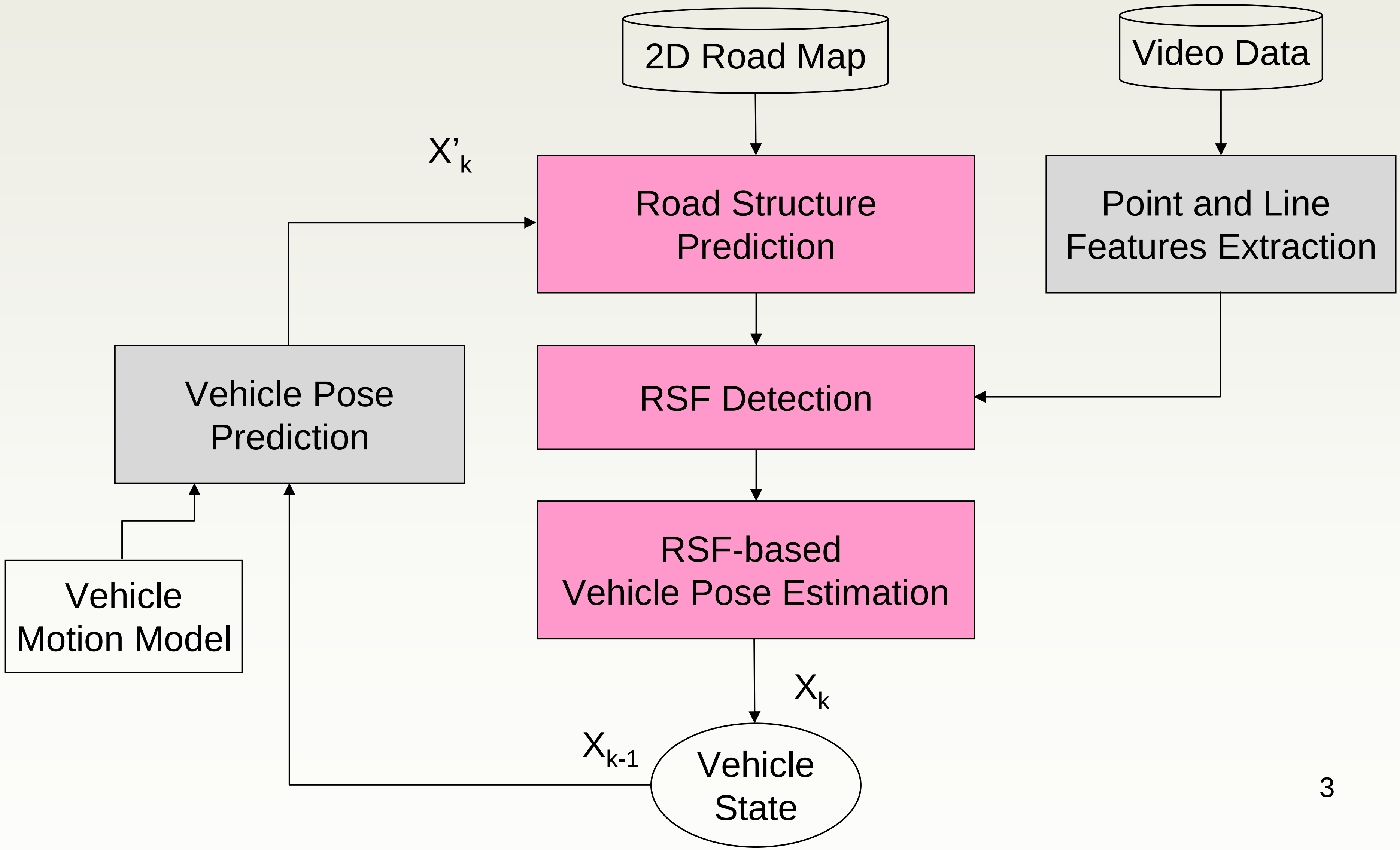
unclear verticle line

Combine all the features, define a Road Structural feature (RSF)

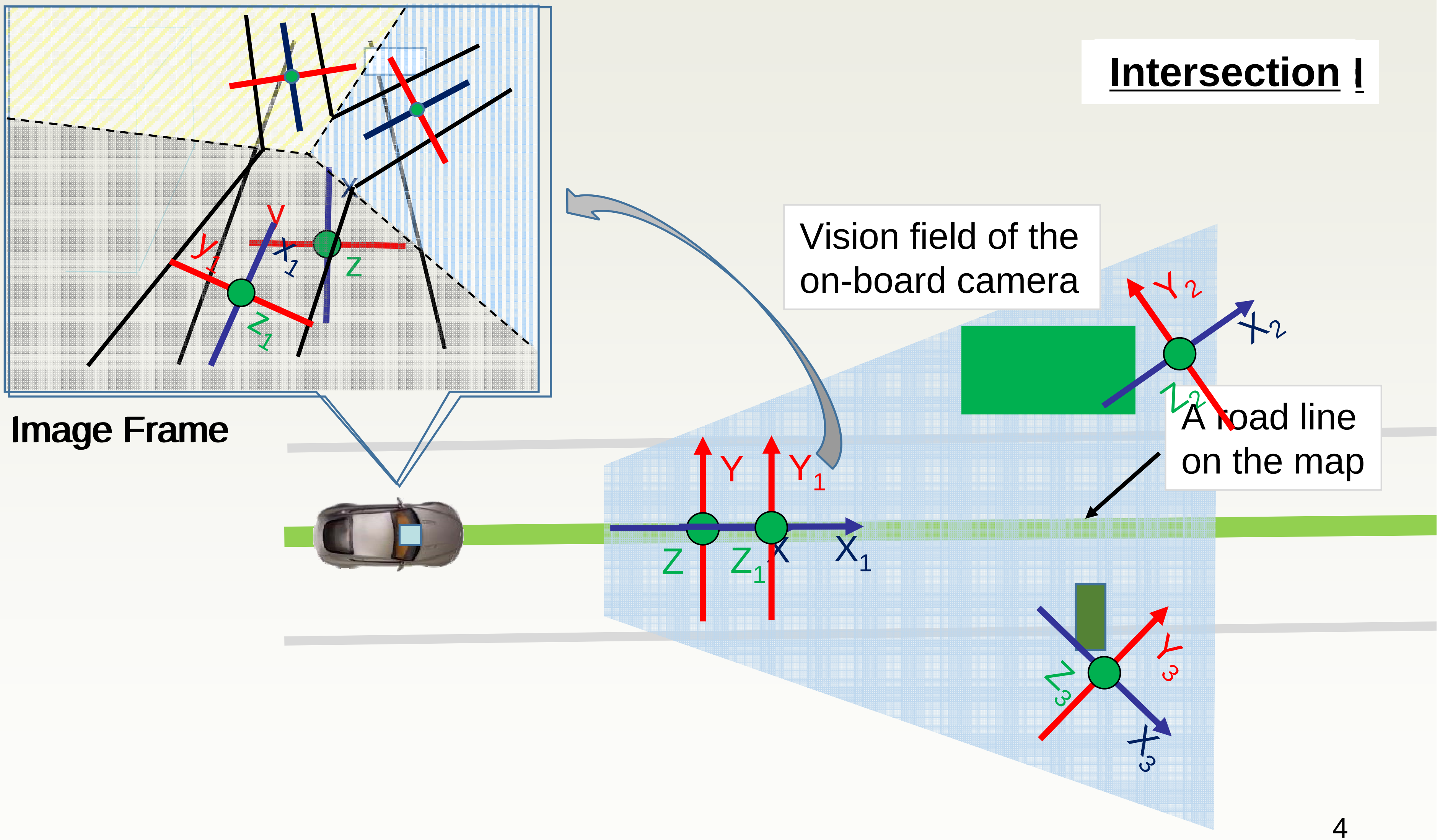
$$RSF = \{\mathbf{L}_x, \mathbf{L}_y, \mathbf{L}_z, \mathbf{P}\}$$



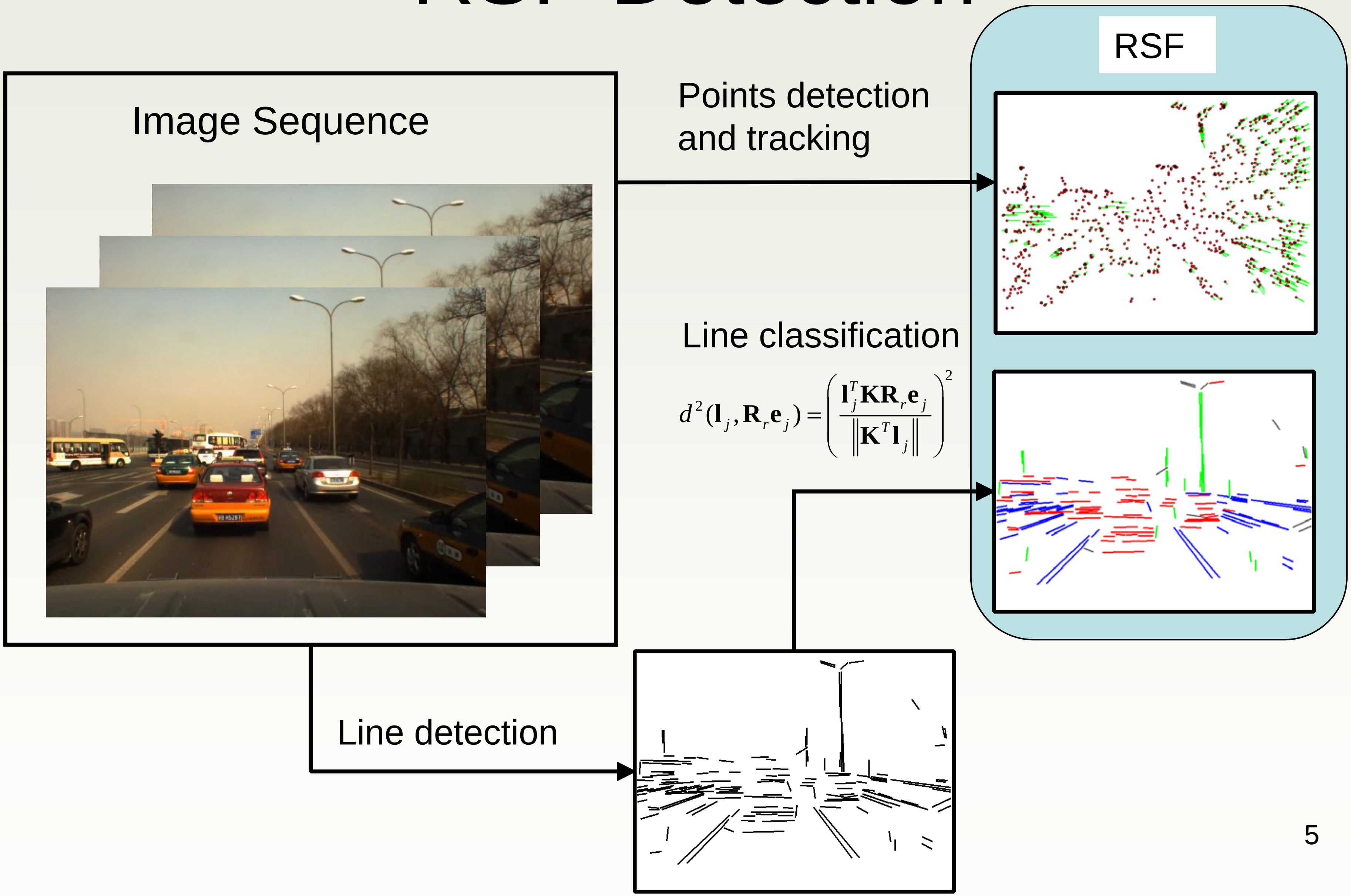
System Framework



Road Structure Prediction



RSF Detection



RSF-based Vehicle Pose Estimation

- 1) Sample RSF candidate with

$$C_{RSF} = \{\mathbf{l}_1, \mathbf{l}_2 \in \mathbf{L}_u, \mathbf{l}_3 \in \mathbf{L}_v, u \neq v, \mathbf{p}_1, \mathbf{p}_2 \in \mathbf{P}\}$$

- 2) Calculate x_t with given C_{RSF}

$$\left.\begin{array}{c} \mathbf{l}_1 \times \mathbf{l}_2 = \mathbf{v}_1 = \mathbf{KRd}_1 \\ \mathbf{R} = (\mathbf{I} - \mathbf{S})^{-1}(\mathbf{I} + \mathbf{S}) \\ \mathbf{S} = \begin{bmatrix} 0 & -c & -b \\ c & 0 & -a \\ b & a & 0 \end{bmatrix} \\ \mathbf{l}_3^T \mathbf{KRd}_3 = 0 \end{array}\right\} \Rightarrow \mathbf{R} \qquad \left.\begin{array}{c} \mathbf{R}_{rel} = \mathbf{RR}'^T \\ \mathbf{F} = \mathbf{K}^{-T} [\mathbf{t}_{rel}]_\times \mathbf{R}_{rel} \mathbf{K}^{-1} \\ p_1^T \mathbf{F} p_1' = 0 \\ p_2^T \mathbf{F} p_2' = 0 \end{array}\right\} \Rightarrow \mathbf{t}_{rel}$$

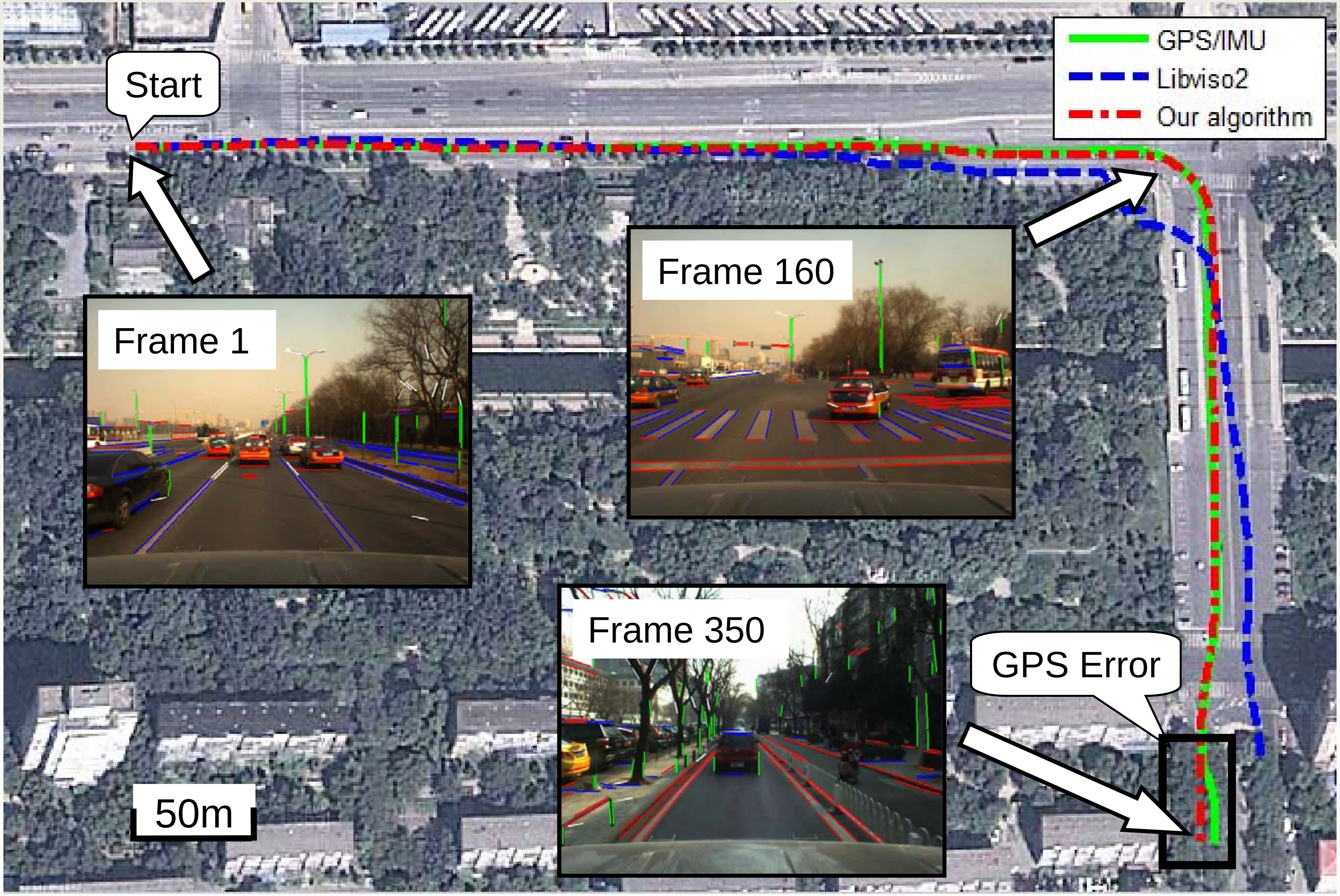
$$\mathbf{t}_k = \mathbf{t}_{rel} - \mathbf{R}_{rel}^T \mathbf{t}_{k-1}$$

$||t_{rel}||$ is calculated by the speed data

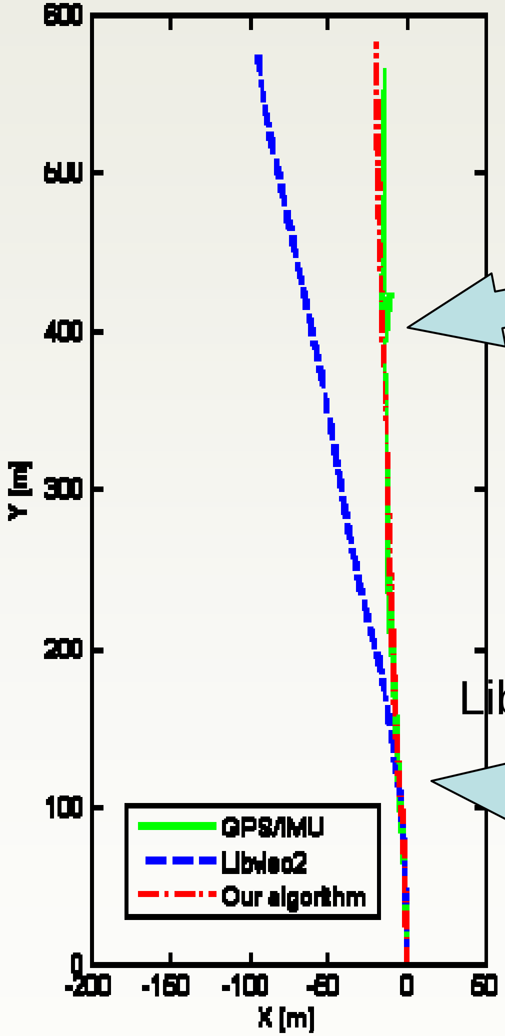
- 3) Evaluate x_t using an observation error measurement

$$E = E_l + \lambda E_p = \sum_j length^2(\mathbf{l}_j) \cdot \left(\frac{\mathbf{l}_j^T \mathbf{KR}\hat{\mathbf{d}}_j}{\left\| \mathbf{K}^T \mathbf{l}_j \right\|} \right)^2 + \lambda \sum_i \left(d^2(\mathbf{p}_i, \mathbf{Fp}_i') + d^2(\mathbf{p}_i', \mathbf{F}^T \mathbf{p}_i) \right)$$

Experimental Results (Normal Traffic)

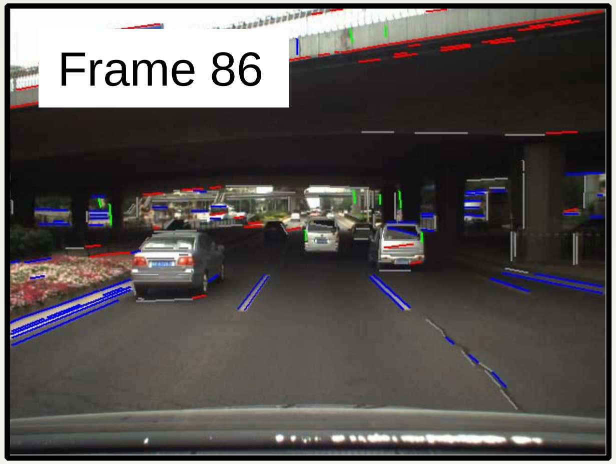


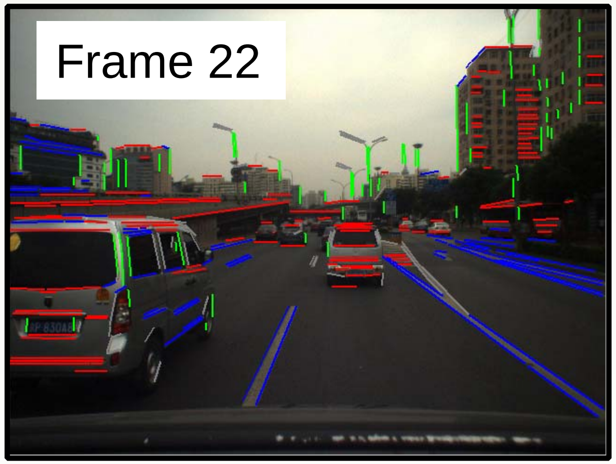
Experimental Results (Straight Road)



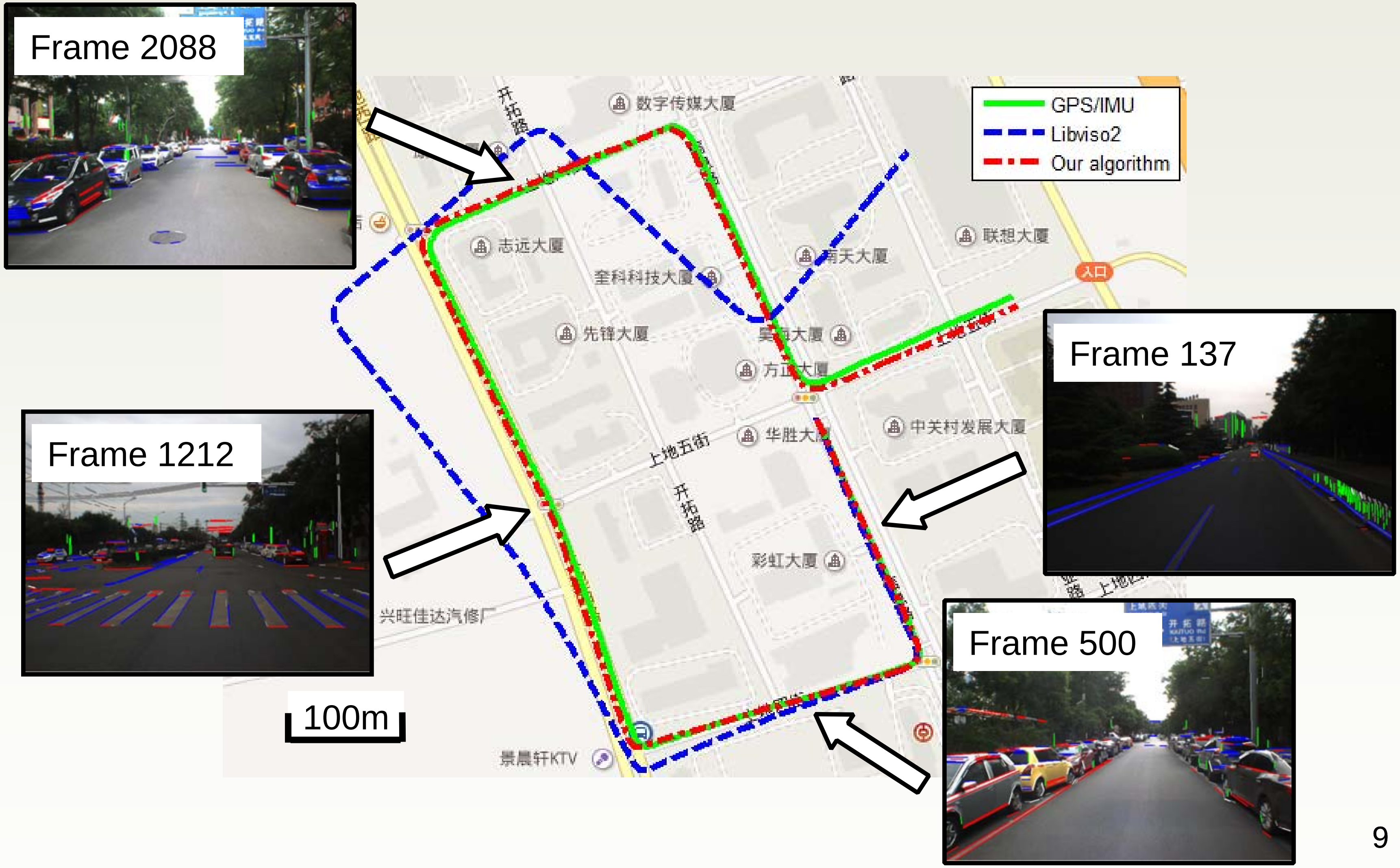
GPS Error

Libviso2 Error





Experimental Results (Complex Situation)



The 2010 IEEE International Conference on Robotics and Automation

Scene Understanding in a Large Dynamic Environment through a Laser-based Sensing

H. Zhao, **Yiming Liu**, X. Zhu, Y. Zhao, H. Zha

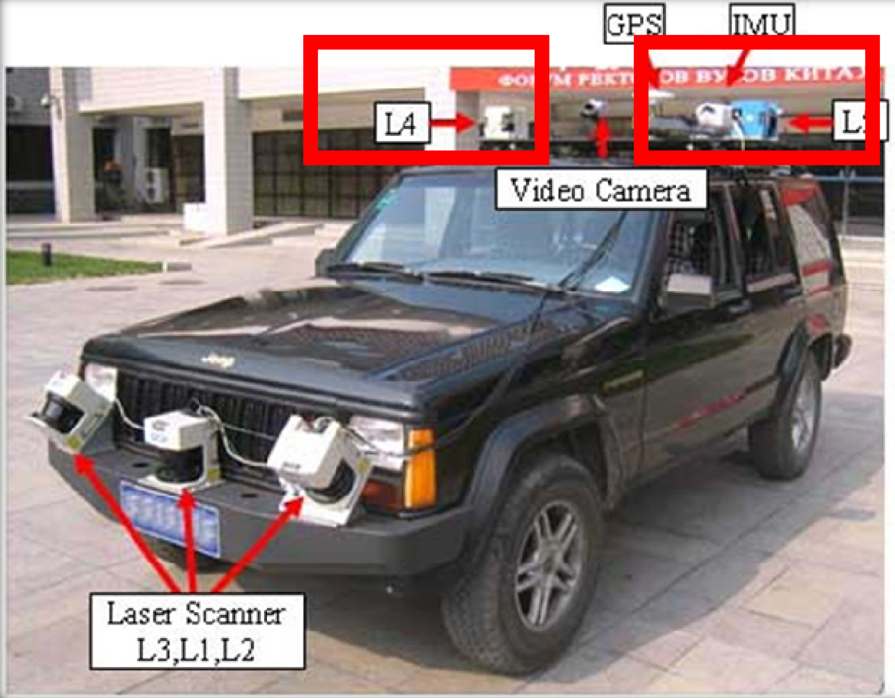
Peking University

Outline

- Introduction
- Problem formulation
- Framework
- Experimental results
- Summary
- Future work

Introduction – Data Acquisition

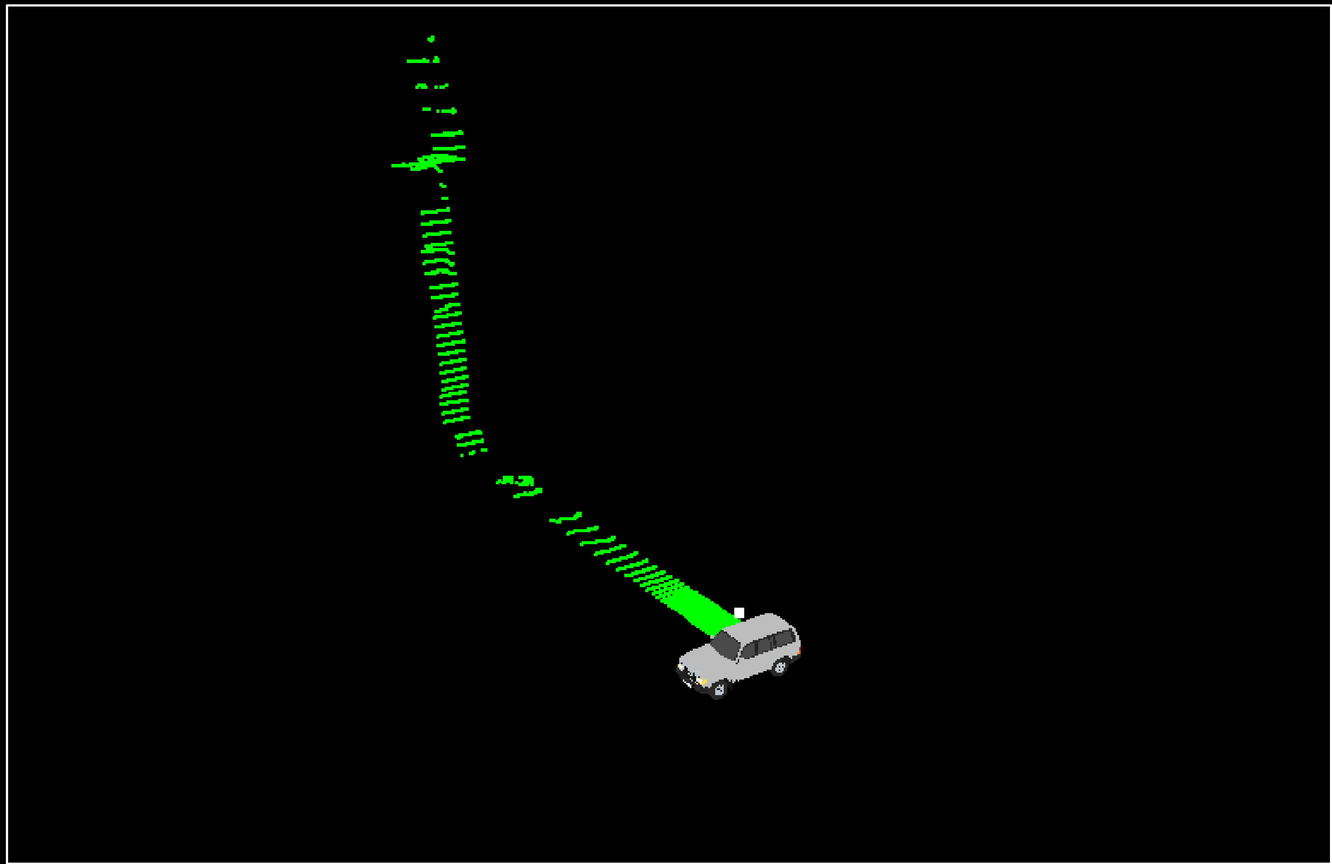
- We use a moving platform with SLAM to acquire the range data of the whole environment



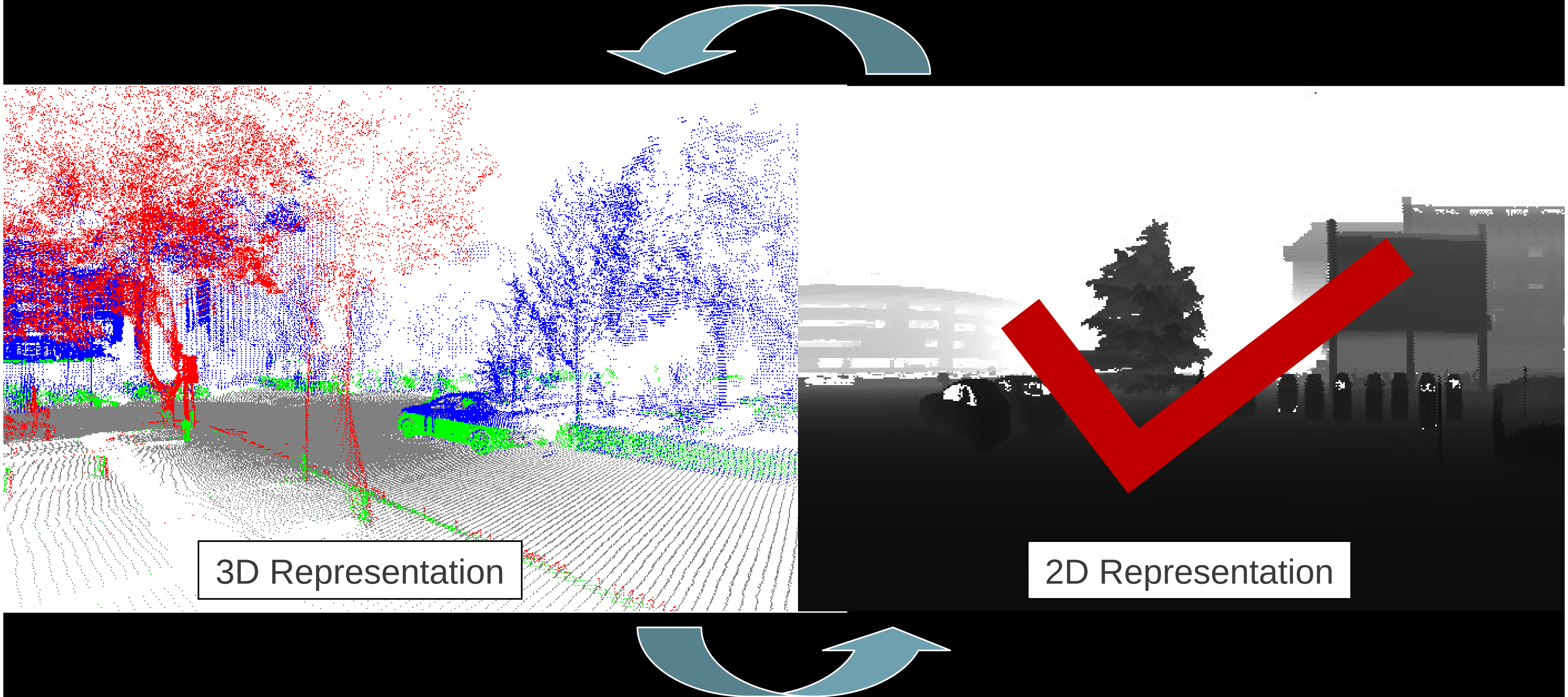
Laser #	Scan Mode	Major Purpose
L1	Horizontal	SLAM with MODT
L2	Downward	Road geometry
L3	Upward	Objects above the road
L4	Vertical Right	Objects to the right of the car
L5	Vertical Left	Objects to the left of the car

Our Platform

Introduction – Data Acquisition



Introduction – Data Acquisition

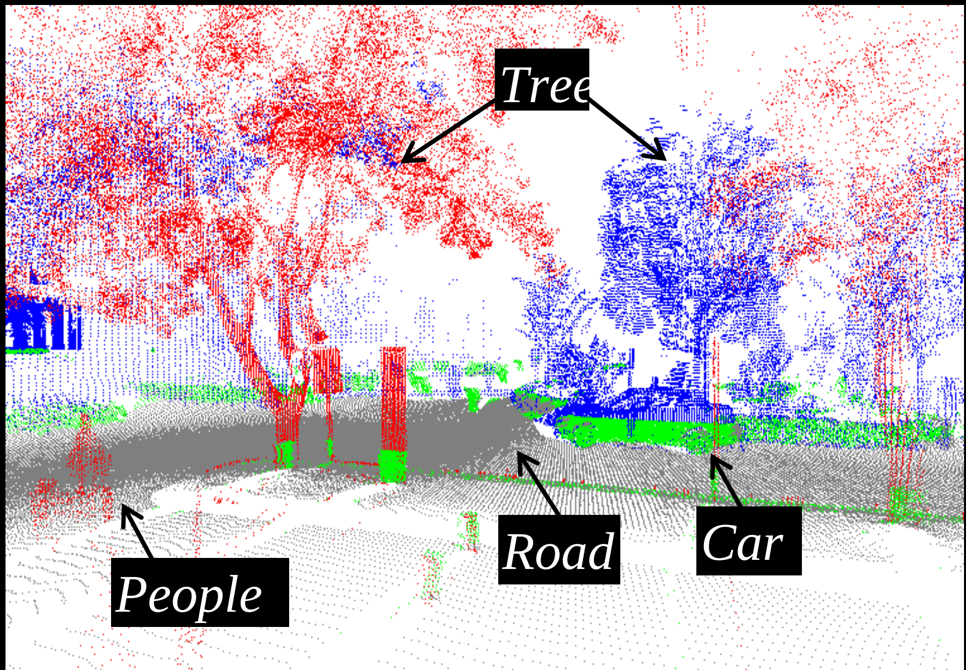


Equally convertible to each other

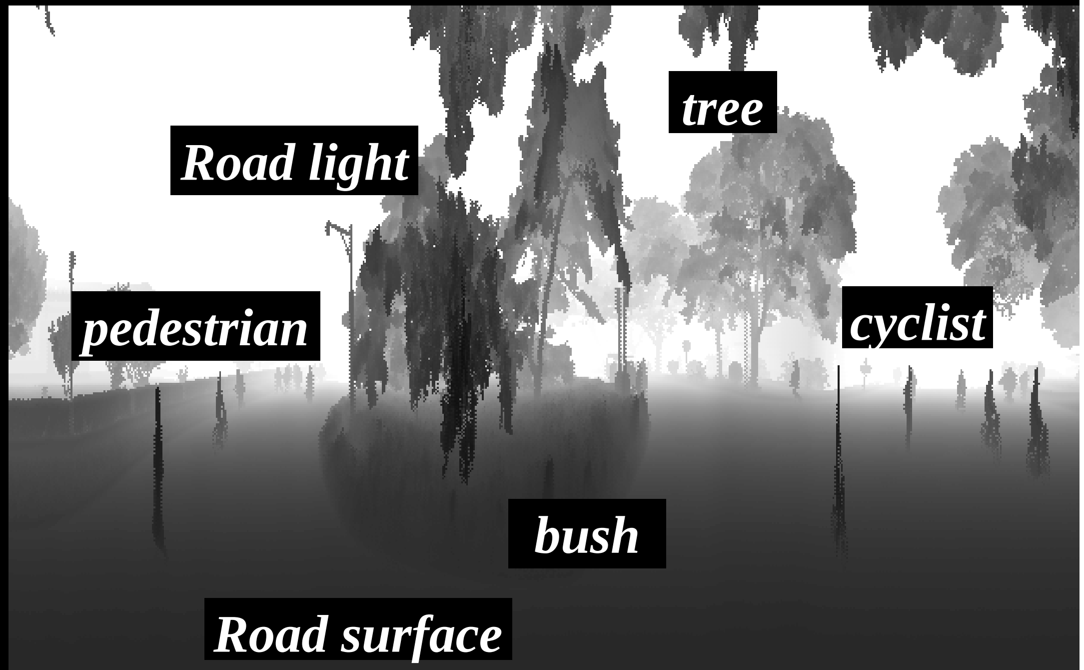
Same data organized in two different forms

Problem Formulation

- People can easily understand the scene



3D Laser Points



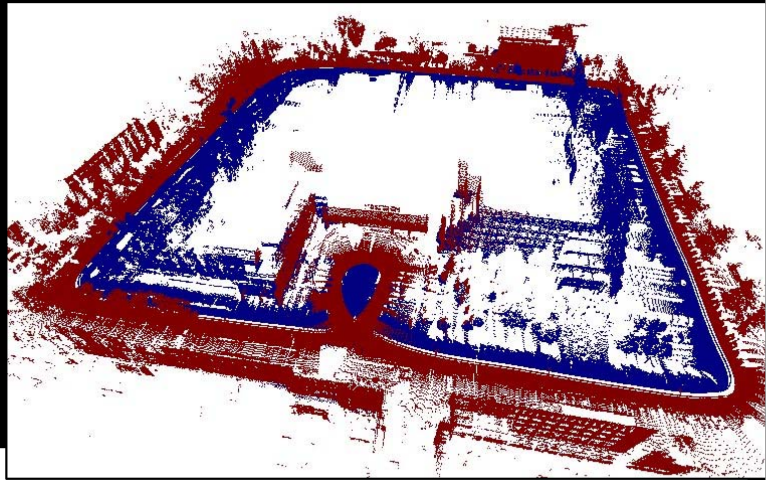
2D Range Image

Introduction - Our Objective

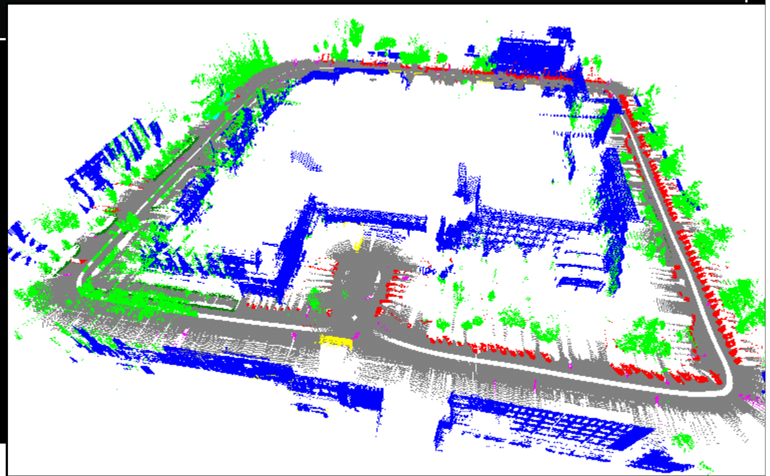
- We aim to provide a map with high-level representations.
- This map enables a robot to have semantic knowledge of the environment which is large and dynamic, such as objects, their types and so on.

Make the robot understand the scene!

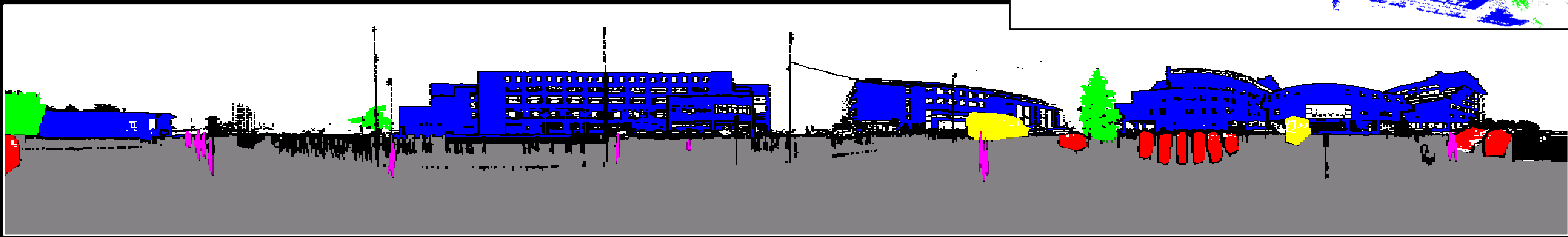
Problem Formulation



- **Input** - Range Image



- **Output** - Segments with semantics



Segmentation

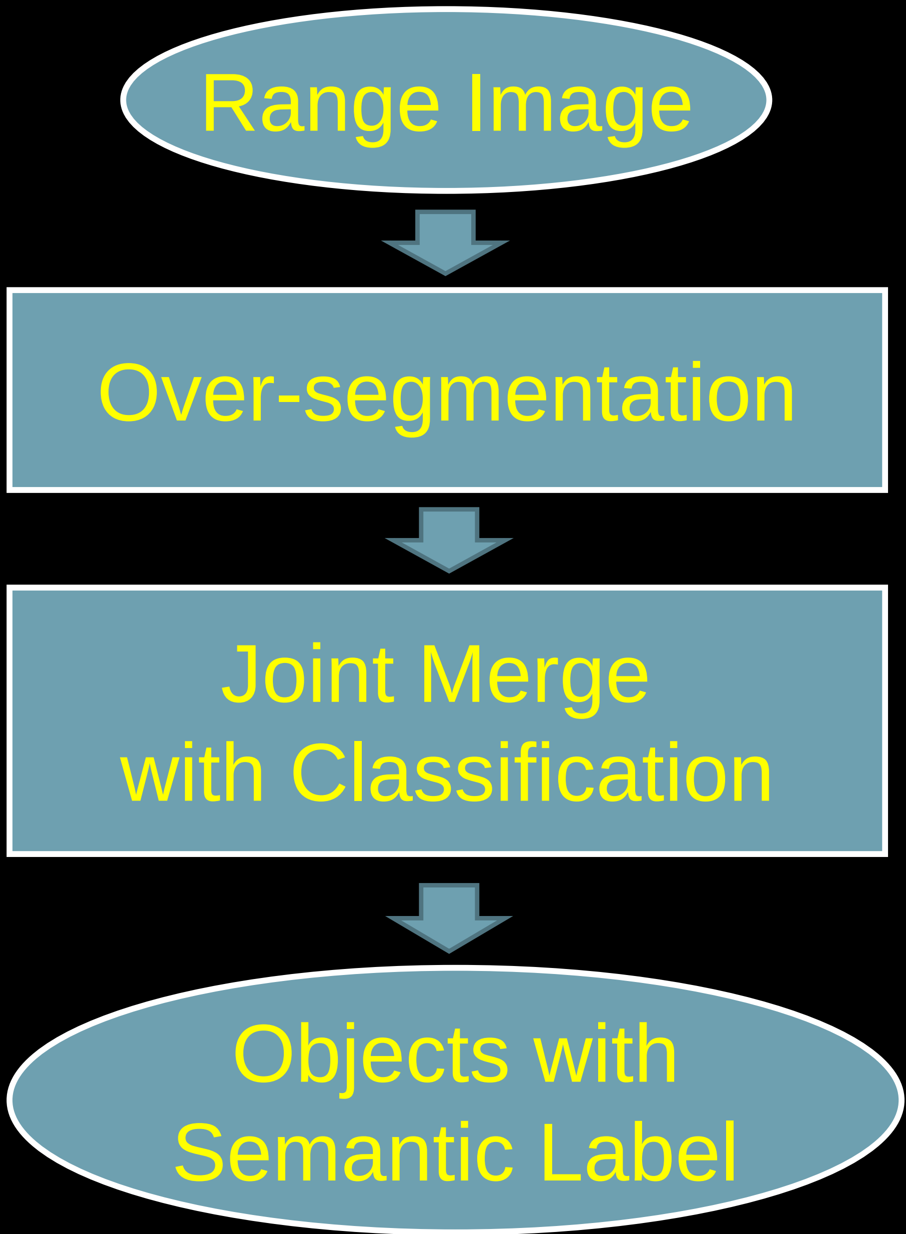
and

Classification

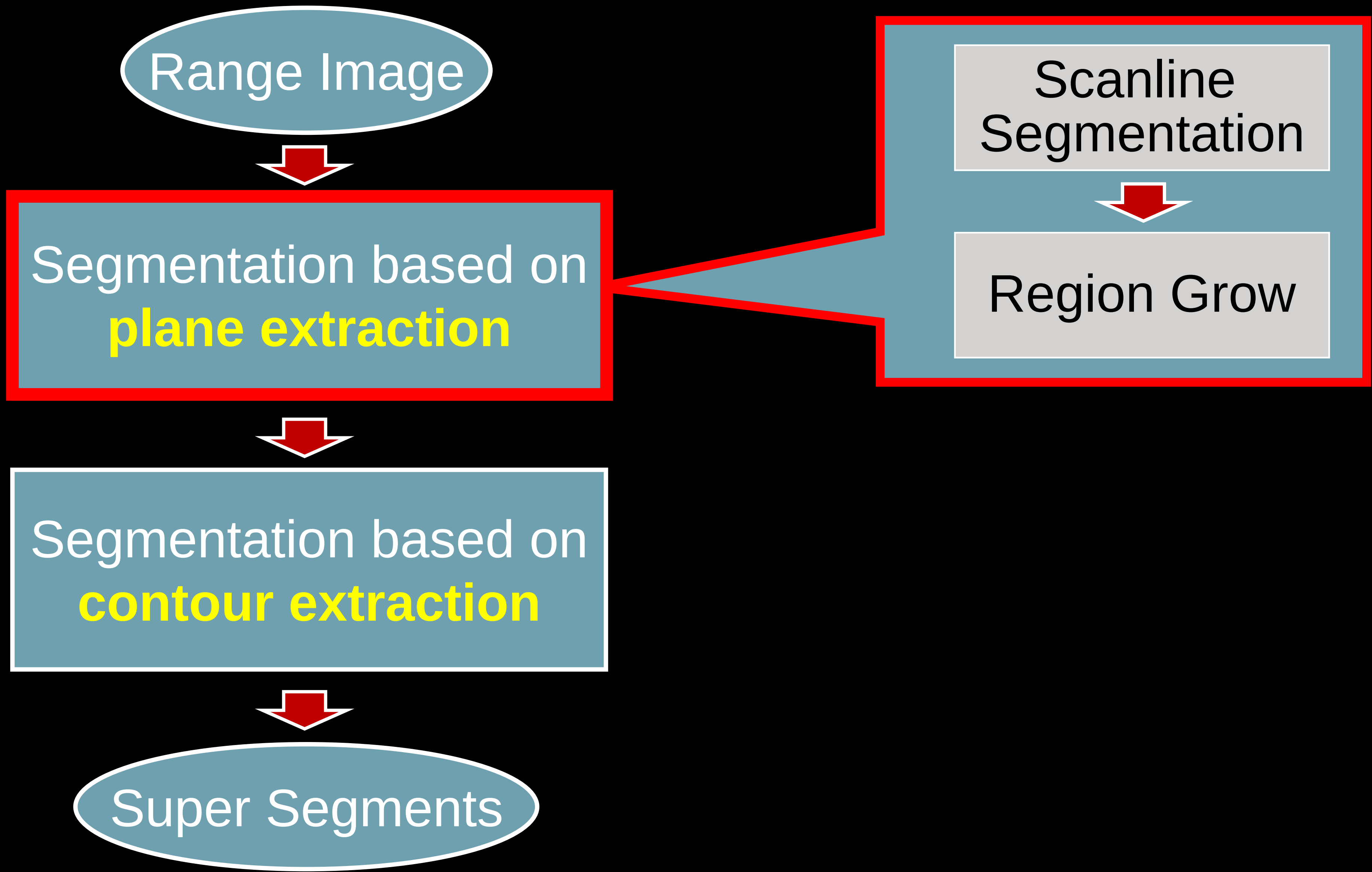
Traditional Method

- Sequential framework
  - segmentation -> classification
- Challenges
  - Many kinds of objects in complex environments
  - Based on an uniformed segmentation rule to all kinds of objects
    - Different objects might be segmented into one
    - One object might be segmented into different pieces
- Classification and segmentation should not be separated

Framework - Flowchart

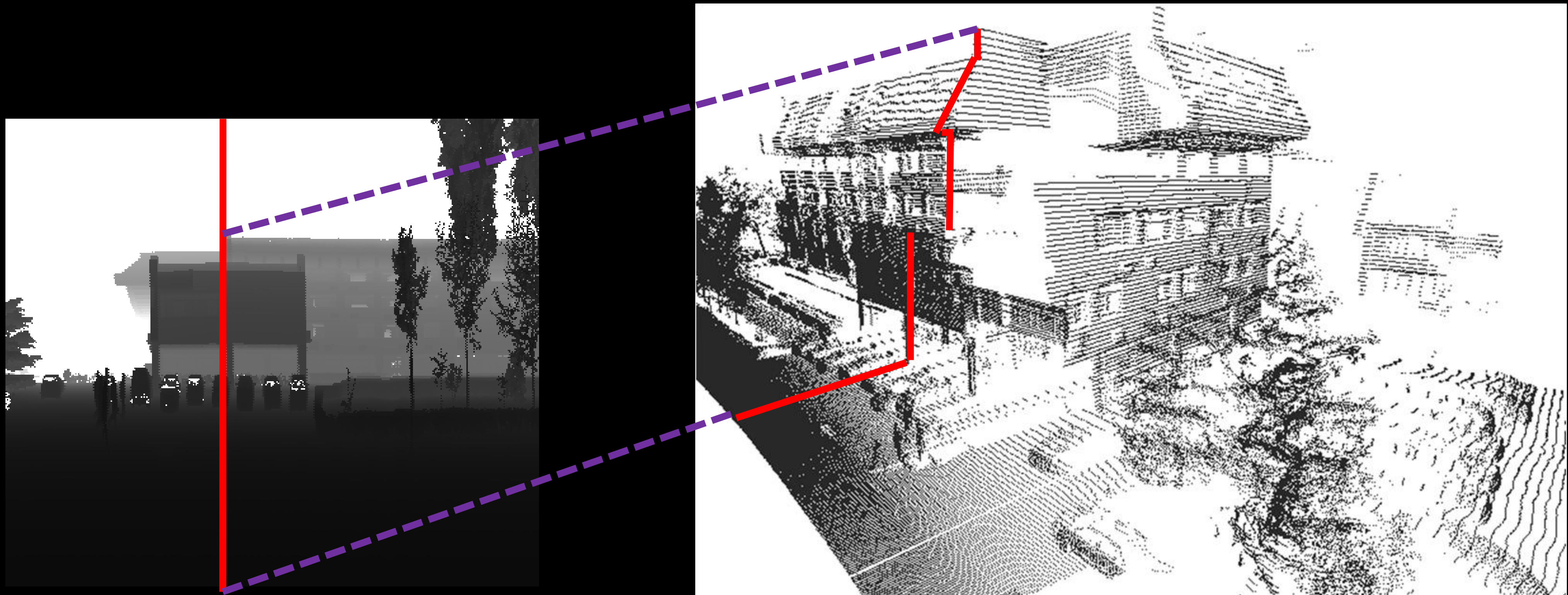


Over-segmentation



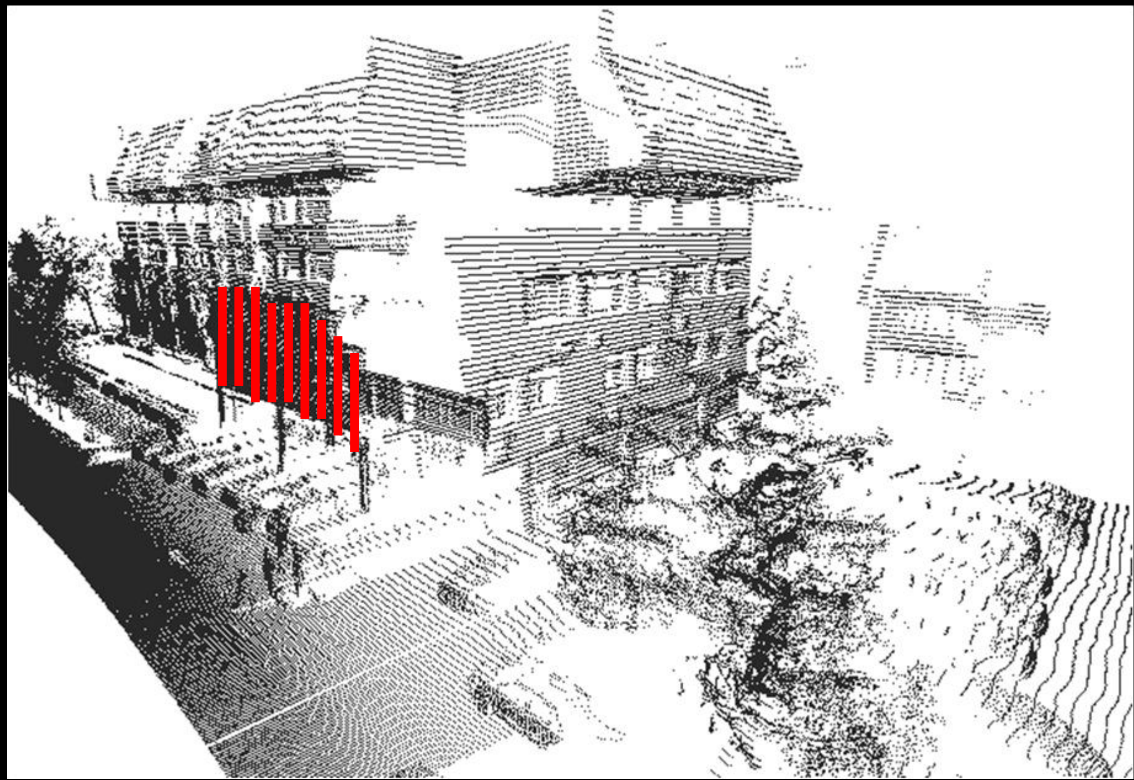
Plane extraction

- First, we separate every scanline into straight line segments.

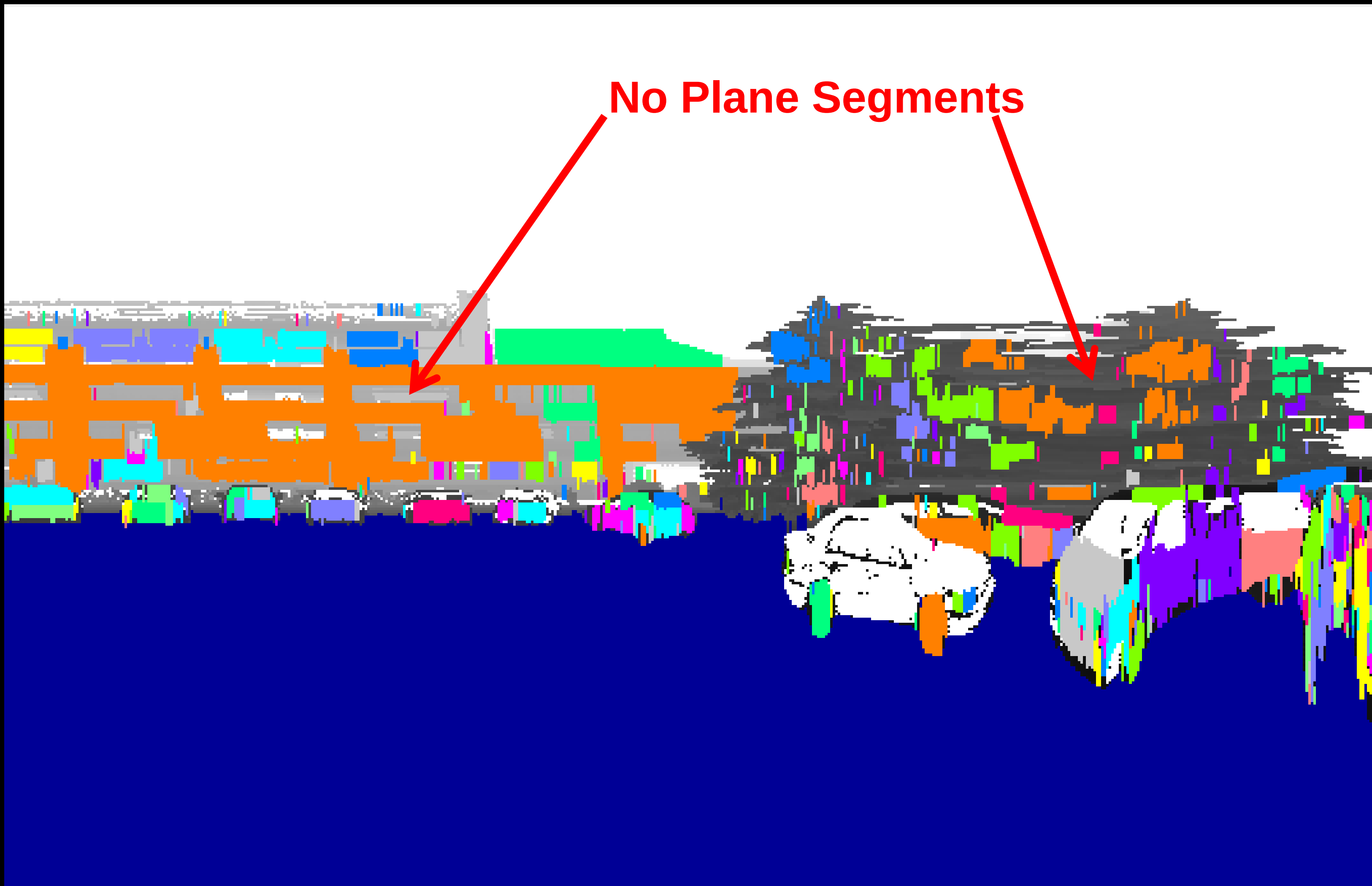


Plane extraction

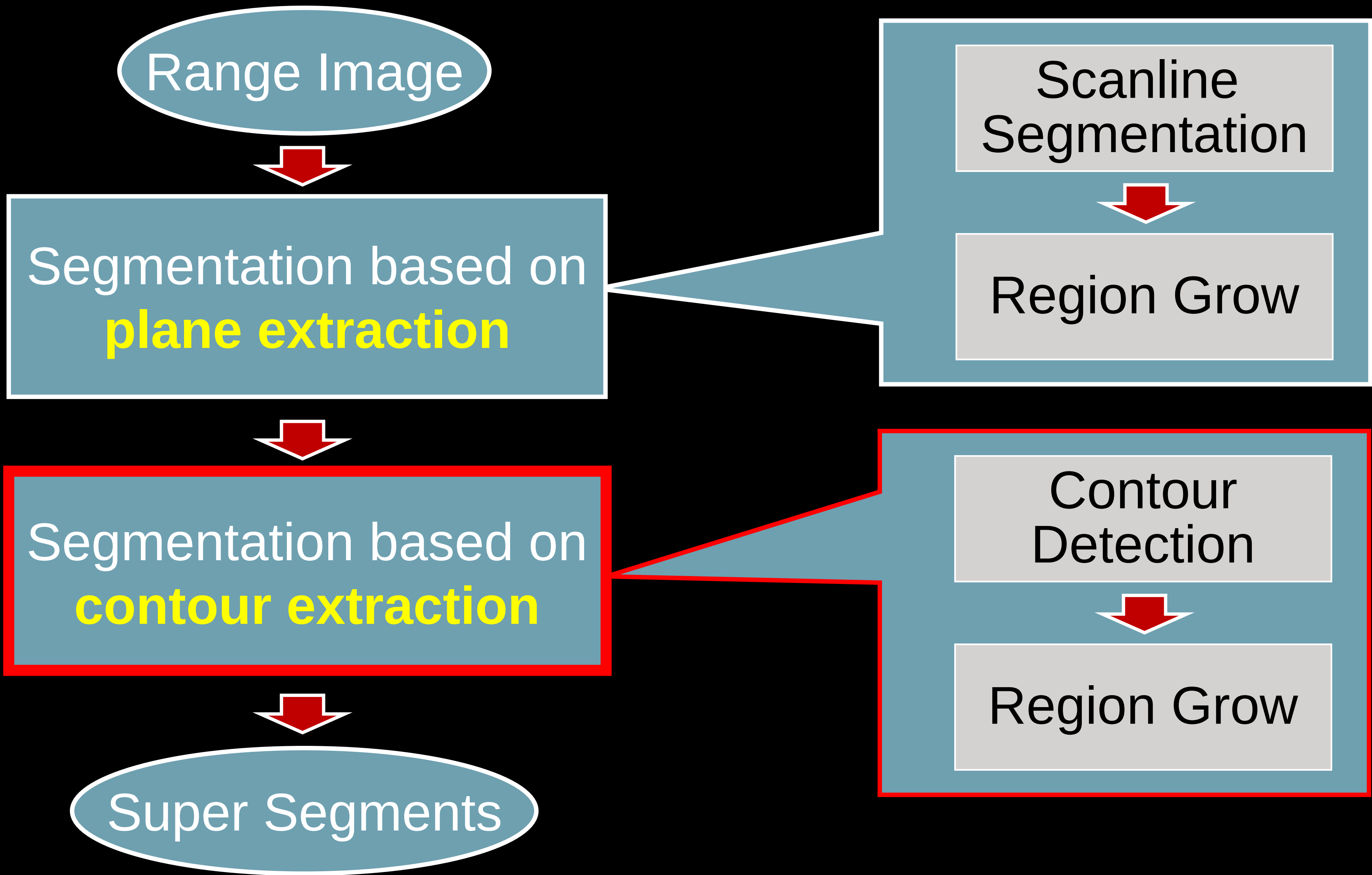
- Then, we grow all these straight line segments into planar regions.



Plane Extraction Results



Over-segmentation

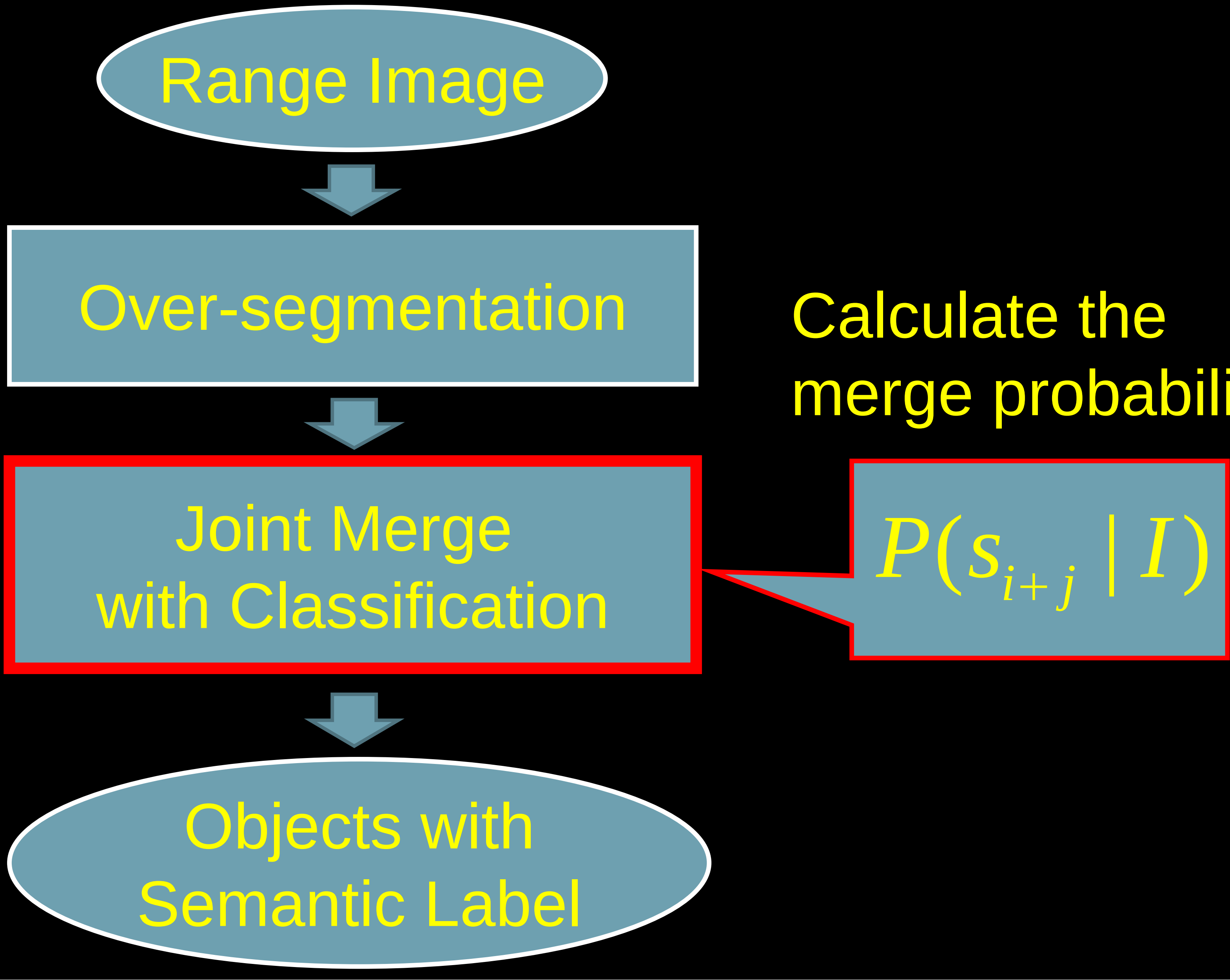


Contour Detection

Red Point: contour point

Over-segmentation Results

Flowchart Review



Joint Merge with Classification

$$P(s_{i+j} \mid I) \propto \sum_{l \in L} P(y_i = l \mid I) \cdot P(y_j = l \mid I) \cdot P(s_{i+j} \mid y_{i+j} = l, I)$$

The probability for a segment to be a certain class
Segment Classification

Given object class, the likelihood of two segments be the measurement to a single object
Likelihood

Segments Classification

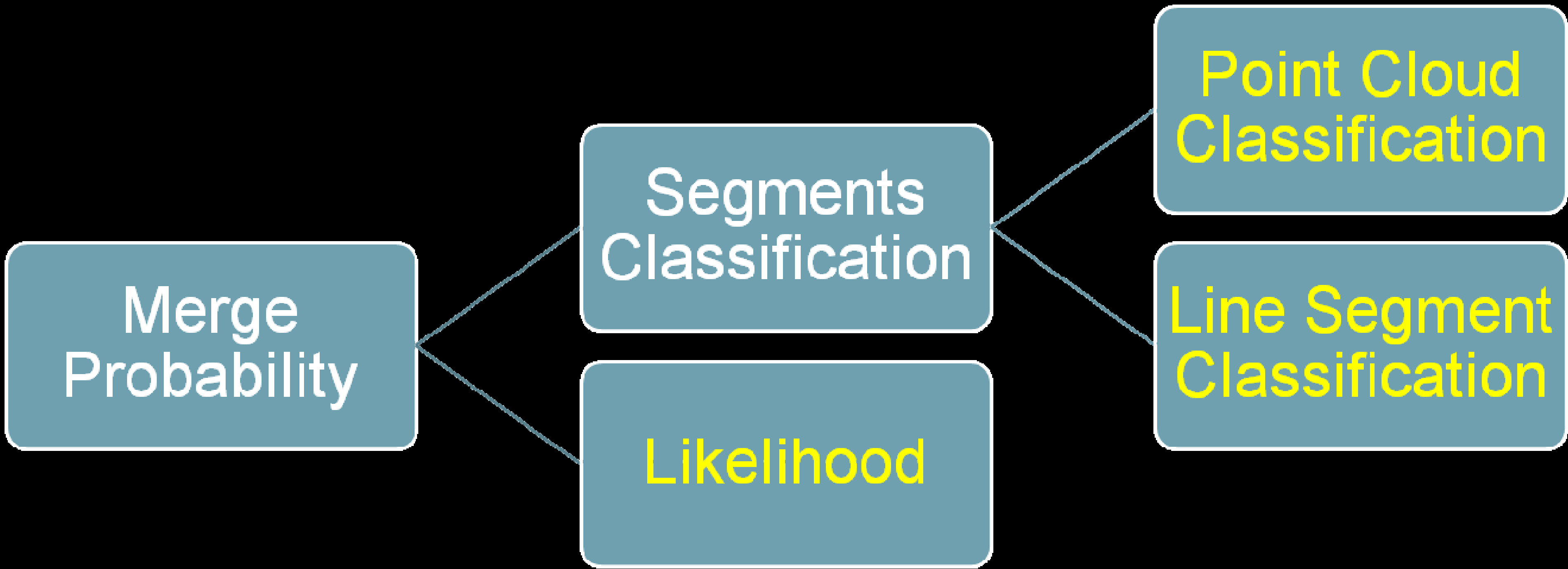
$$\sum_{l \in L} \underbrace{P(y_i = l \mid I)} \cdot P(y_j = l \mid I)$$

$$P(y_i = l \mid I) = \frac{1}{Z} \underbrace{P(\bigcup_k y_i^{(k)} = l \mid I)}_{\text{Points Cloud Classification}} \bullet \underbrace{\prod P(y_i^{(k)} = l \mid I)}_{\text{Line Segment Classification}}$$

Points Cloud Classification

Line Segment Classification

Joint Merge with Classification



Segments Classification

- Point Cloud Classification
  - SVM
- Line Segment Classification
  - Naive Bayesian Classification

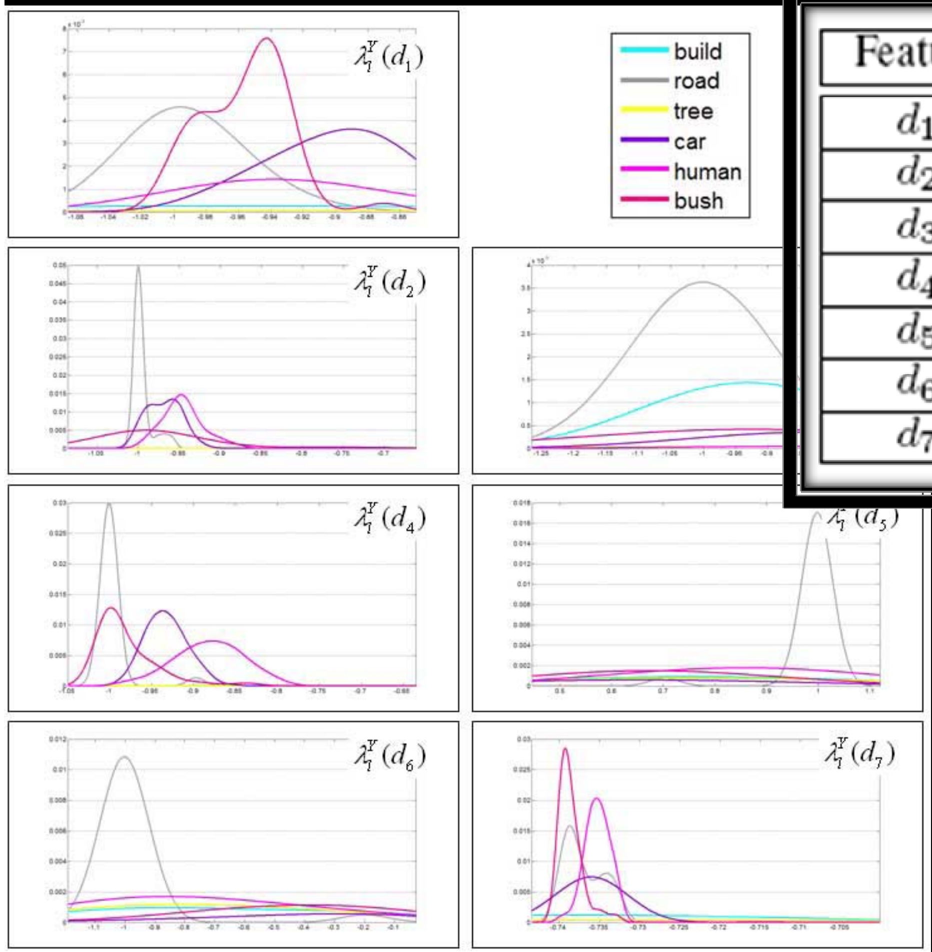
Training Sample

- We only use a small number of samples to train the point cloud classifier.

Class	Line Segments	Point Clouds
Building	9394	96
Road	10714	23
Tree	4122	148
Car	6080	41
People	394	120
Bush	1176	39
Bus	253	1
Total	32133	468

Feature Selection

- SVM - We selected 7 most discriminative features among more than 30 features

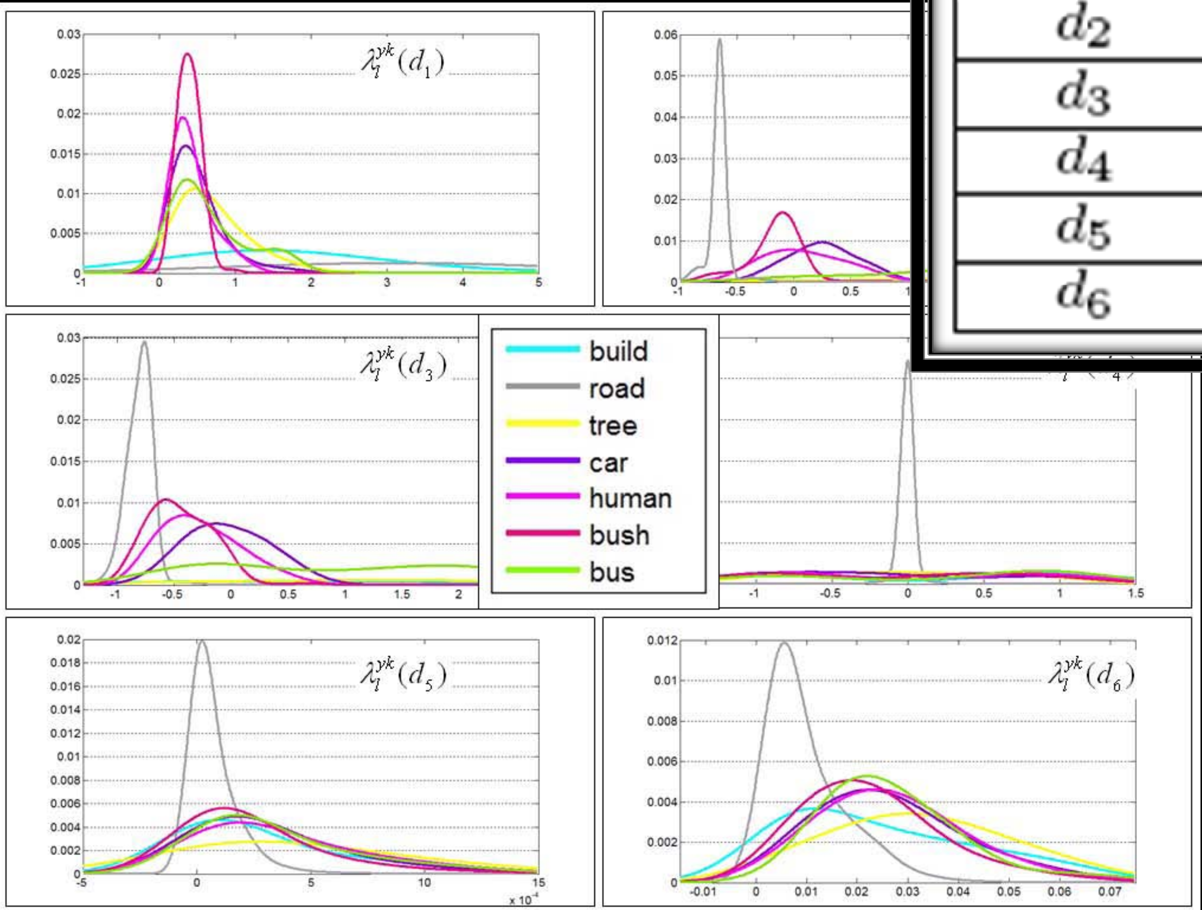


Feature	Definition
d_1	minimal height value
d_2	maximal height value
d_3	ratio of boundary point number vs total point number
d_4	mean of height values
d_5	variance of a histogram distribution on normal vectors
d_6	major picks of a histogram distribution on normal vectors
d_7	ratio of width vs. length

Feature Selection

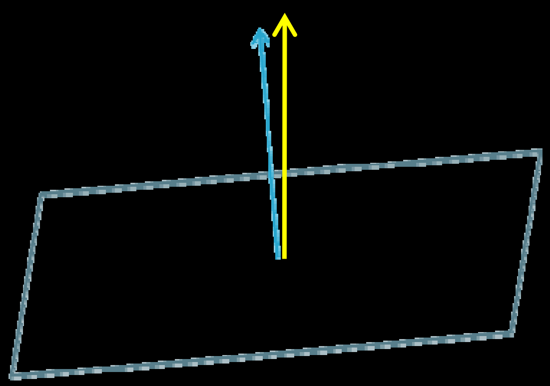
- Naive Bayesian Classification - We selected 6 features

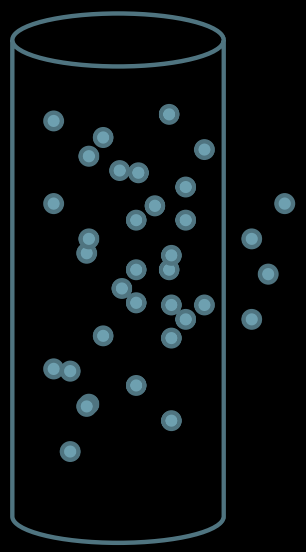
Feature	Definition
d_1	length of the scan line segment
d_2	maximal height value
d_3	minimal height value
d_4	Z factor of the directional vector
d_5	mean of line regression
d_6	variance of line regression

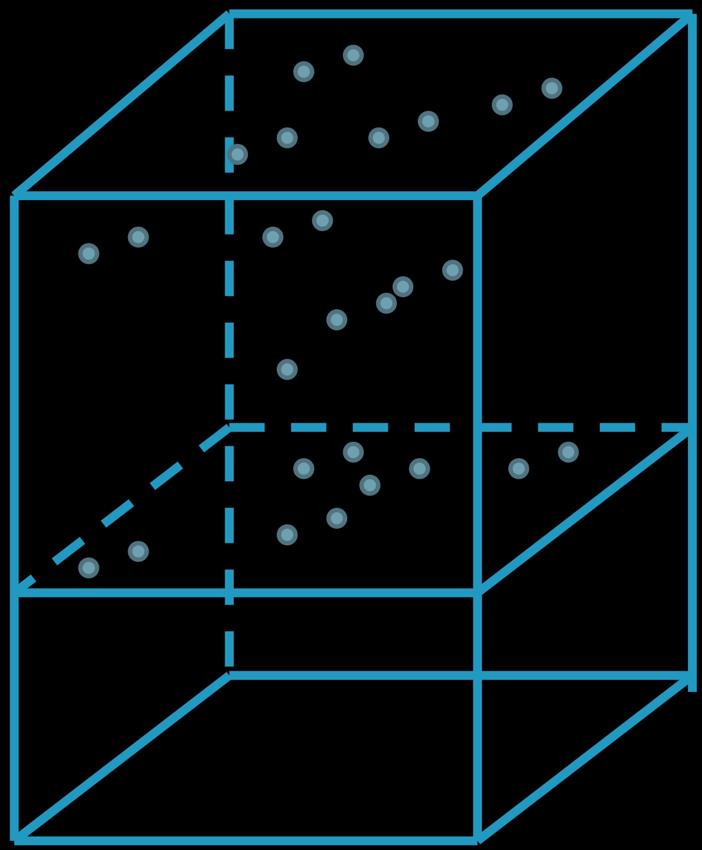


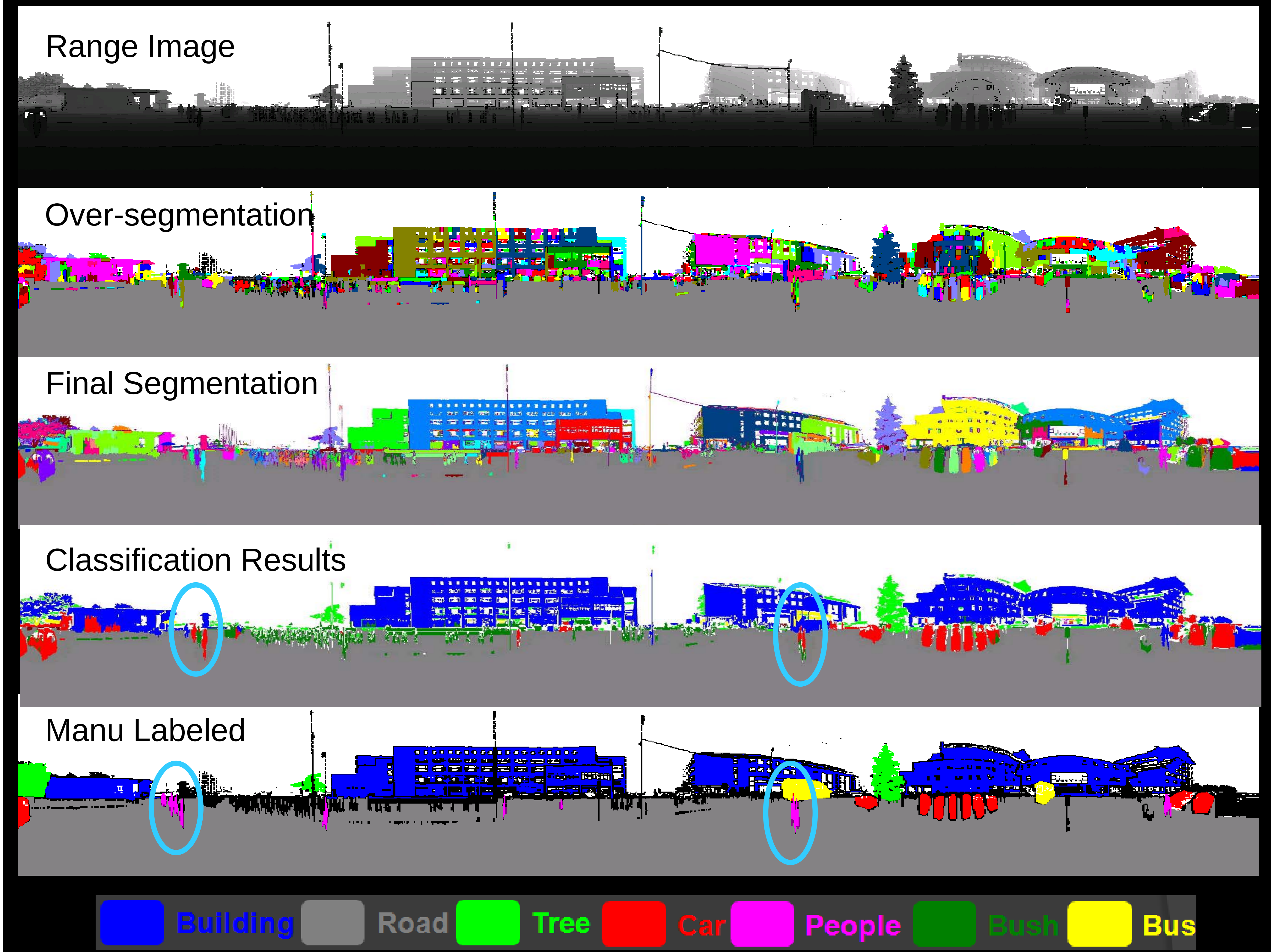
Likelihood

- 7 classes, 7 models
  - Plane fitting for road and building
  - Cube fitting for car, bus and bush
  - Cylinder fitting for people
  - Line fitting for tree



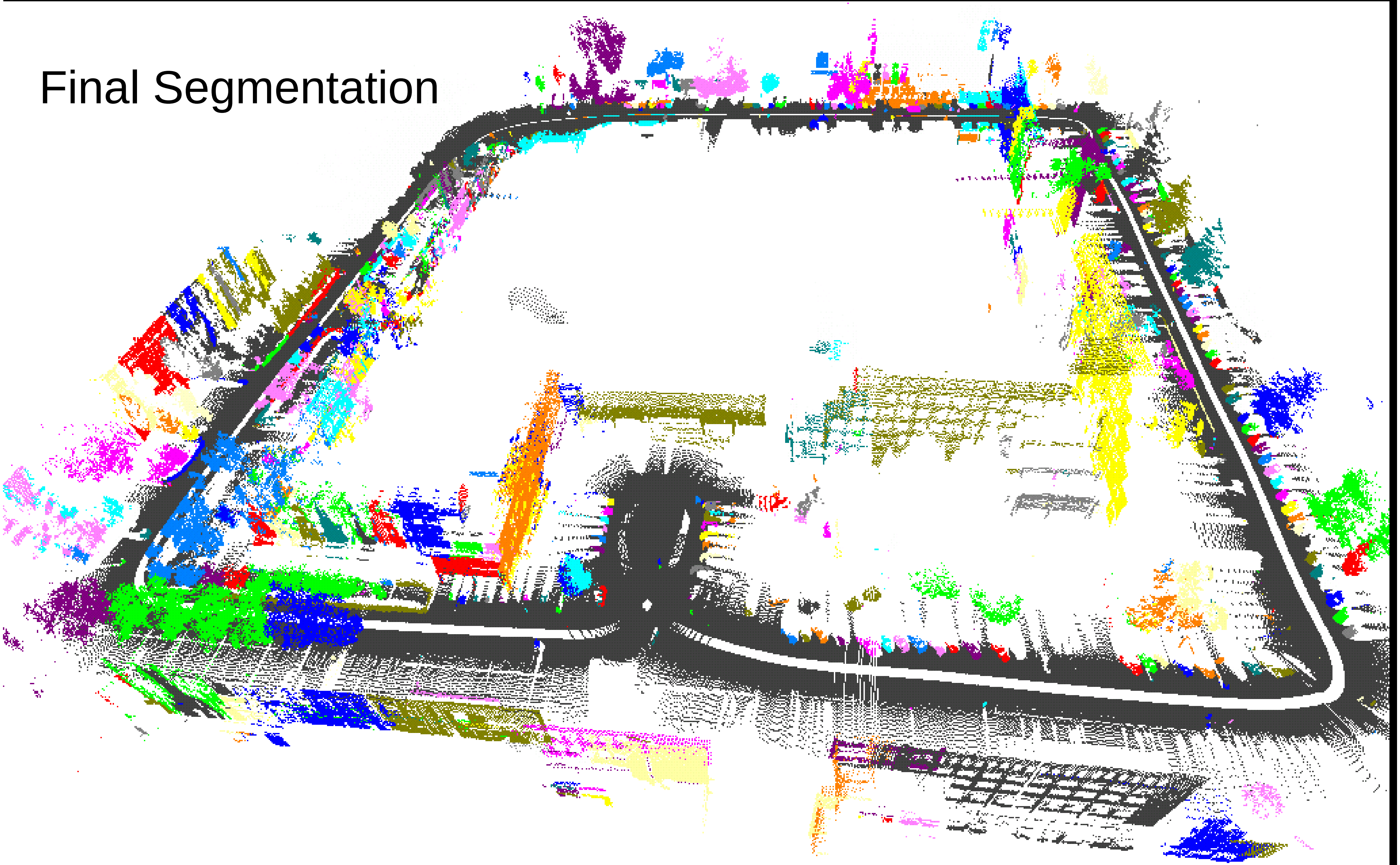




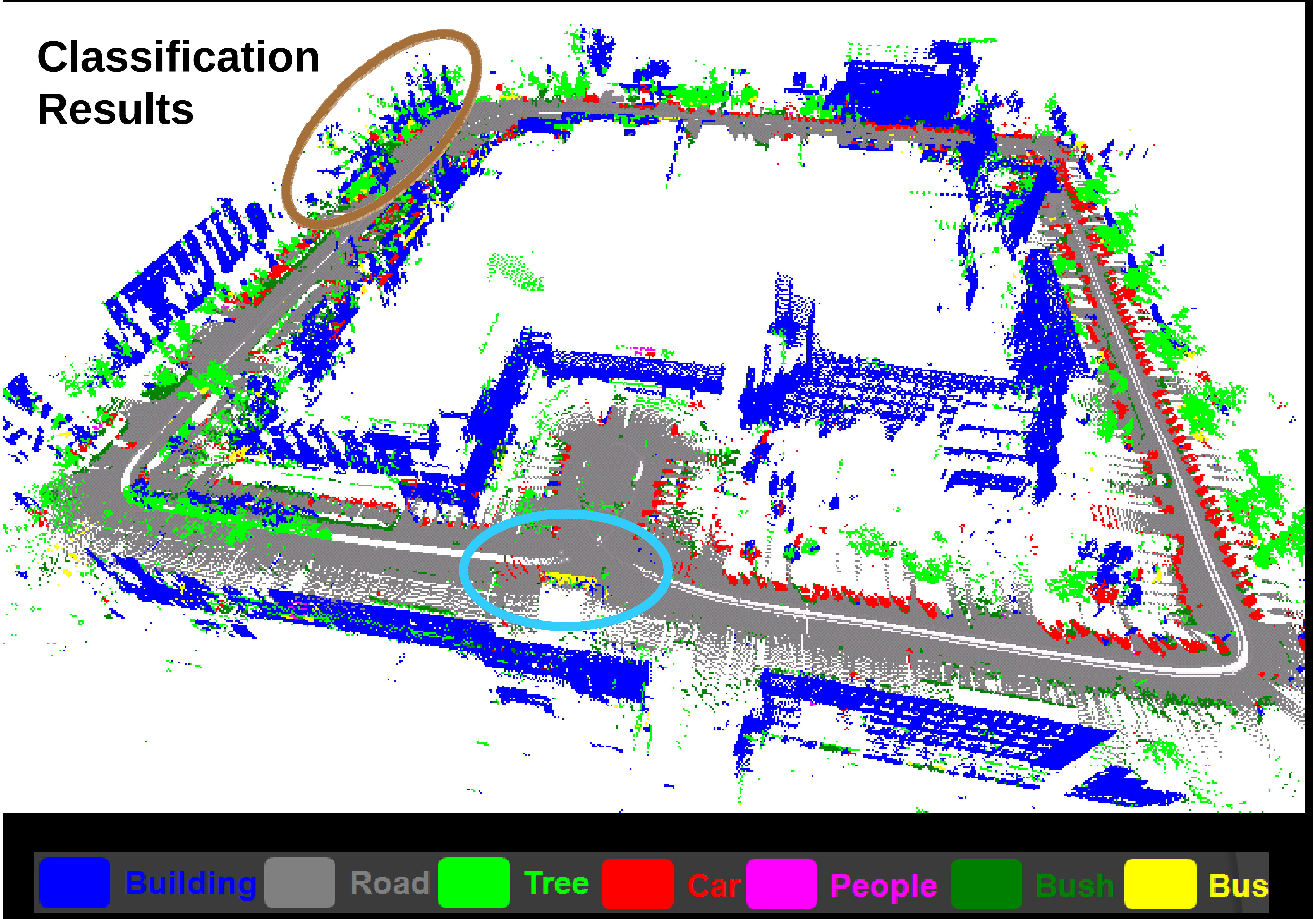


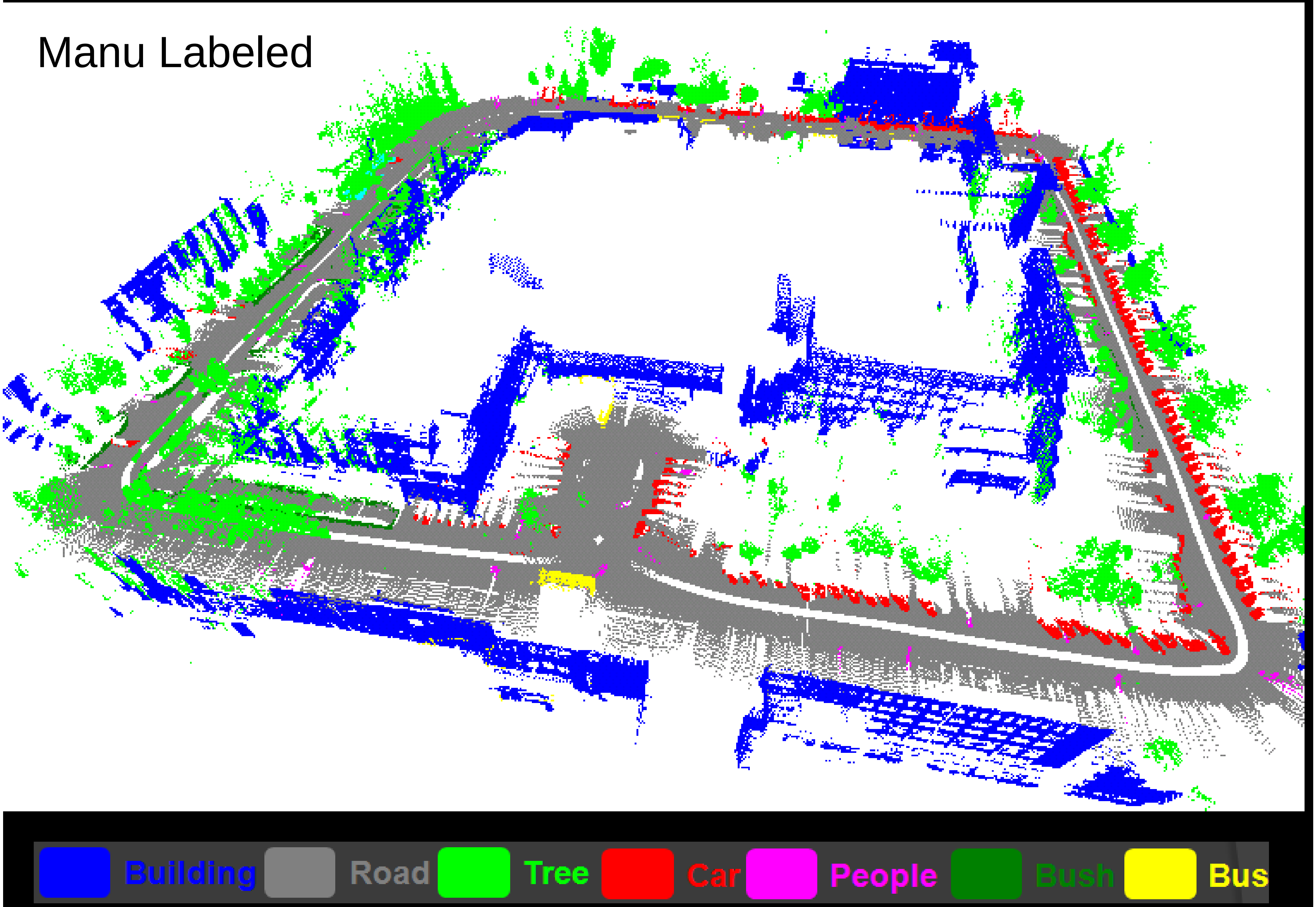
Range Image
Over-segmentation
Final Segmentation
Classification Results
Manu Labeled
Building
Road
Tree
Car
People
Bush
Bus

Final Segmentation



Classification Results




Manu Labeled
Building
Road
Tree
Car
People
Bush
Bus

Summary

- We develop a framework of joint segmentation and classification.
- The experimental results are encouraging.

- But there are still problems to be solved
  - Implementation of the framework needs to be improved.
  - Classification accuracy, especially people, are not satisfying due to limited training samples and partial observation.

Future work

- Improve our framework
  - How to deal with the segments containing no line segment
  - Points should be a special form of lines
- Make more training samples
  - We can make it together
  - Our data are available in

http://poss.pku.edu.cn

IEEE Int. Conf. on Robotics and Automation (ICRA), 2012

Computing Object-based Saliency in Urban Scenes Using Laser Sensing

*Yipu Zhao, M. He, H. Zhao, F. Davoine, and H. Zha

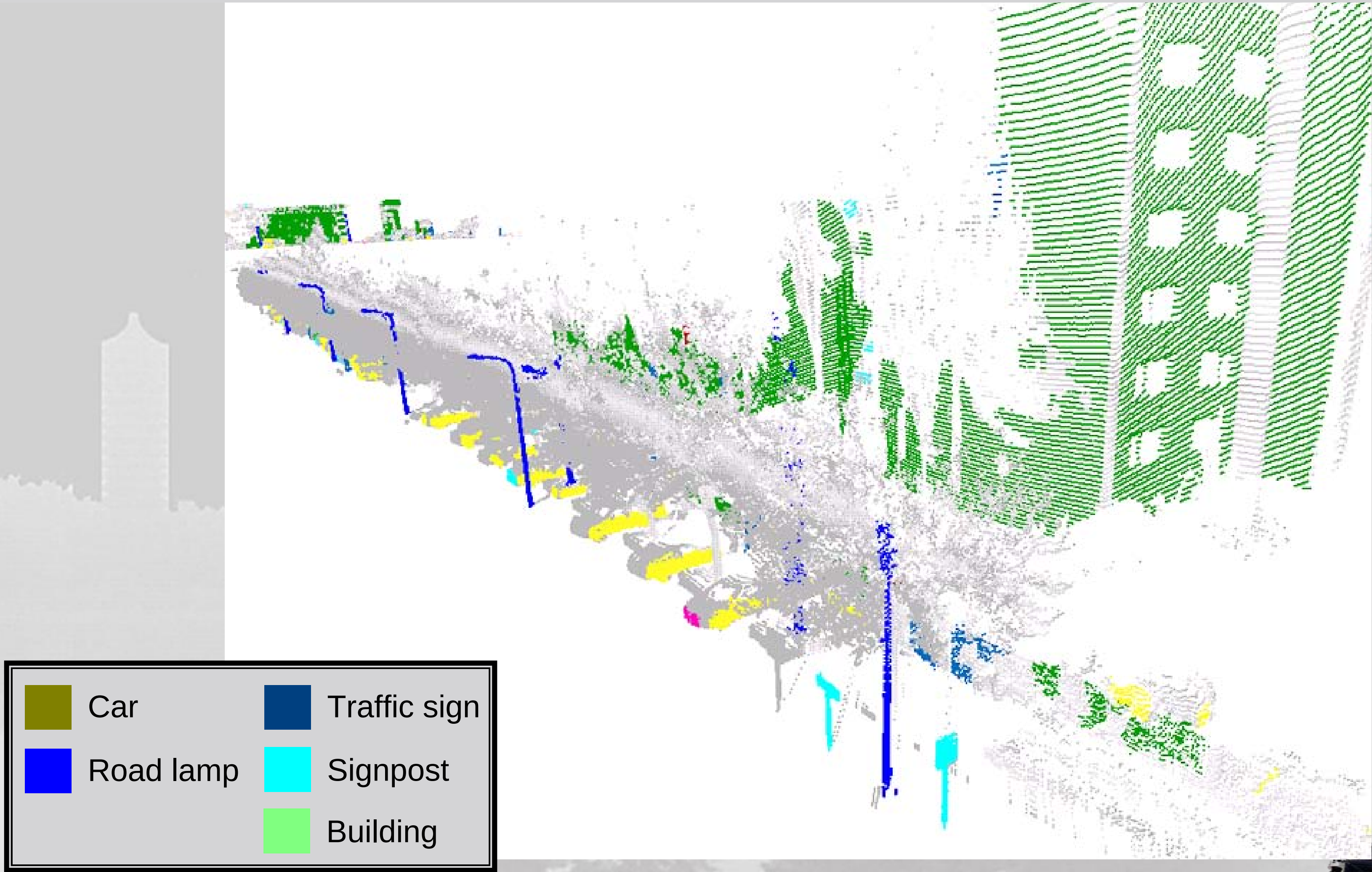
Department of EECS, Peking University
Sino-French Lab, CNRS & LIAMA

PEKING UNIVERSITY

POSS
PKU OMNI SMART SENSING

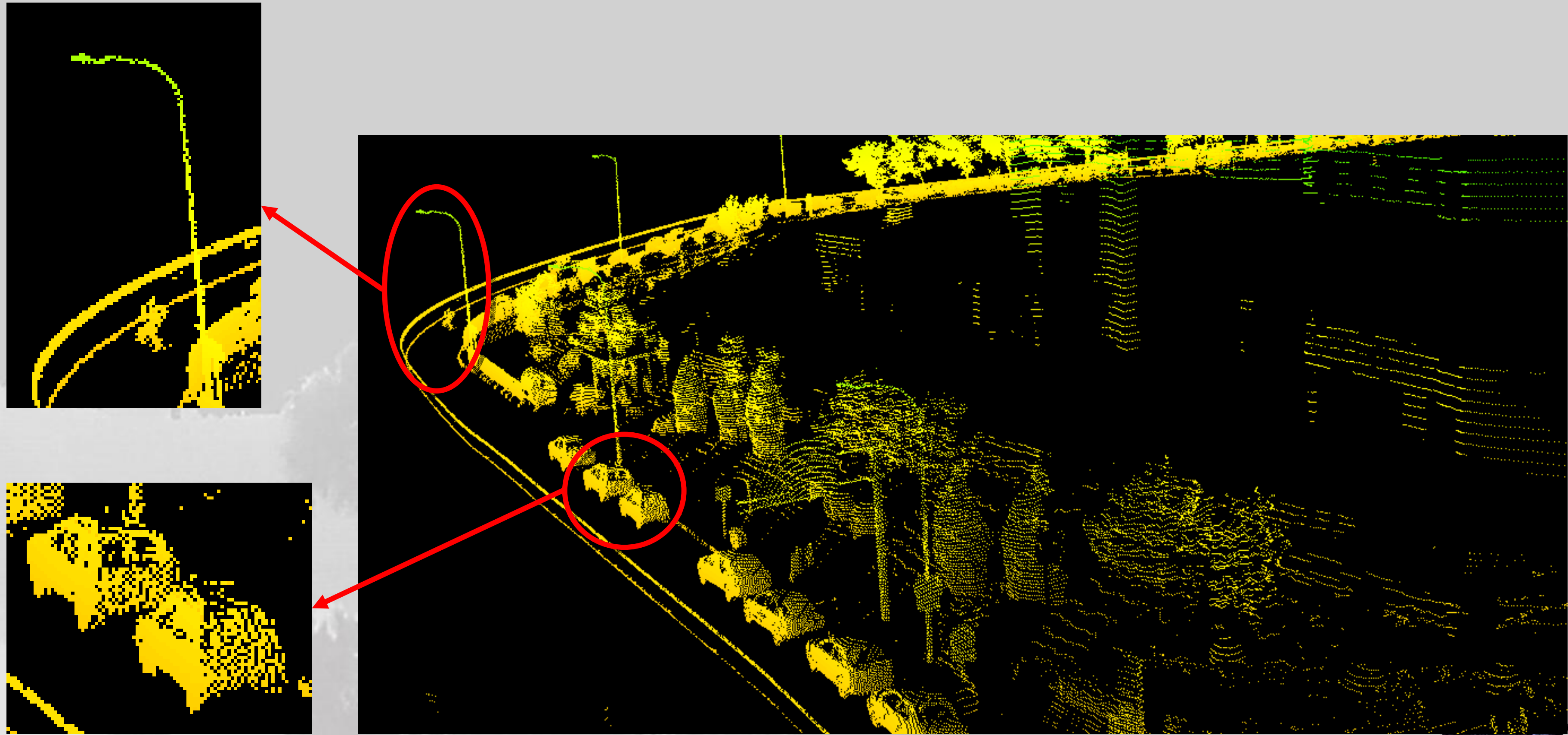
Motivation

- **Object discovery from mobile laser scanning.**



Background

- Different applications may concern different objects.
- Put more focus on the objects of interest.



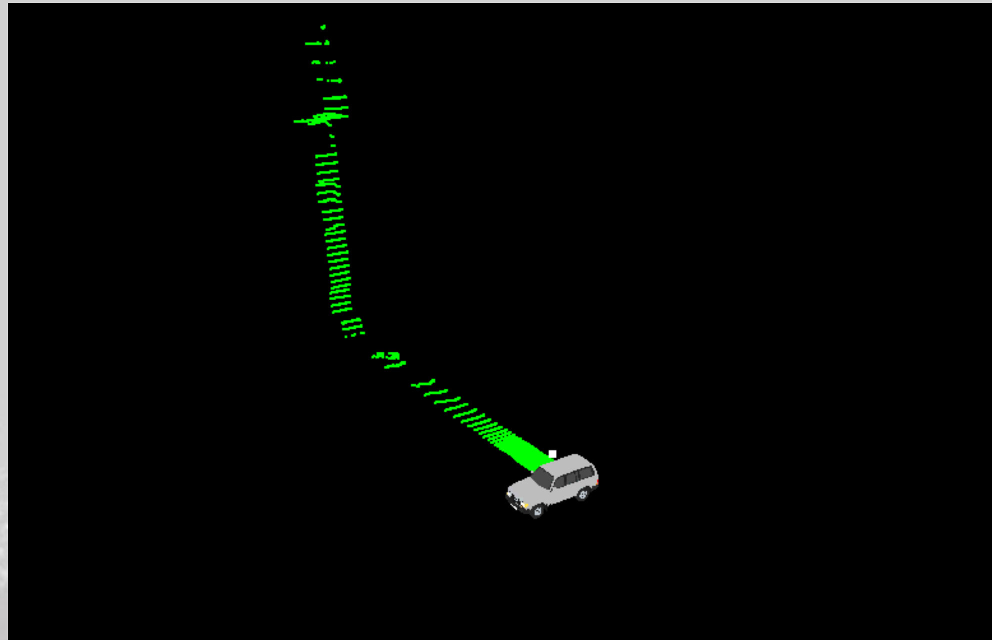
This research

Objective: Compute the object-based saliency of laser points

Laser points → **Computing Object-based Saliency** → Object Detection

Step1: Geometric Feature Extraction → **Step2**: Object Candidate Generation → **Step3**: Object-based Saliency Computing

Experimental platform



Step1: Geometric Feature Extraction

- Four types of geometric feature
  - Vertical line, horizontal line, vertical plane, horizontal plane

- Flowchart

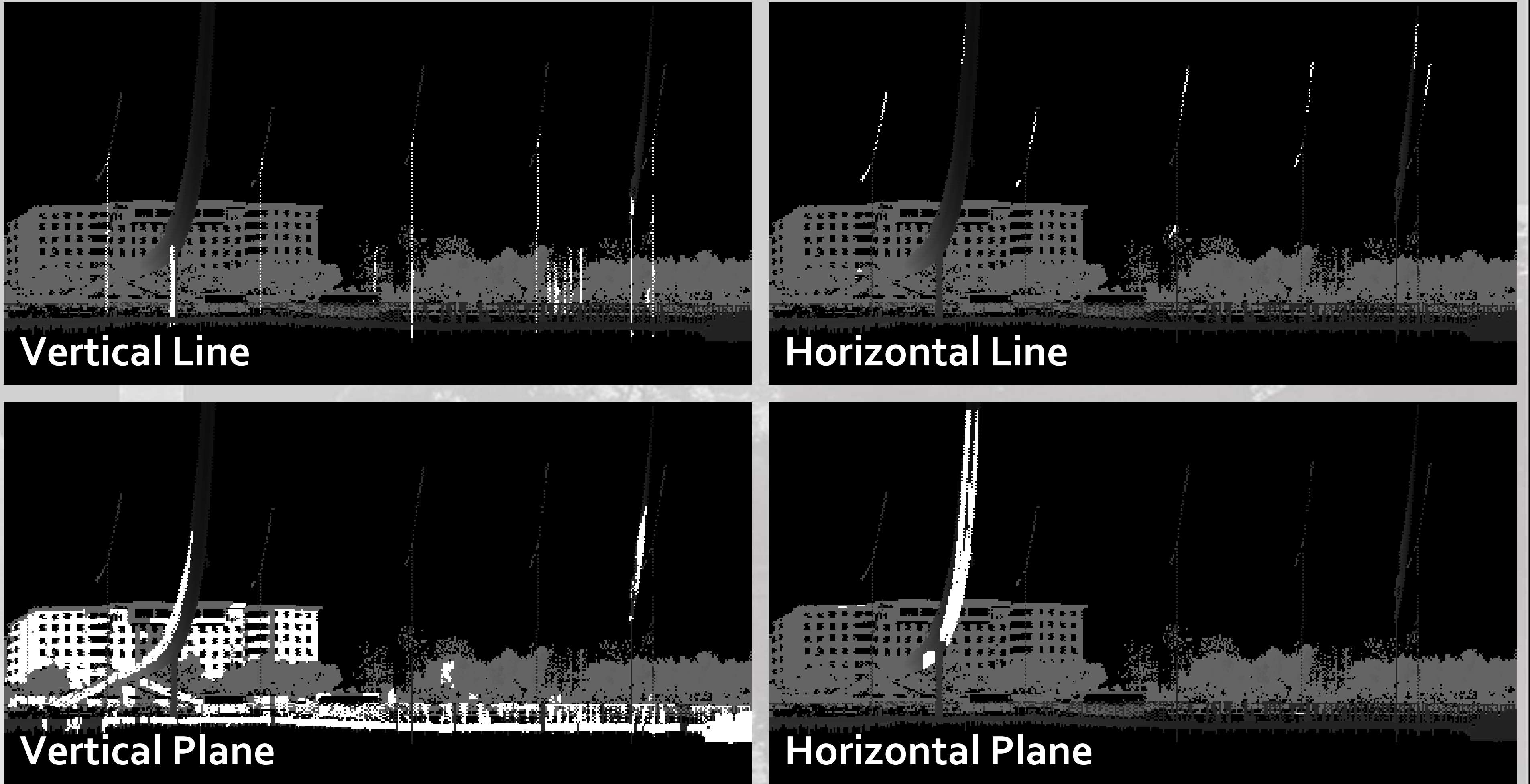
Range Image → Seed Selection → Region Growing → Geometric Features

Seed Selection:
1. Estimating some pixel-level features
2. Based on such features, a decision tree is applied to select seeds

Region Growing:
Starting from the seeds, all neighboring pixels with similar pixel-level features (e.g. norm) are grown in an iteration

1. Geometric Feature Extraction

- Extraction results



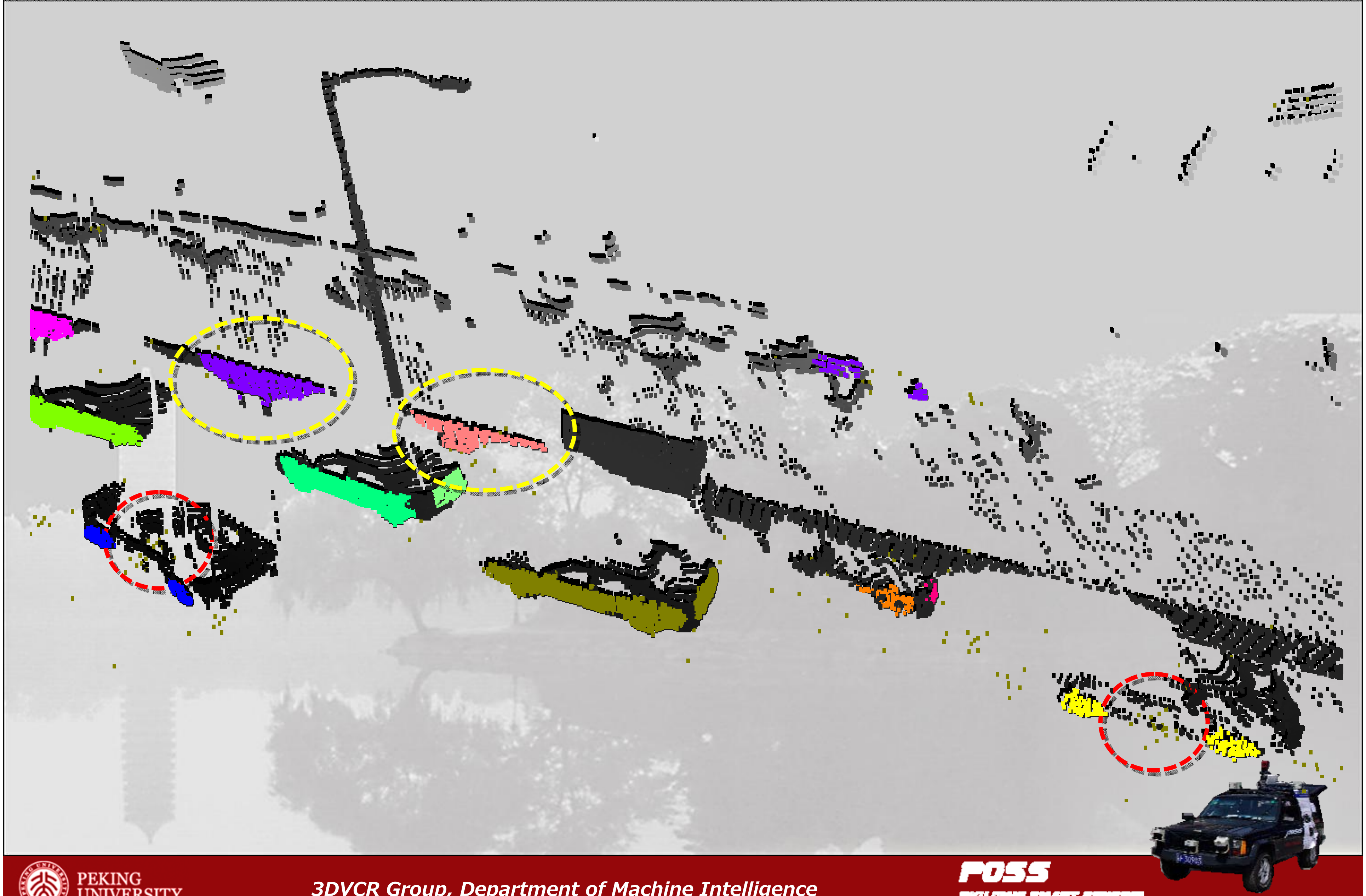
2. Object Candidate Generation

- For each class, the objects can be considered as combination of geometric features
  - Car: several surface planes
  - Road lamp: a long pole
  - Traffic sign: a big board with some supporting sticks

- The object candidates are generated by finding corresponding combination of geometric features

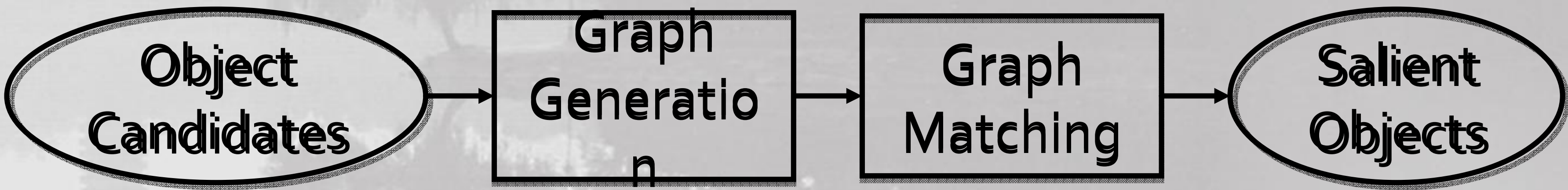
Geometric Features → Voting Candidate Centers → Clustering Centers → Object Candidates

2. Object Candidate Generation



3. Object-based Saliency Computing

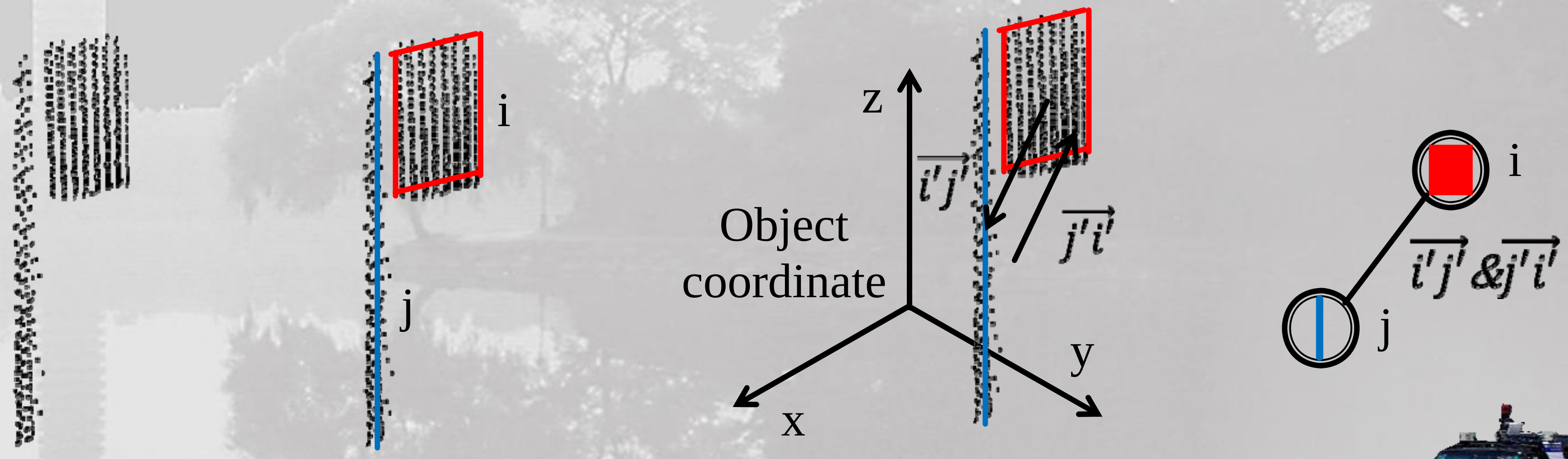
- Given the object candidates, the object-based saliency is depend on
  - Type & size of the related geometric features
  - Spatial relationship of different geometric features
- To contain these two information
  - A graphical object representation is introduced
- Flowchart



3.1 Graph Generation

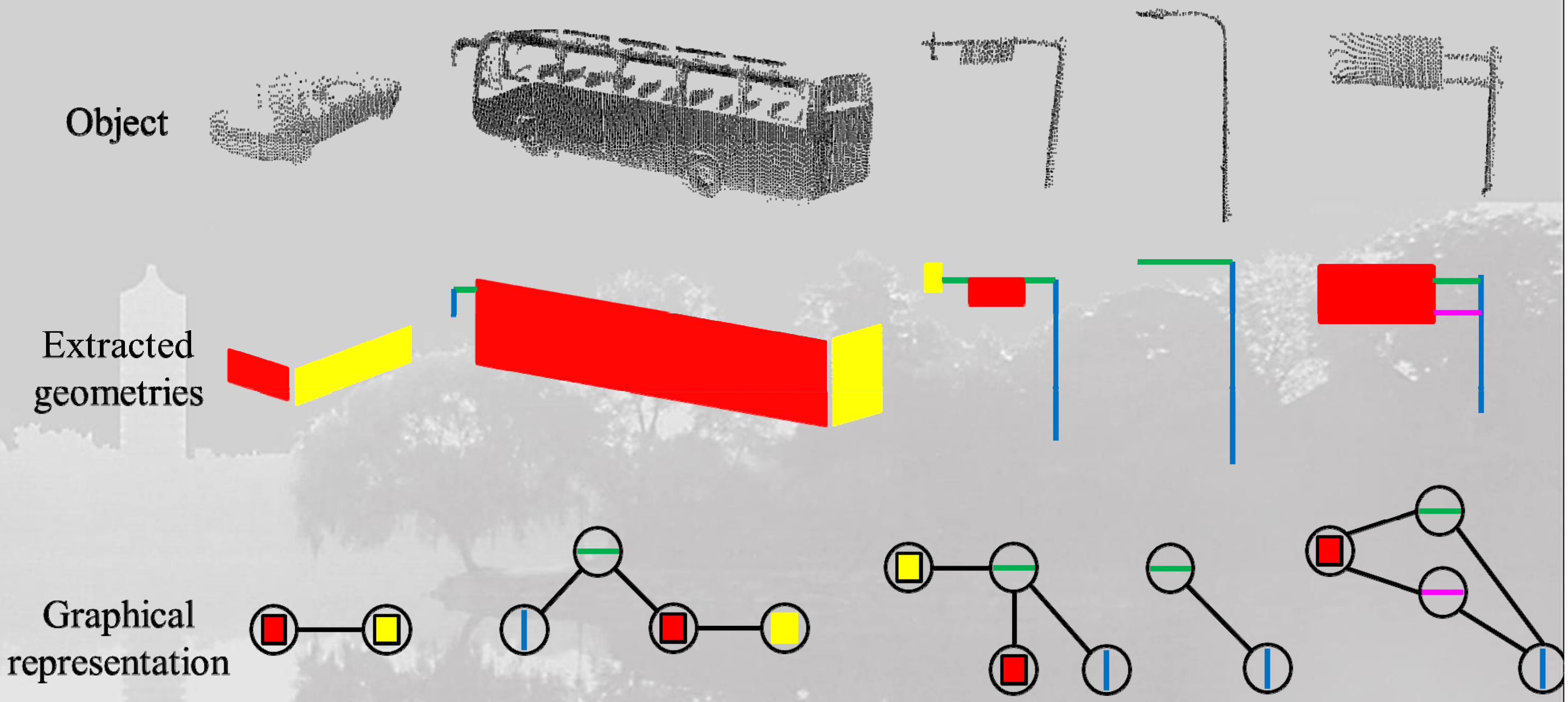
- Objective: build a invariant graph representation for each object candidate
  - Node: Type & size of geometric features
  - Edge: Spatial relationship of different geometric features

- An example



3.1 Graph Generation

- Some model graphs of salient objects



3.2 Graph Matching

- Objective: Given a model graph $G_m = (N_m, E_m)$ & a data graph $G_d = (N_d, E_d)$, a matching score will be evaluated between them

- Step 1. run inexact graph matching
  - Only concern edge attributes
  - Generate 2 sub-graphs $G_{ms} = (N_{ms}, E_{ms})$ & $G_{ds} = (N_{ds}, E_{ds})$

- Step 2. evaluate matching score

$$D(G_m, G_d) = \max\left(\sum_{k=0}^{card(N_{ms})} S_{N_{ms}^k}, \sum_{k=0}^{card(N_{ds})} S_{N_{ds}^k}\right) / \max\left(\sum_{k=0}^{card(N_m)} S_{N_m^k}, \sum_{k=0}^{card(N_d)} S_{N_d^k}\right)$$

where N^k denotes for the kth node in node set N,
and S_n is the area of node n's corresponding geometric feature

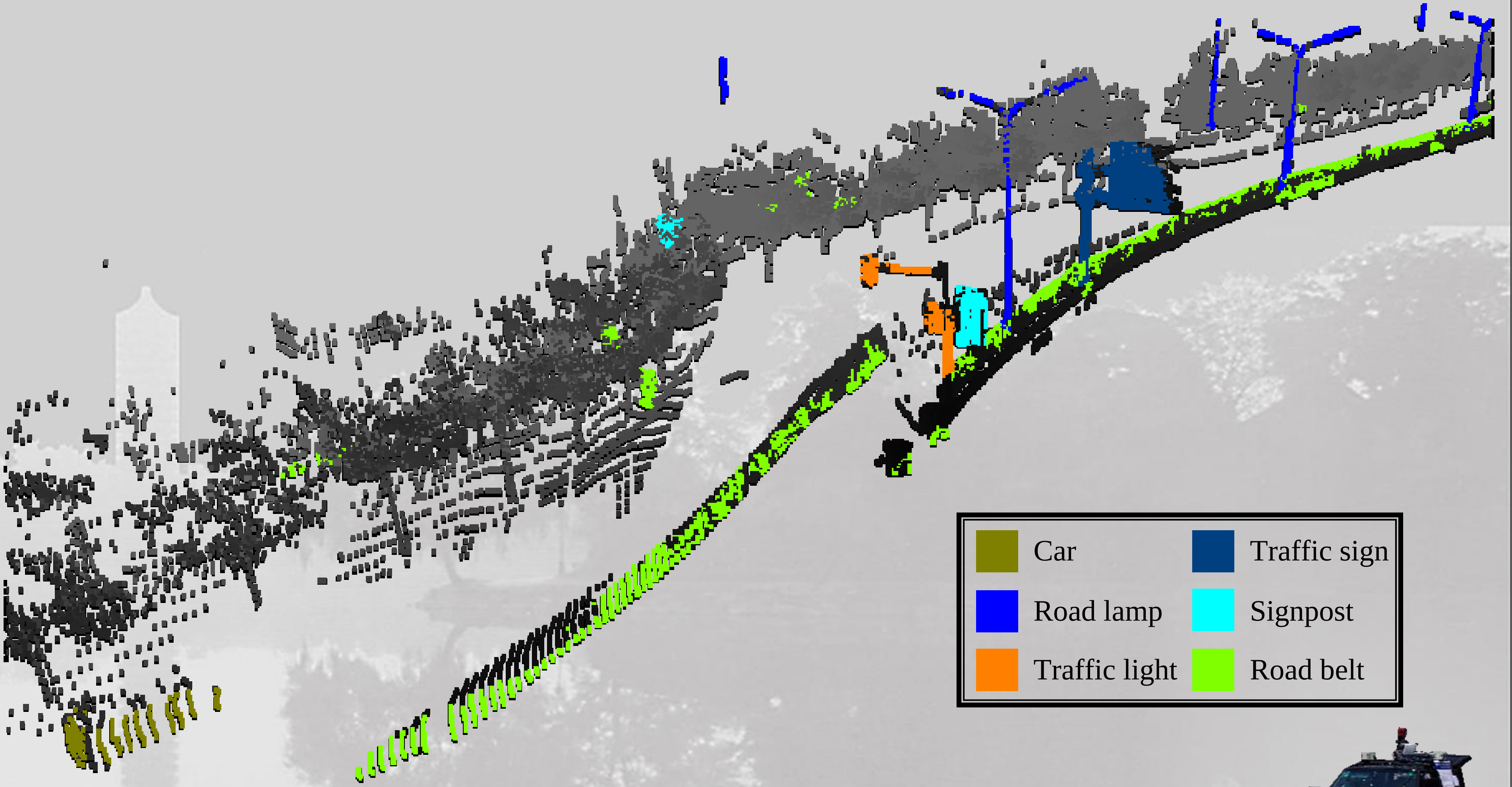
Experiment

- 1. Highway scene (the 4th ring road, Beijing)
  - Collecting time cost: 35 minutes
  - Data volume: about 14,300,000 laser points
  - Sample: 26 model graph for 8 object class
  - Processing time: 18 minutes (on a 2.8GHz & 8G PC)

class	Total	Highlighted	Correctly Highlighted	Precision	Recall
car	61	66	56	0.848	0.918
bus	27	22	20	0.909	0.741
traffic light	7	7	6	0.857	0.857
road lamp	210	196	190	0.969	0.904
signpost	13	18	11	0.611	0.846
traffic sign	62	71	56	0.789	0.903
building	53	43	40	0.930	0.754
road belt	33	33	25	0.758	0.758
all	466	456	404	0.886	0.867

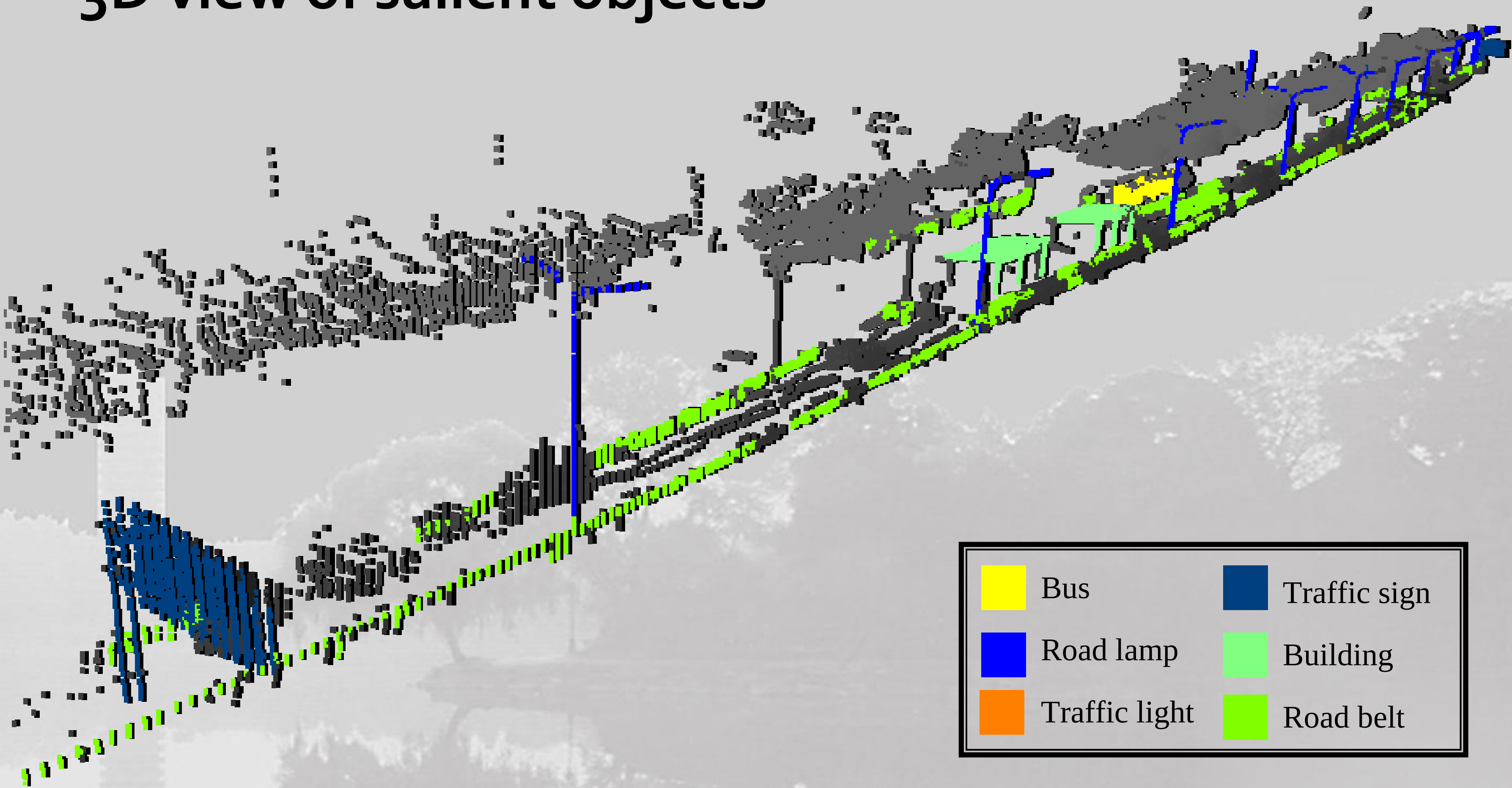
Experiment

- 3D view of salient objects



Experiment

- 3D view of salient objects

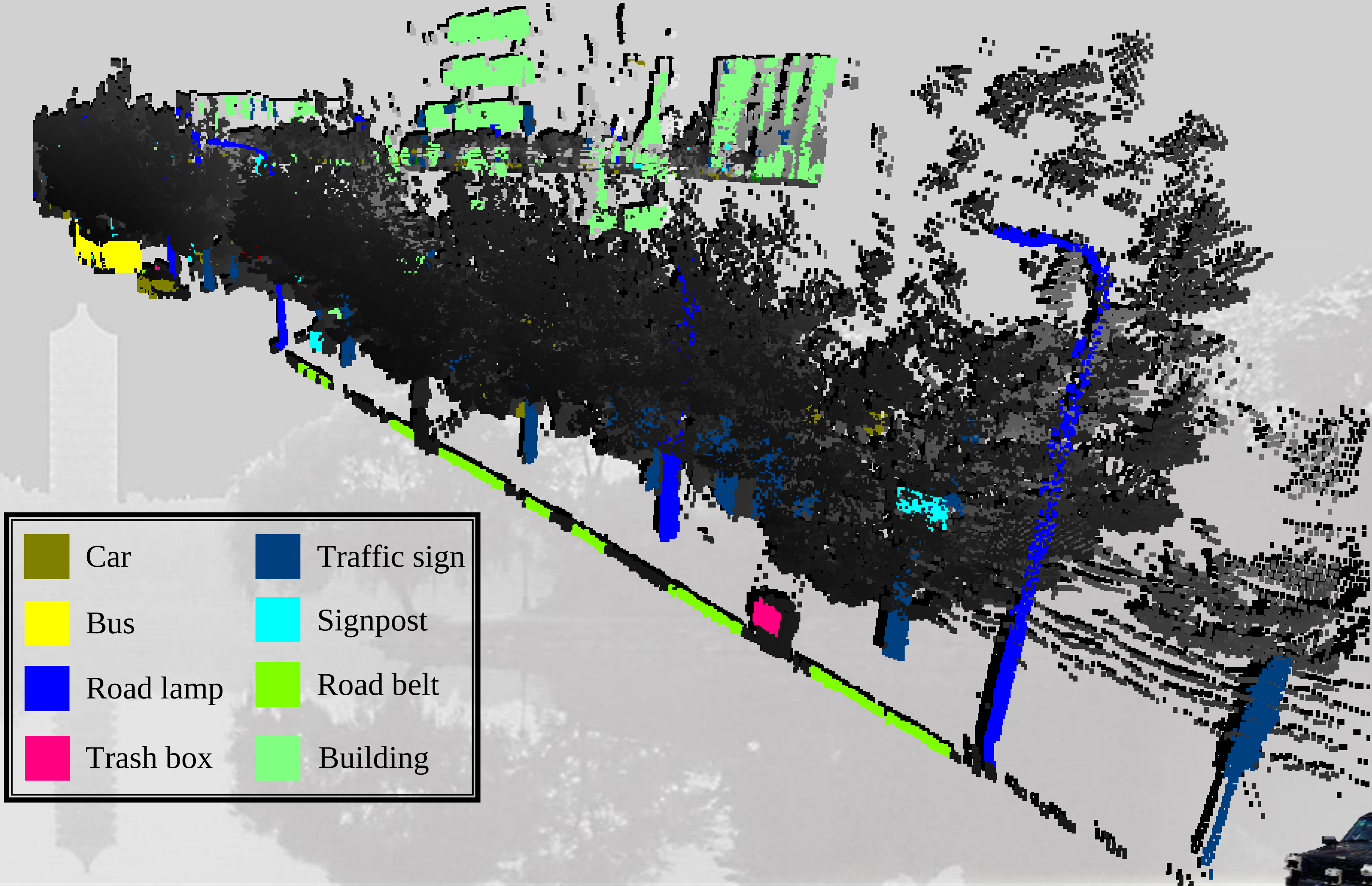


Experiment

- 2. Street scene (Street ShangDi, Beijing)
  - Collecting time cost: 30 minutes
  - Data volume: about 13,210,000 laser points
  - Sample: 38 model graph for 11 object class
  - Processing time: 20 minutes (on a 2.8GHz & 8G PC)

Experiment

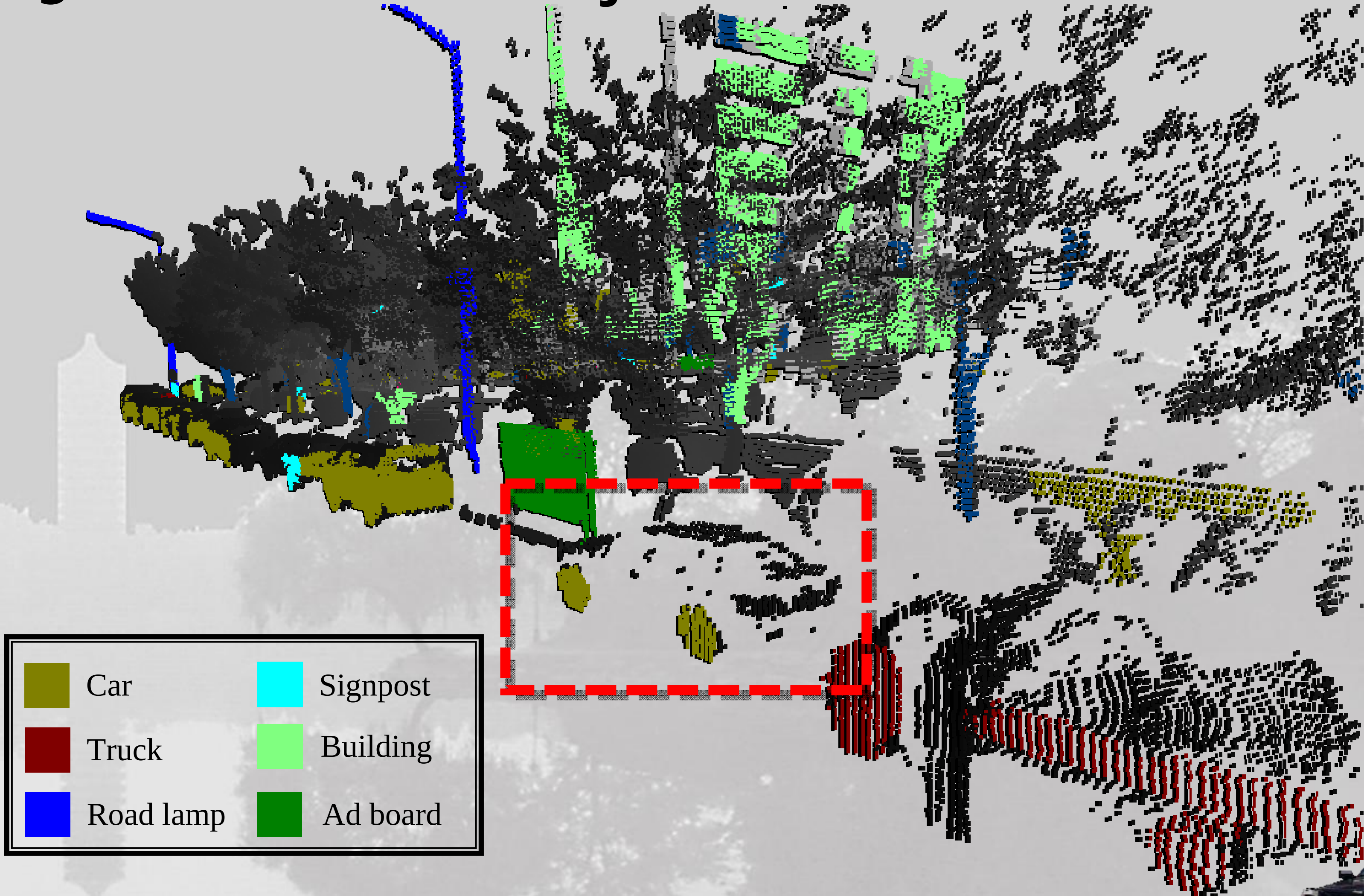
- 3D view of salient objects



Experiment

- **3D view of salient objects**

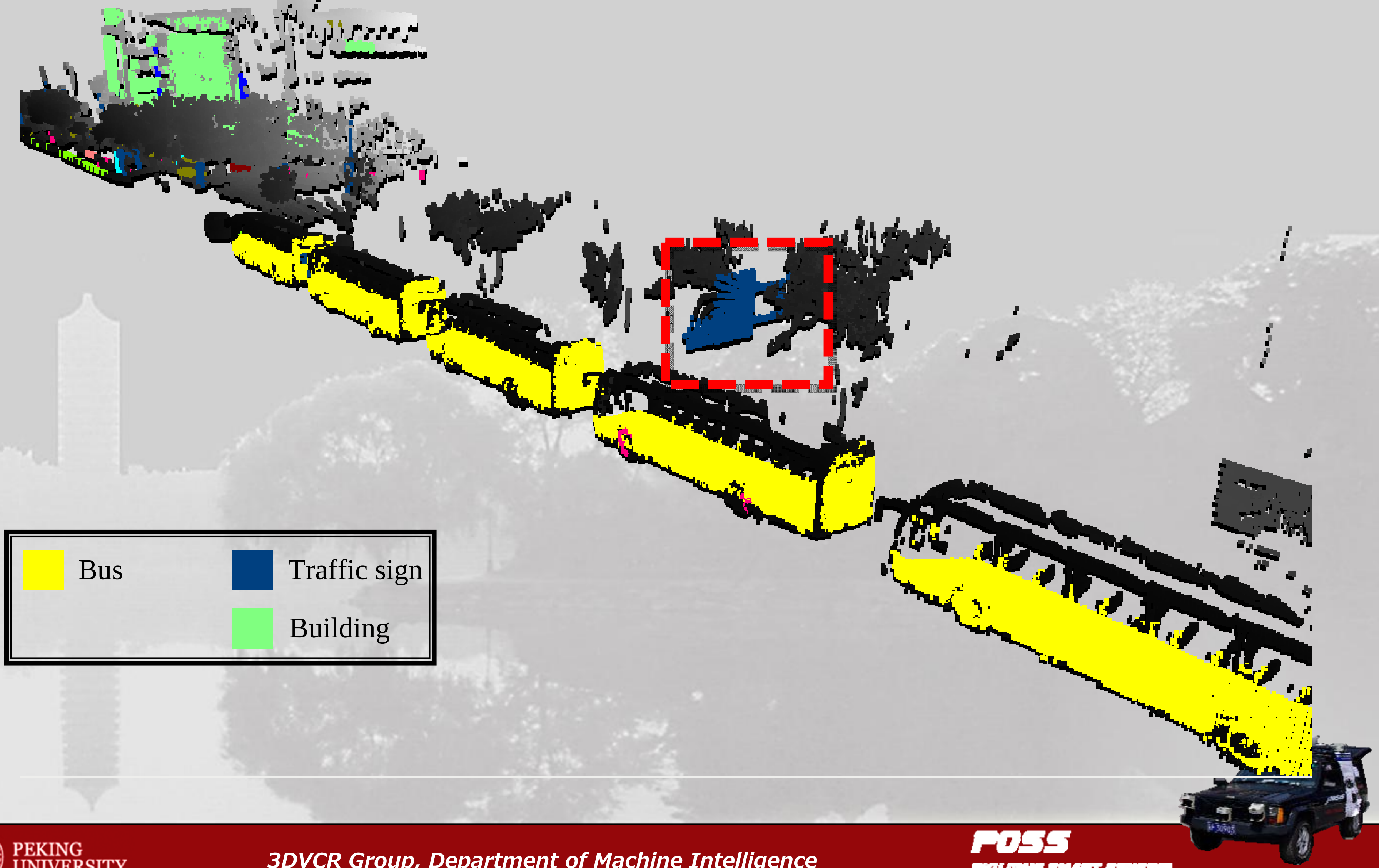
Add graph



Experiment

- 3D view of salient objects

Add graph



Summary

- An object-based saliency computing system that works on urban laser sensing data
  - We can label the regions of objects that belong to the salient object classes
  - The computation and time cost of the entire scene understanding system would reduce greatly since it only focus on those salient objects

- In the future
  - Context needs to be applied, as no high-level knowledge is introduced currently
  - Combine with advanced detection & classification methods

Zhao, H., et al., Detection and Tracking of Moving Objects at Intersections using a Network of Laser Scanners, IEEE Trans. ITS, vol. 13, no.2, 655-670, 2012.

MONITORING AN INTERSECTION USING A NETWORK OF LASER SCANNERS

H. Zhao, J. Cui, H. Zha,

Peking University

K. Katabira, X. Shao, R. Shibasaki,

University of Tokyo

zhaohj@cis.pku.edu.cn

Background (1)

Analyzing and Monitoring the traffic behavior in an intersection

- Efficiently and accurately Collecting the Traffic Data in an Intersection
- Real-timely Detecting Dangeous Situations.

Background (2)

- Vision-based methods suffer mainly on the following difficulties
  - Occlusion
  - Computation Cost
  - Illumination Change

e.g. the camera is required to be set on a tall position, monitoring intersection from the above

<u>To solve the problems</u>

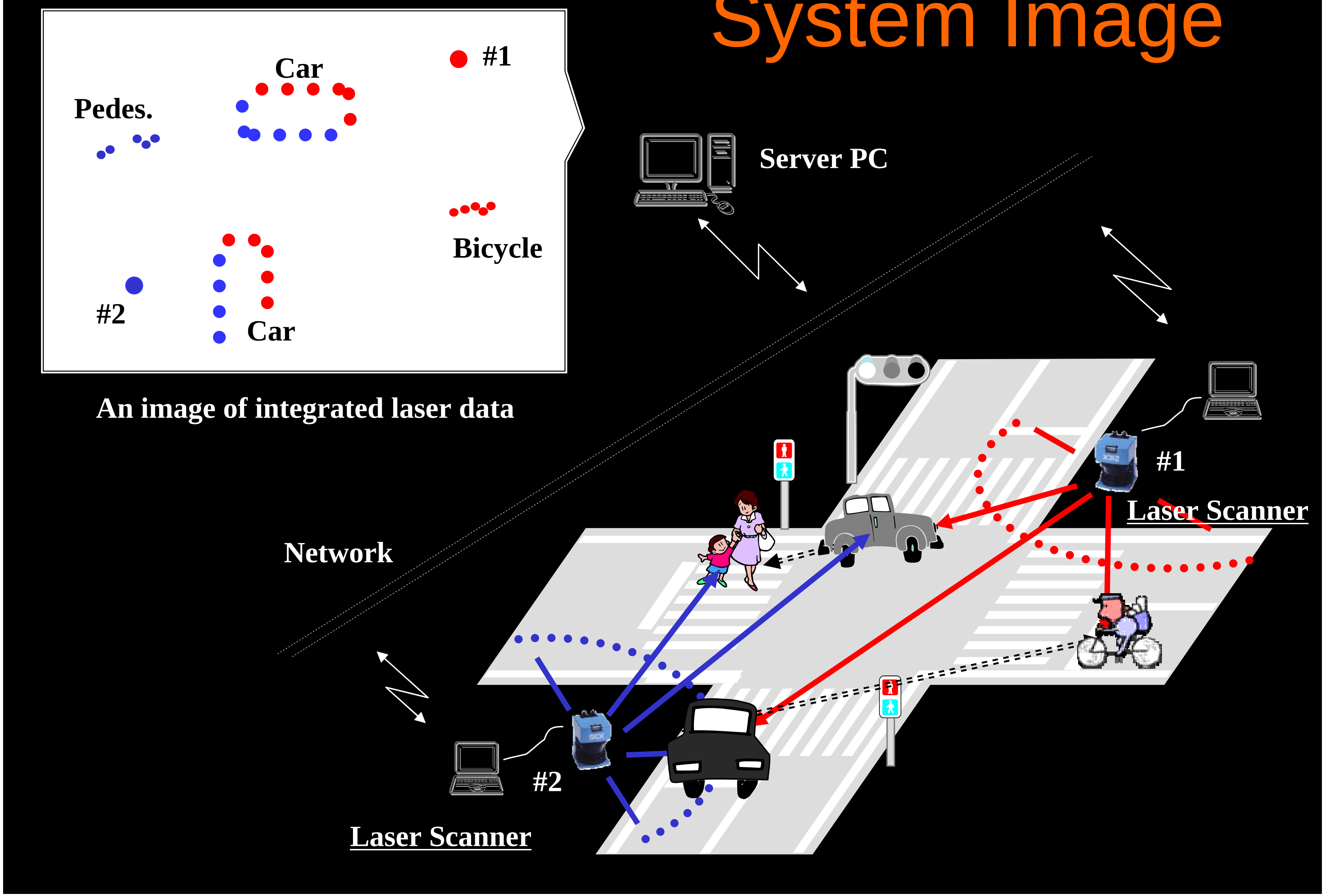
1. Restrict camera's setting condit
2. Target on a simplified situation

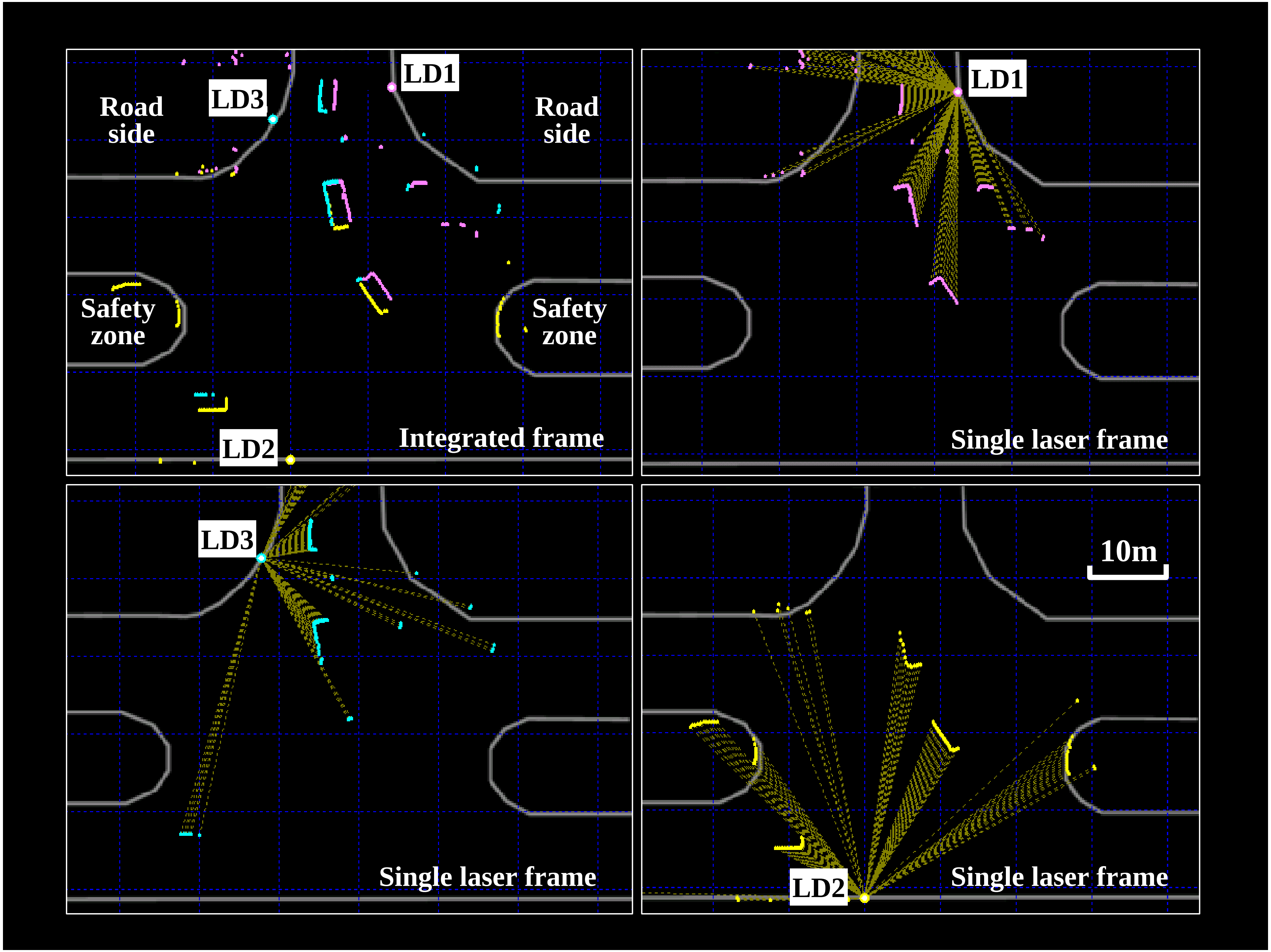
e.g. monitor vehicles of limited lanes, do not discriminate moving objects.

Objective

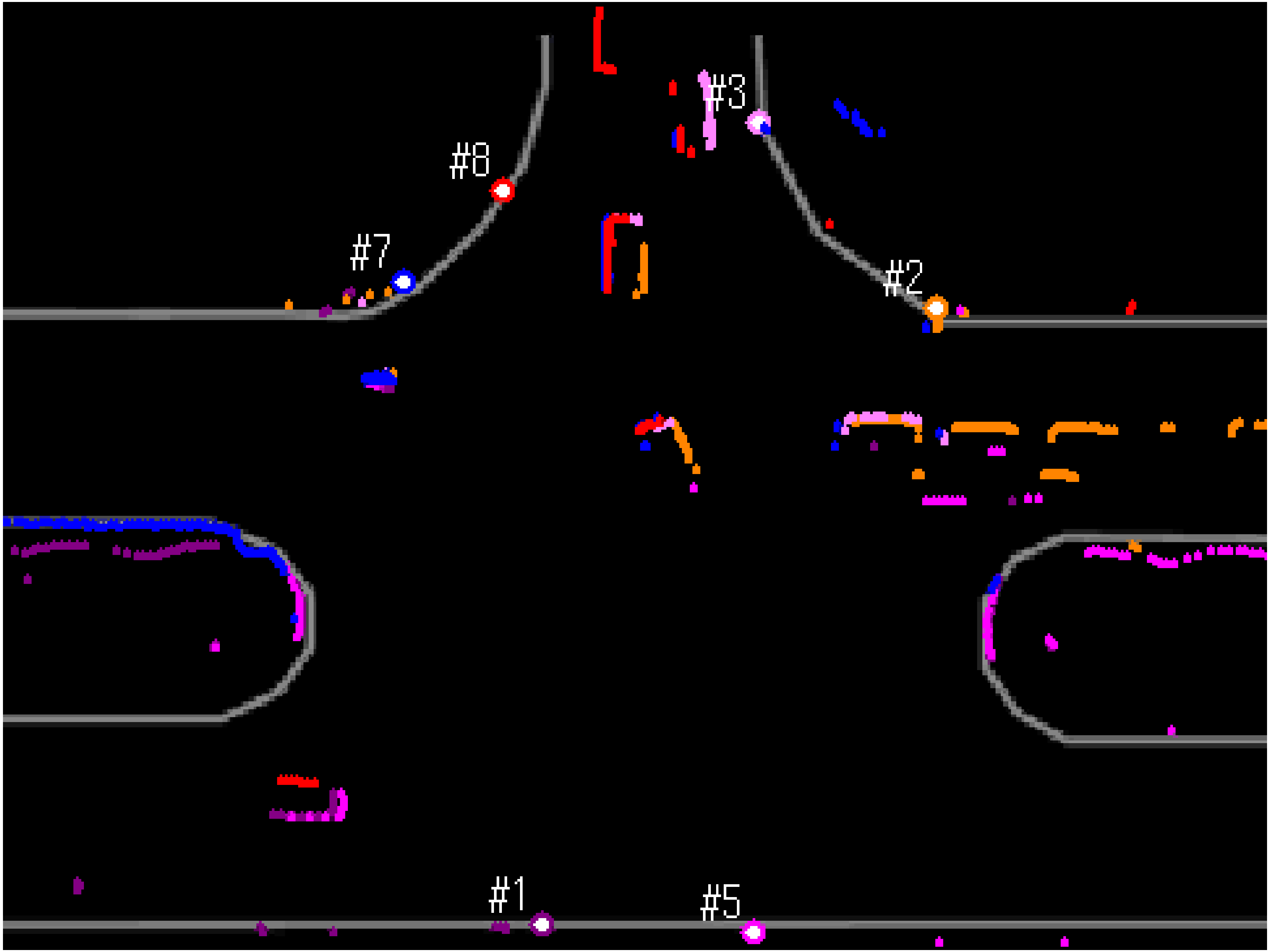
This research propose a novel system for monitoring and collecting **detailed traffic data**, with **easy setting condition**, in an environment of complicated traffic behavior, such as **intersection**, using **a network of single-row laser scanner.**

System Image



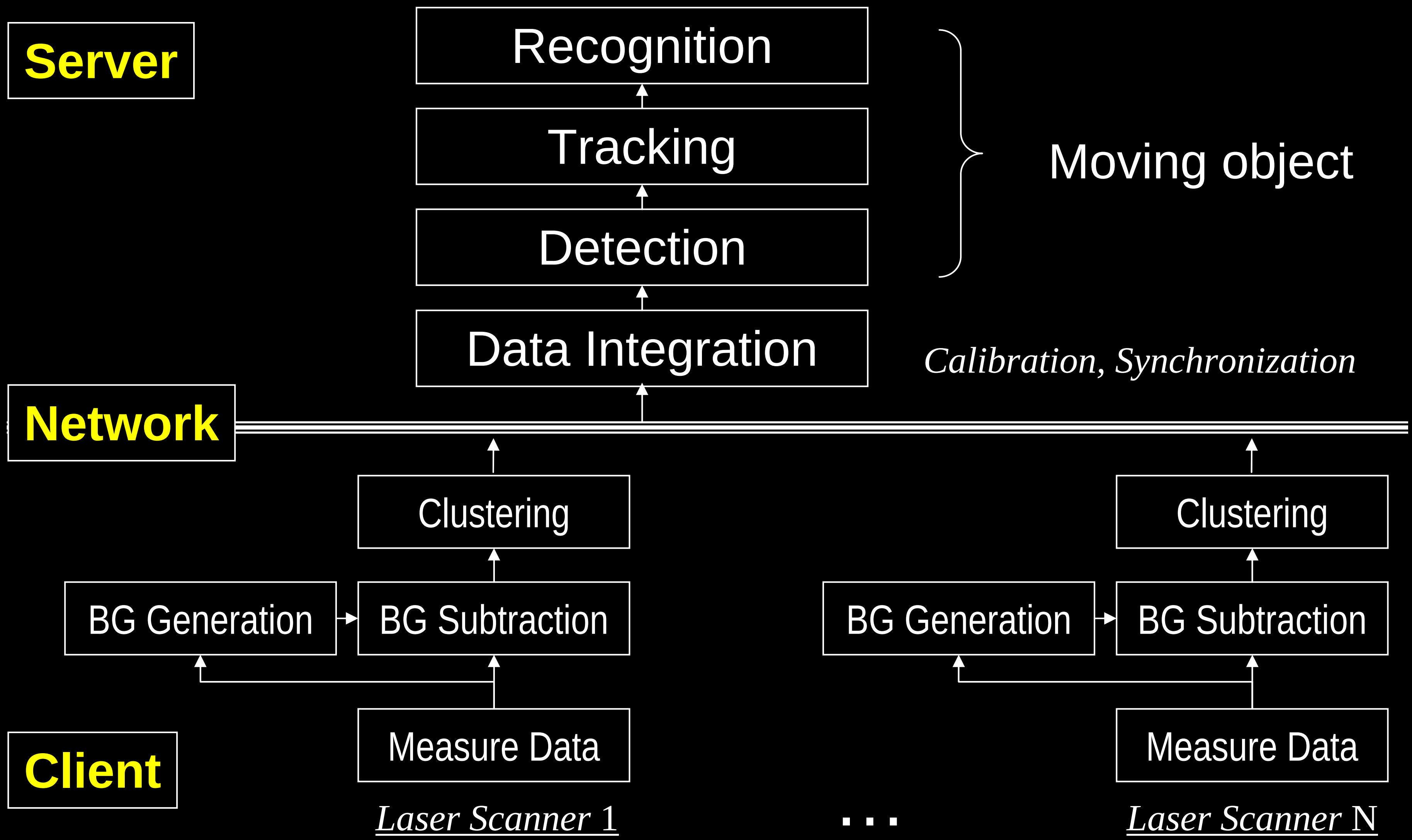


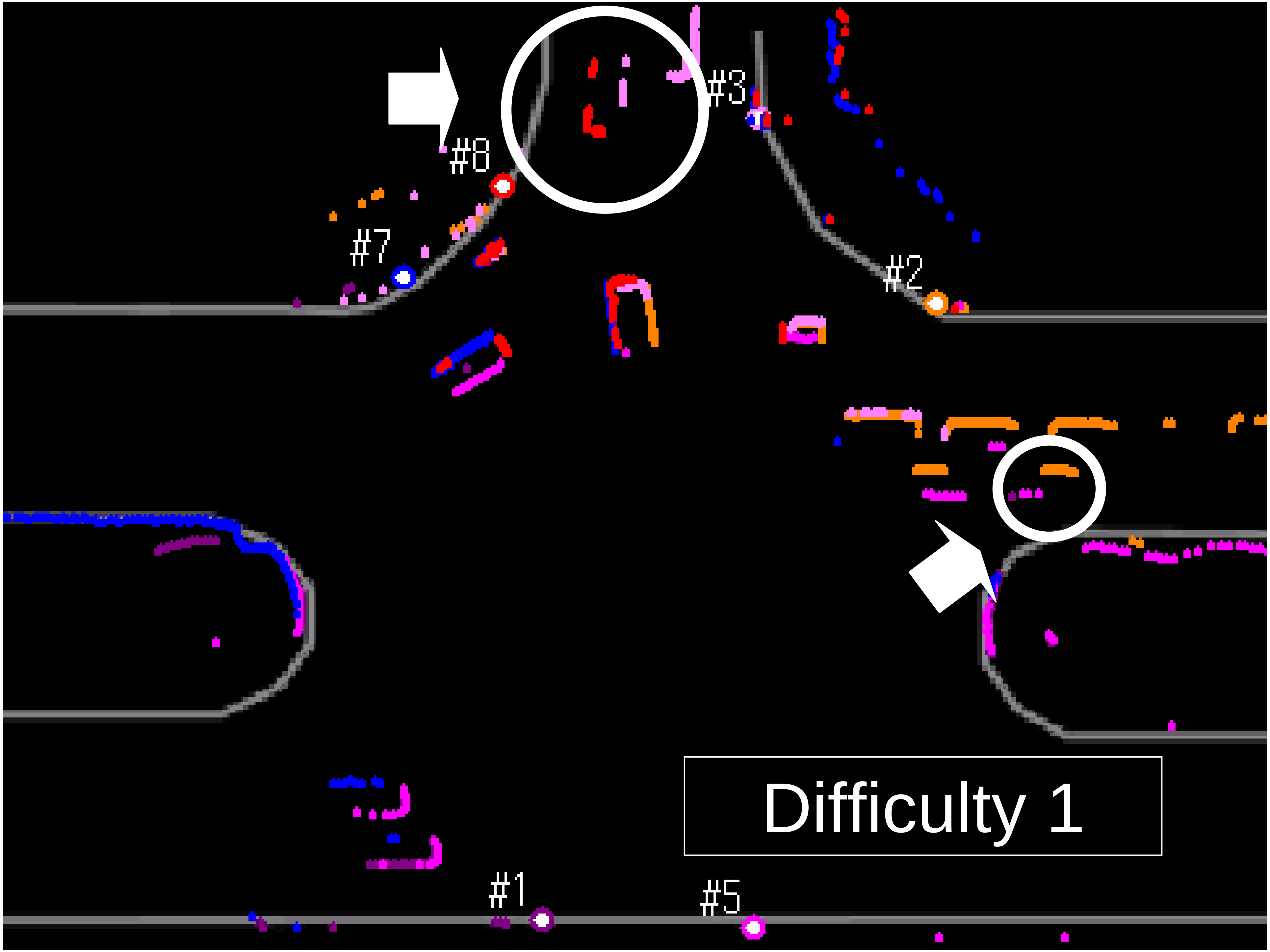
LD1
LD3
Road side
Road side
Safety zone
Safety zone
LD2
Integrated frame
LD1
Single laser frame
LD3
Single laser frame
10m
LD2
Single laser frame

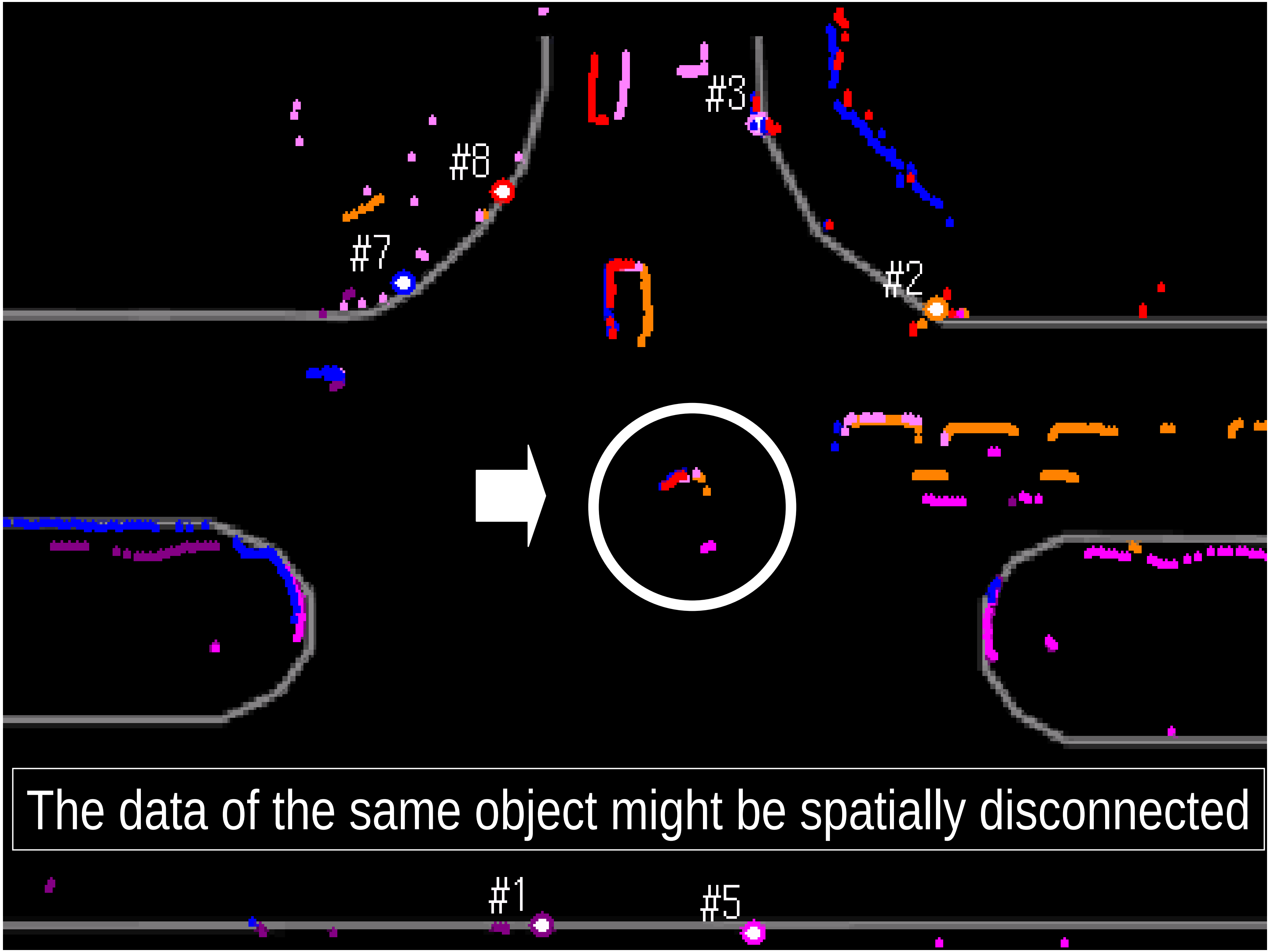


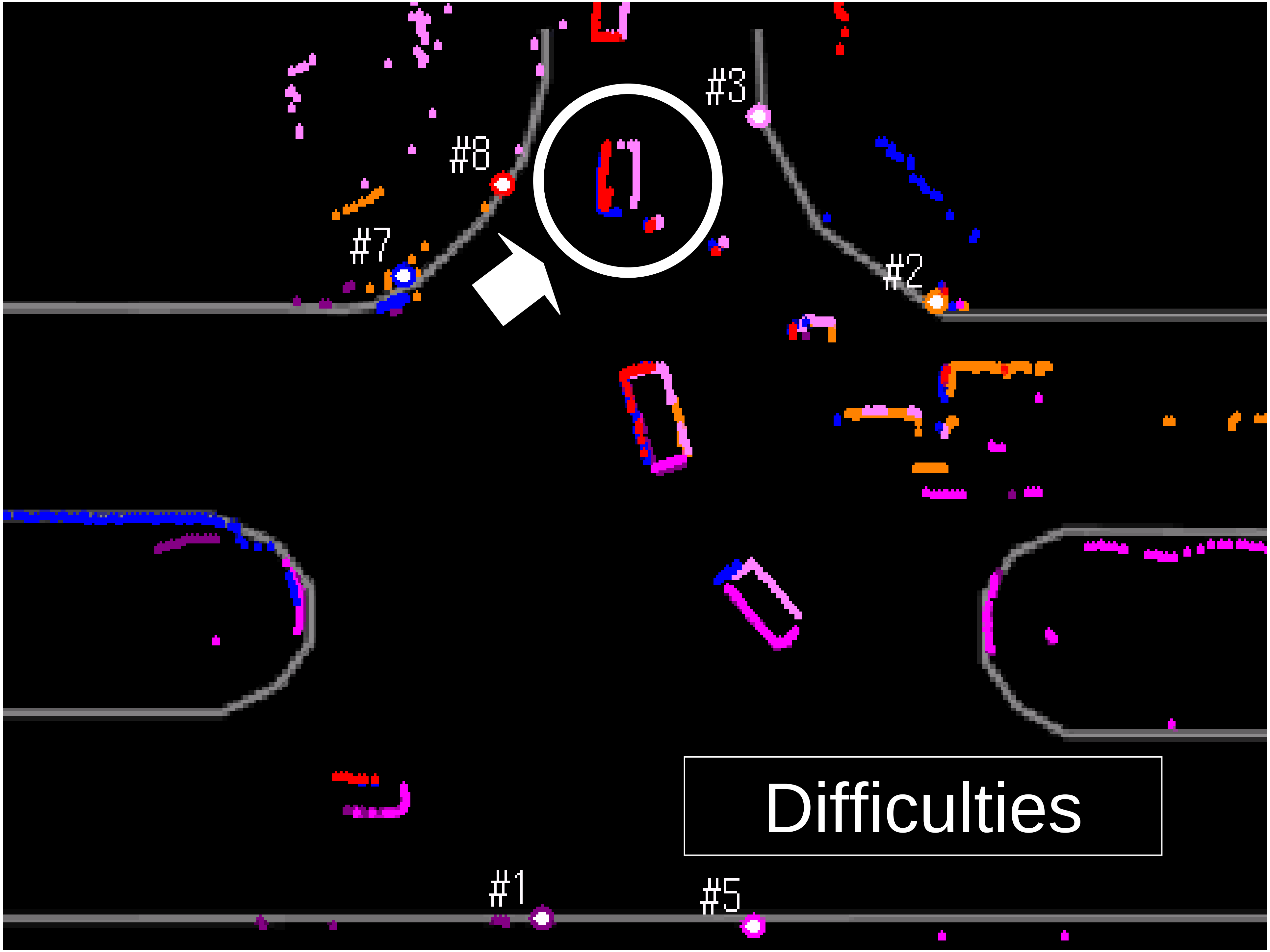
#8
#3
#7
#2
#1
#5

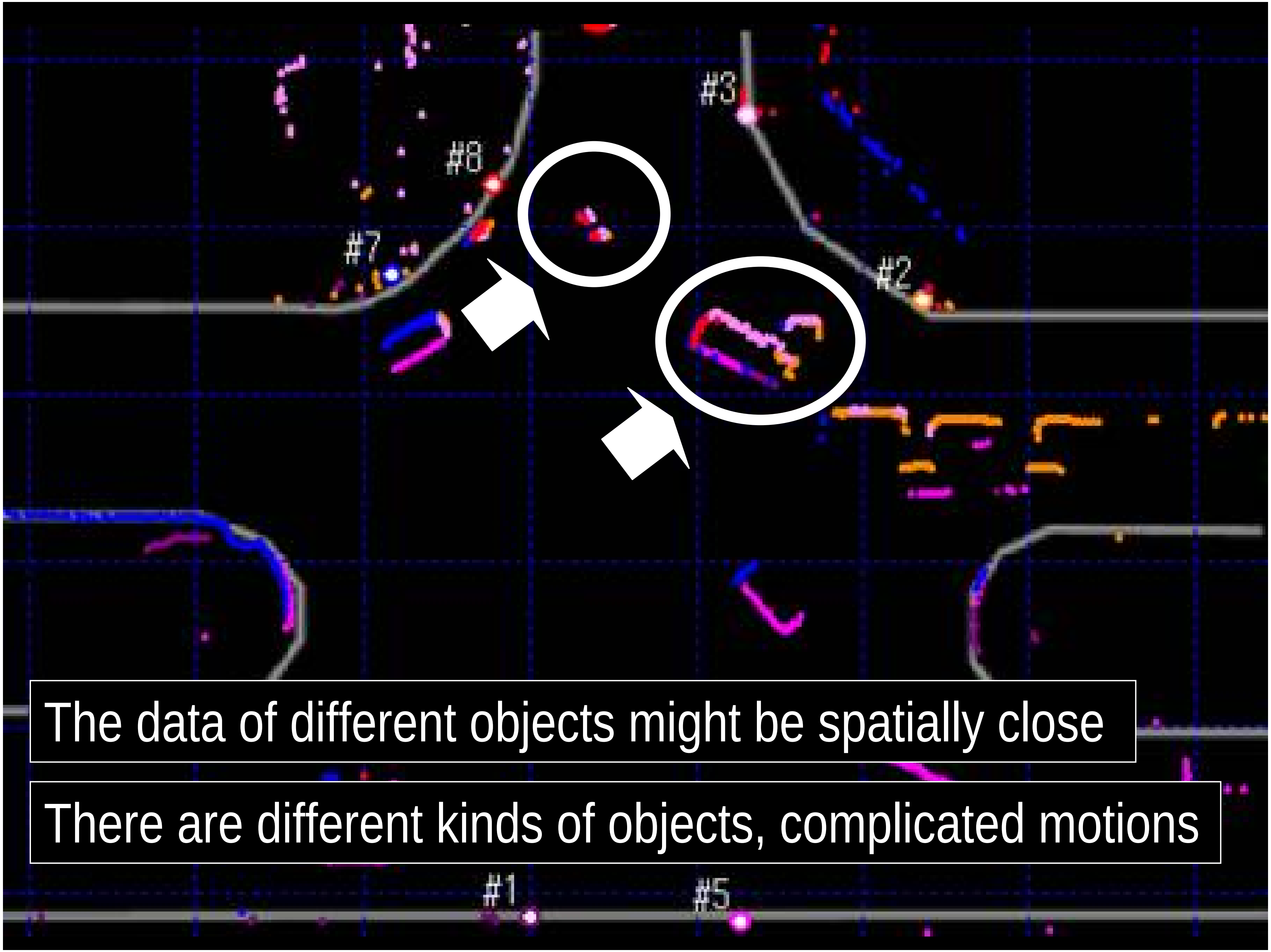
Processing Modules



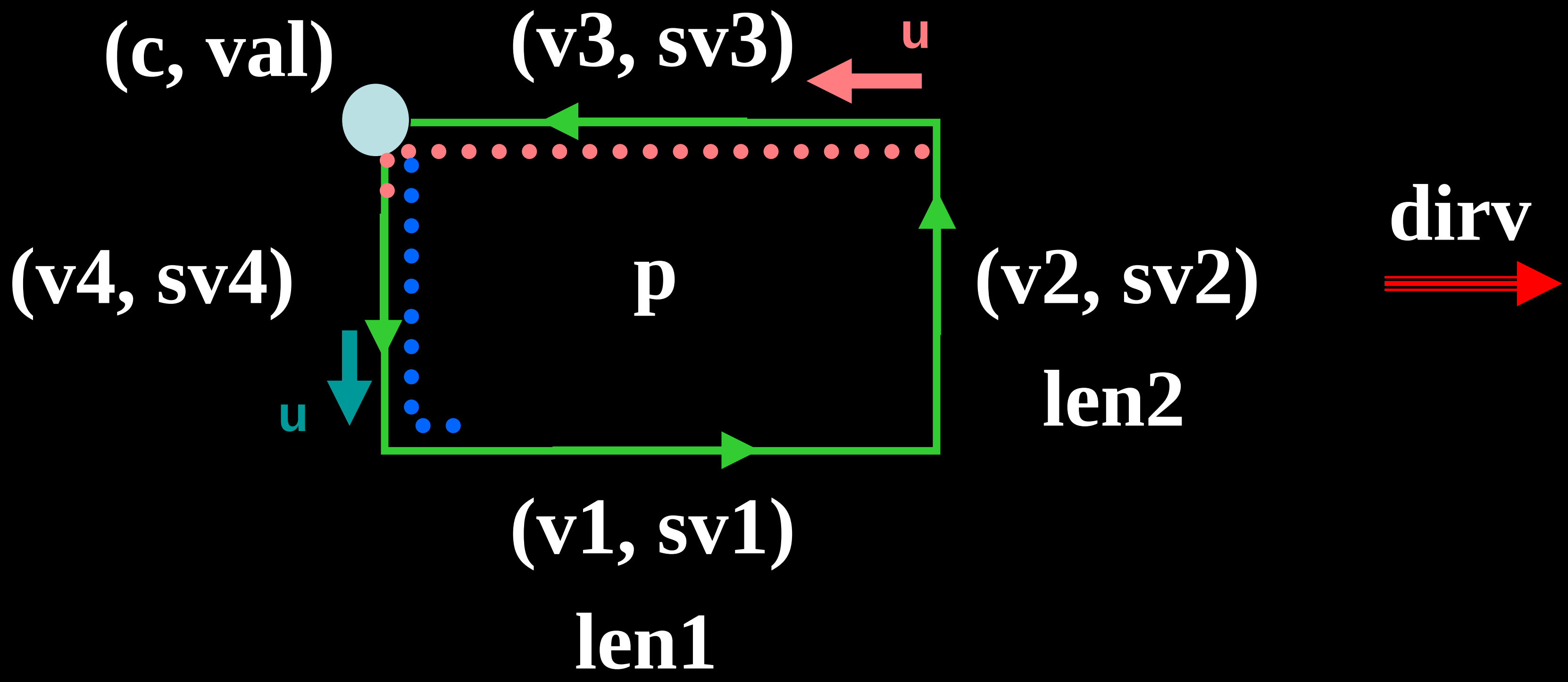

#8
#3
#7
#2
#1
#5
Difficulty 1


#3
#8
#7
#2
The data of the same object might be spatially disconnected
#1
#5


#3
#8
#7
#2
#1
#5
Difficulties


#3
#8
#7
#2
The data of different objects might be spatially close
There are different kinds of objects, complicated motions
#1
#5

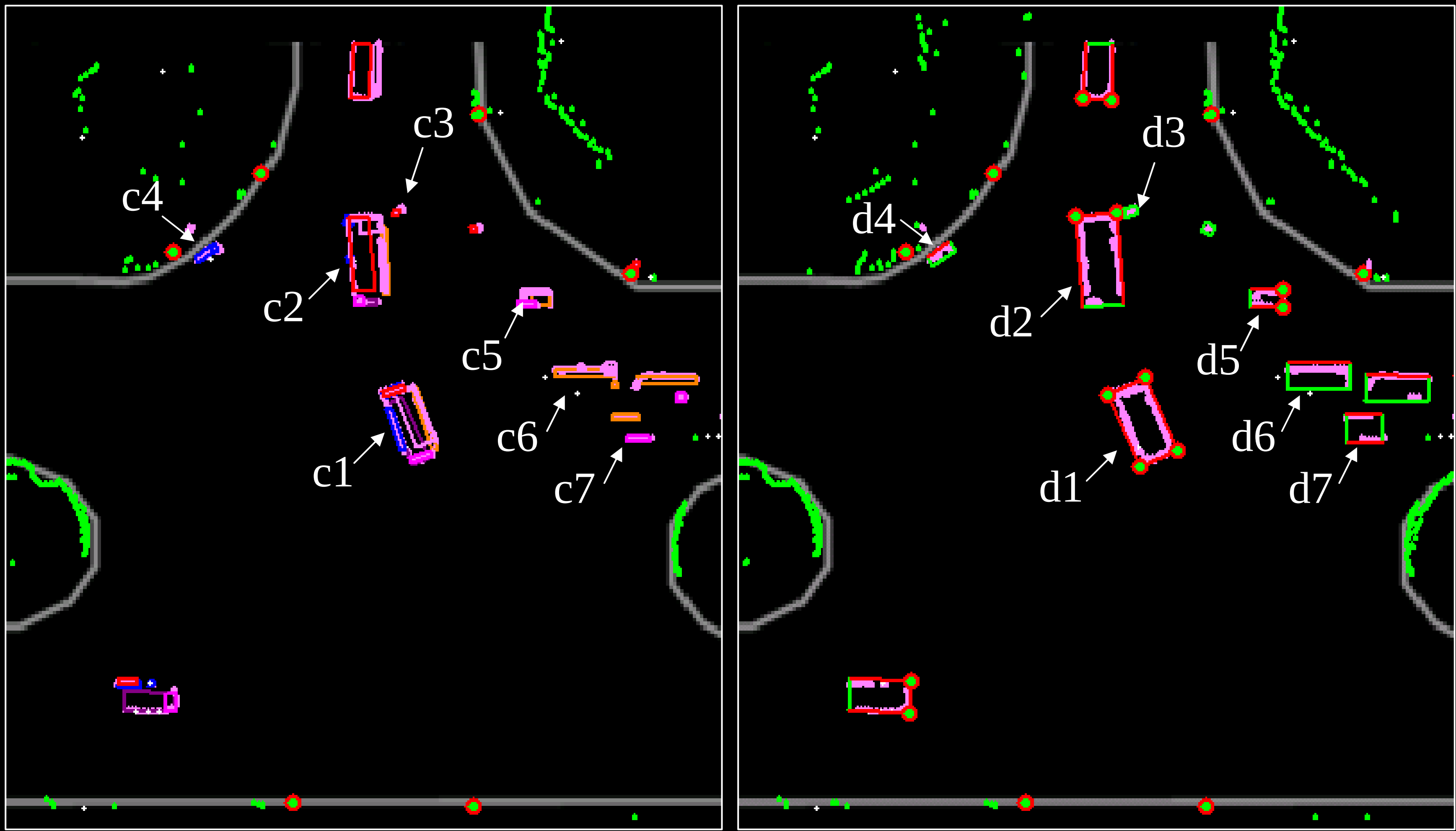
Object Model



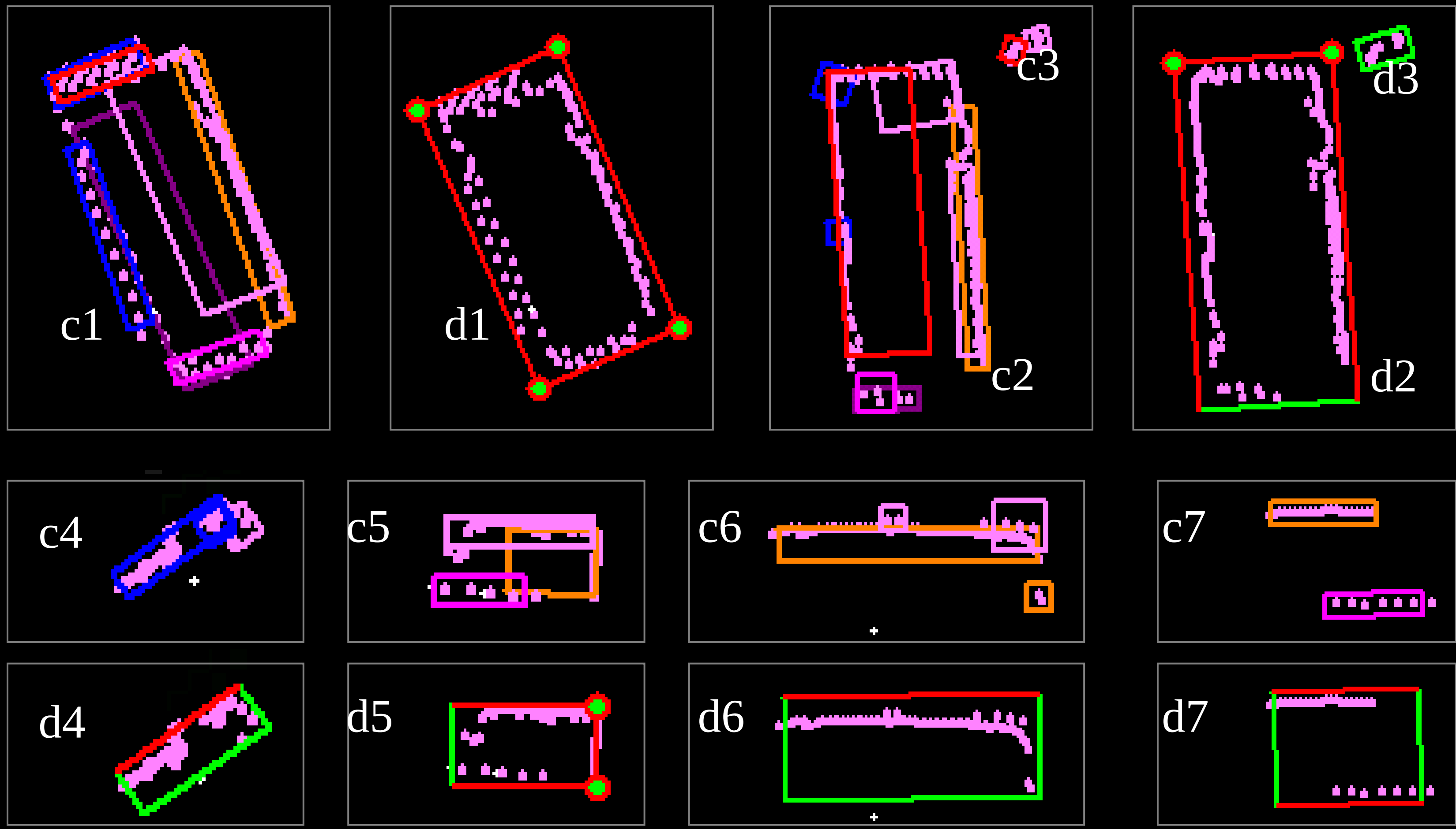
Feature parameters
and their reliabilities

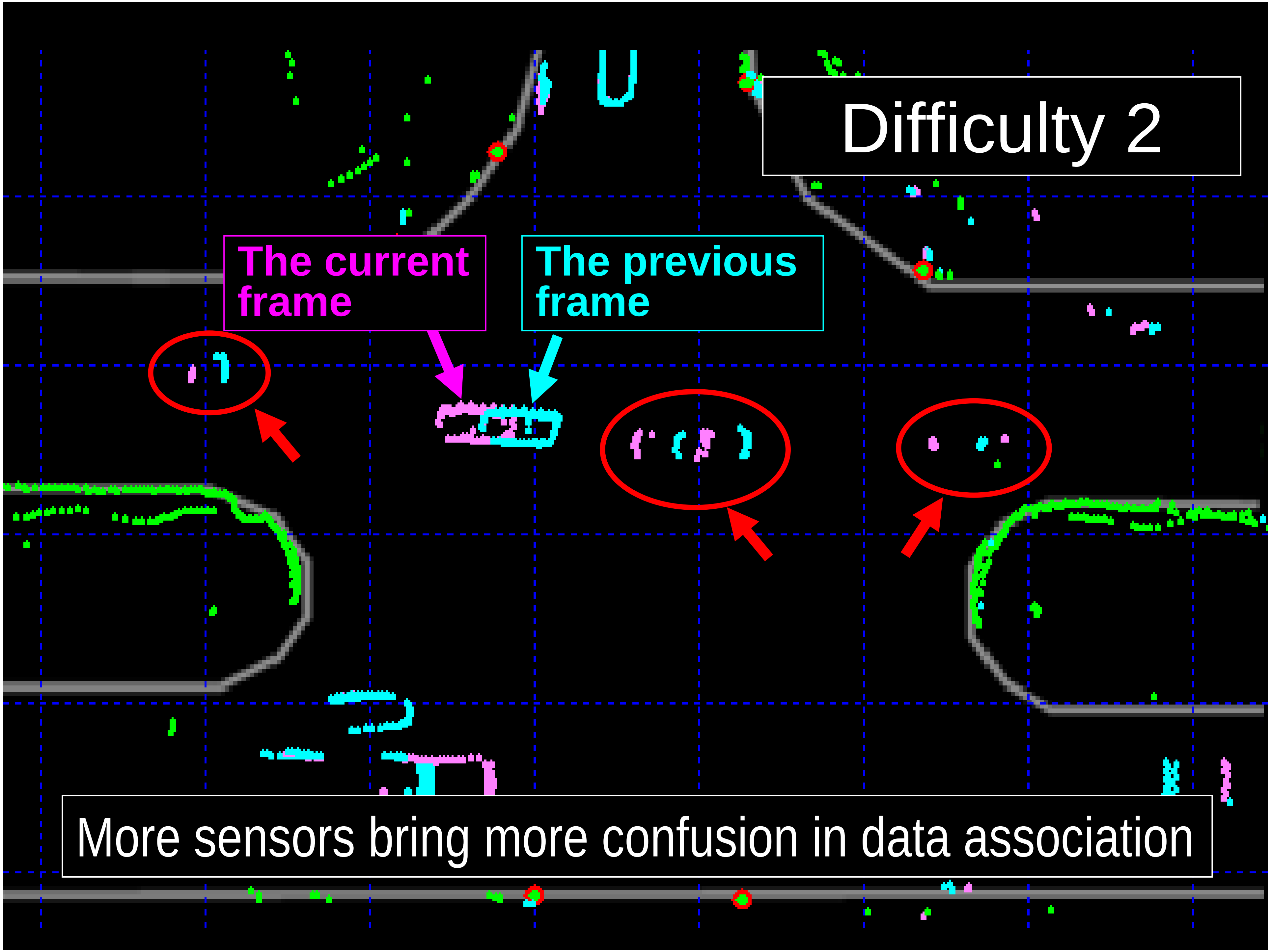
(v1=dirv)

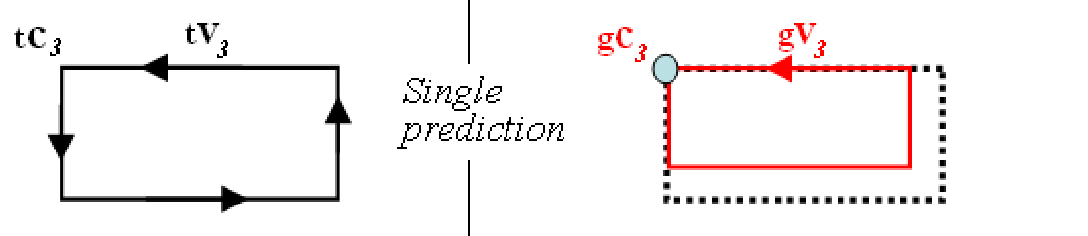
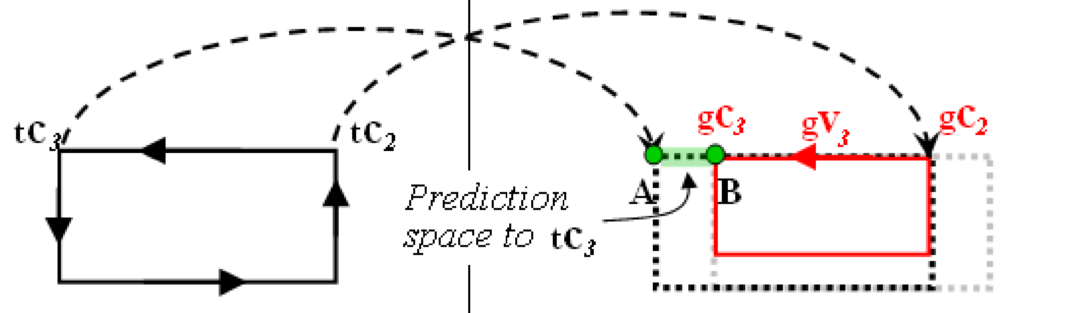
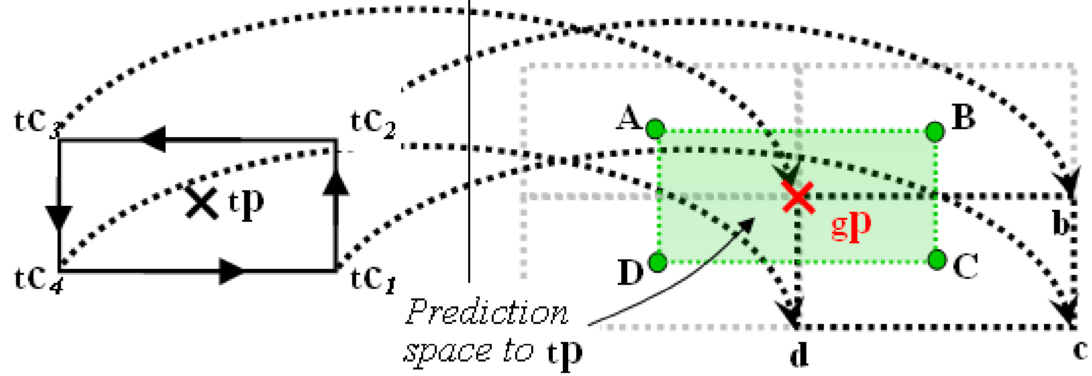
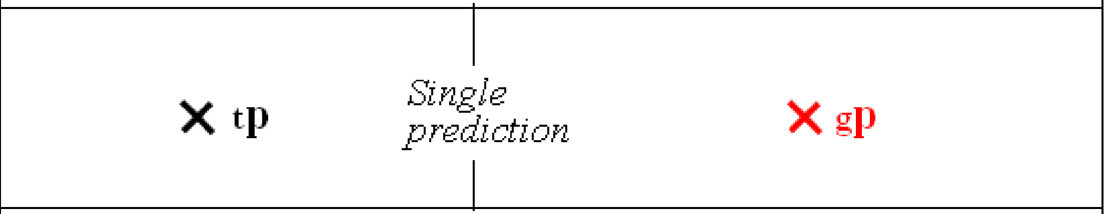
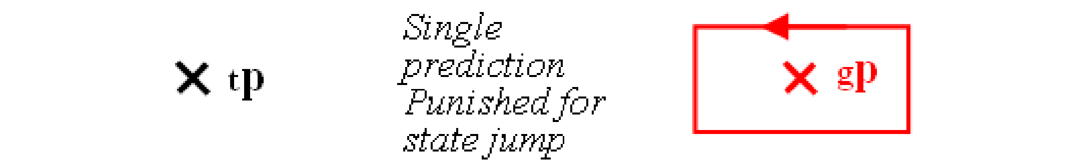
Object Detection Results



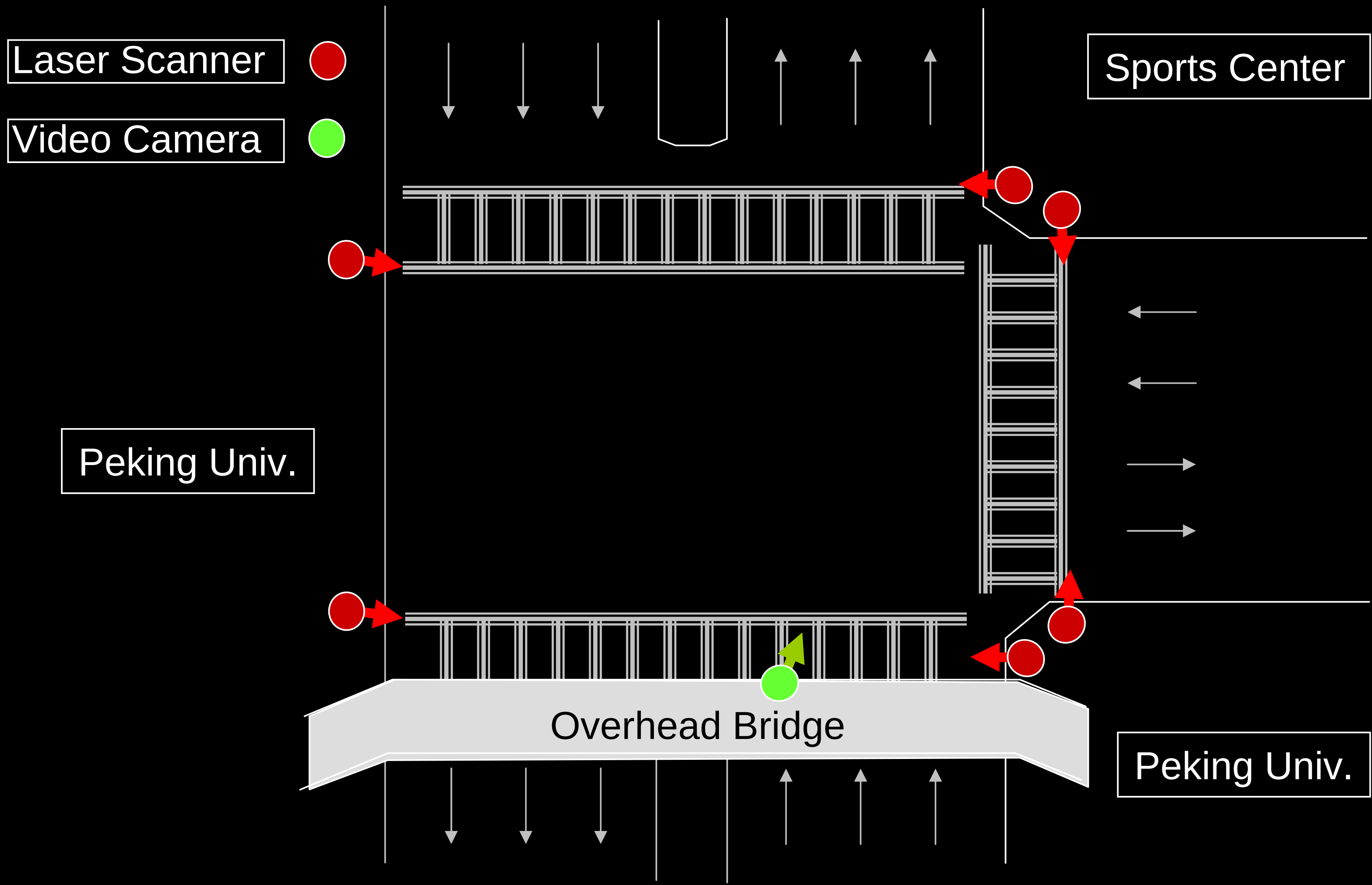
Object Detection Results



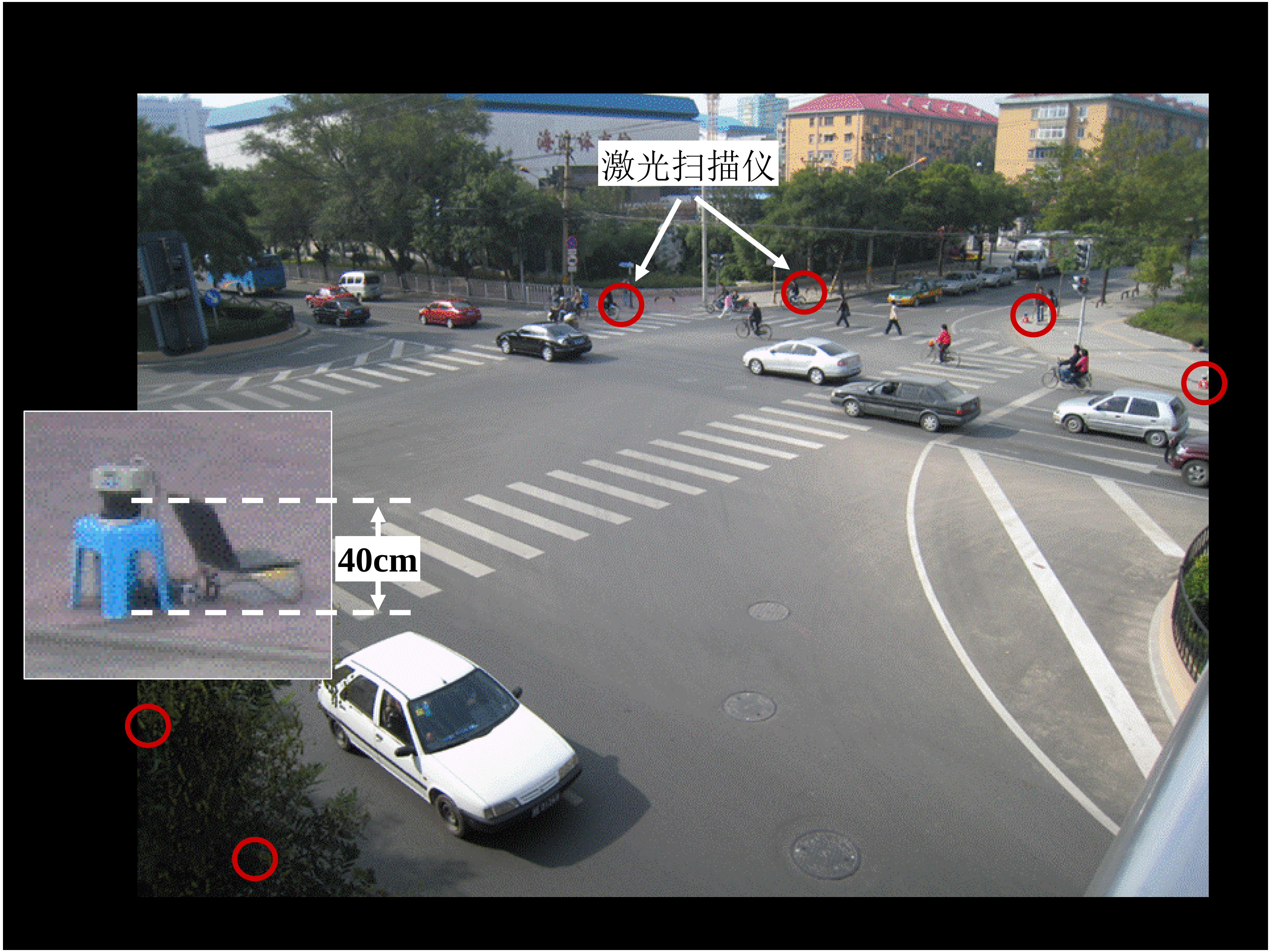

Difficulty 2
The current frame
The previous frame
More sensors bring more confusion in data association

	State t^{k-1} / Observation g^k
<u>Case 1</u> t^{k-1} *has support vectors* g^k *has support vectors, and valid corner points*	
<u>Case 2</u> t^{k-1} *has support vectors* g^k *has support vectors, but **no** valid corner point*	
<u>Case 3</u> t^{k-1} *has support vectors* g^k *has **no** support vector, **nor** valid corner point*	
<u>Case 4</u> t^{k-1} *has **no** support vector* g^k *has **no** support vector*	
Case 5 t^{k-1} *has **no** support vector* g^k *has support vectors*	

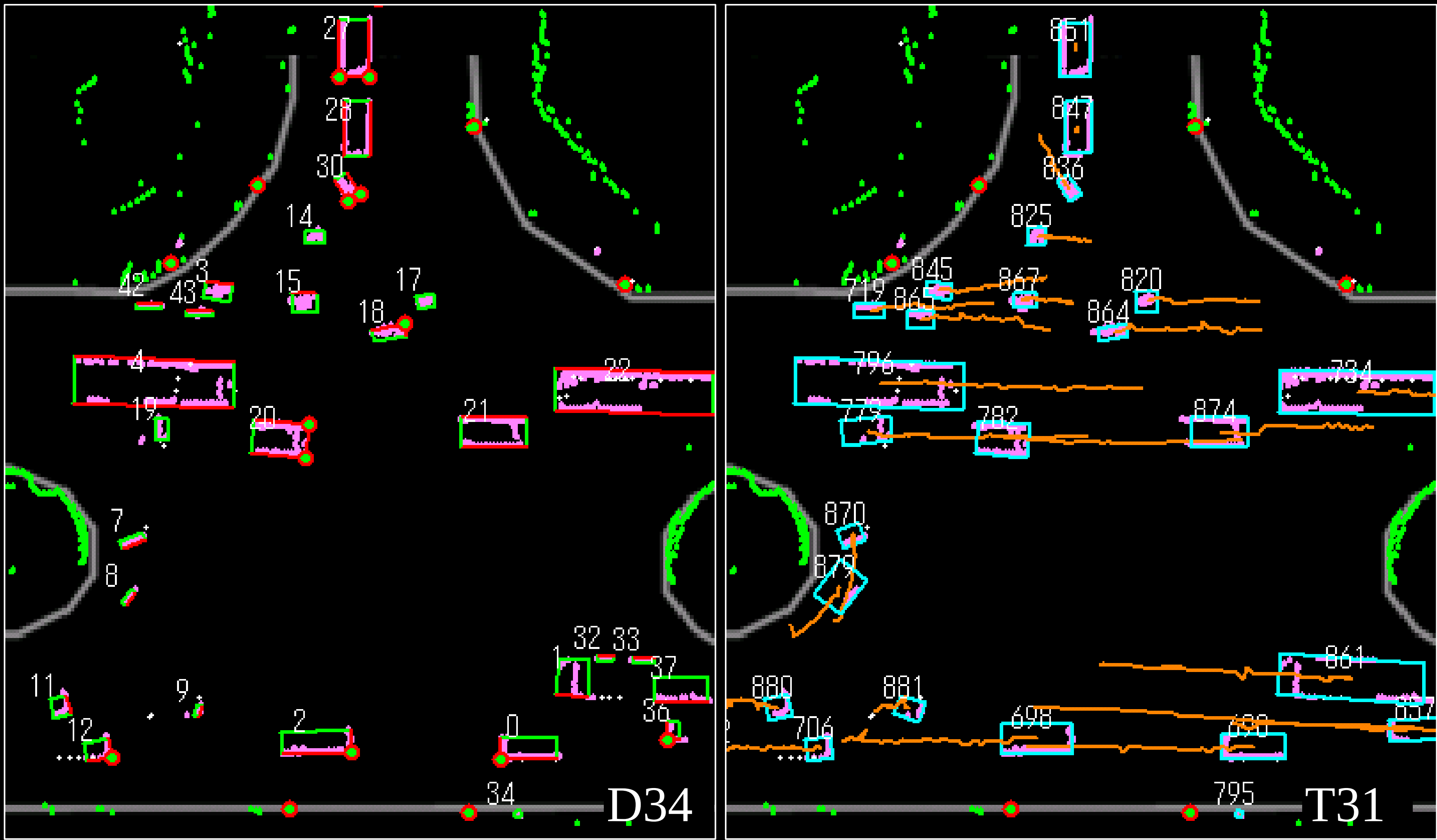
Experiment



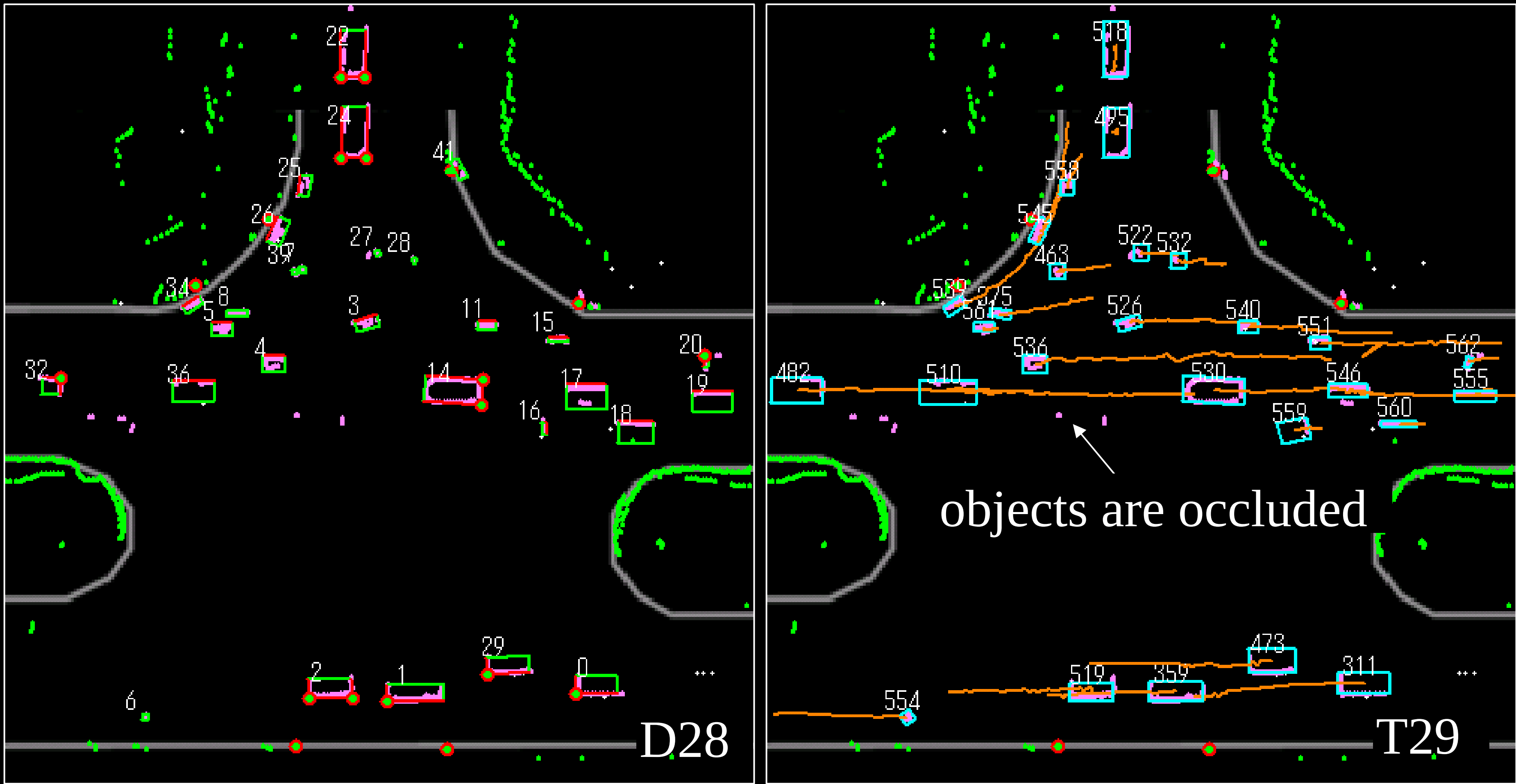
Laser Scanner
Video Camera
Sports Center
Peking Univ.
Overhead Bridge
Peking Univ.


激光扫描仪
40cm

Results



Results



Accuracy Analysis

2007.10.13, 10:00-10:20

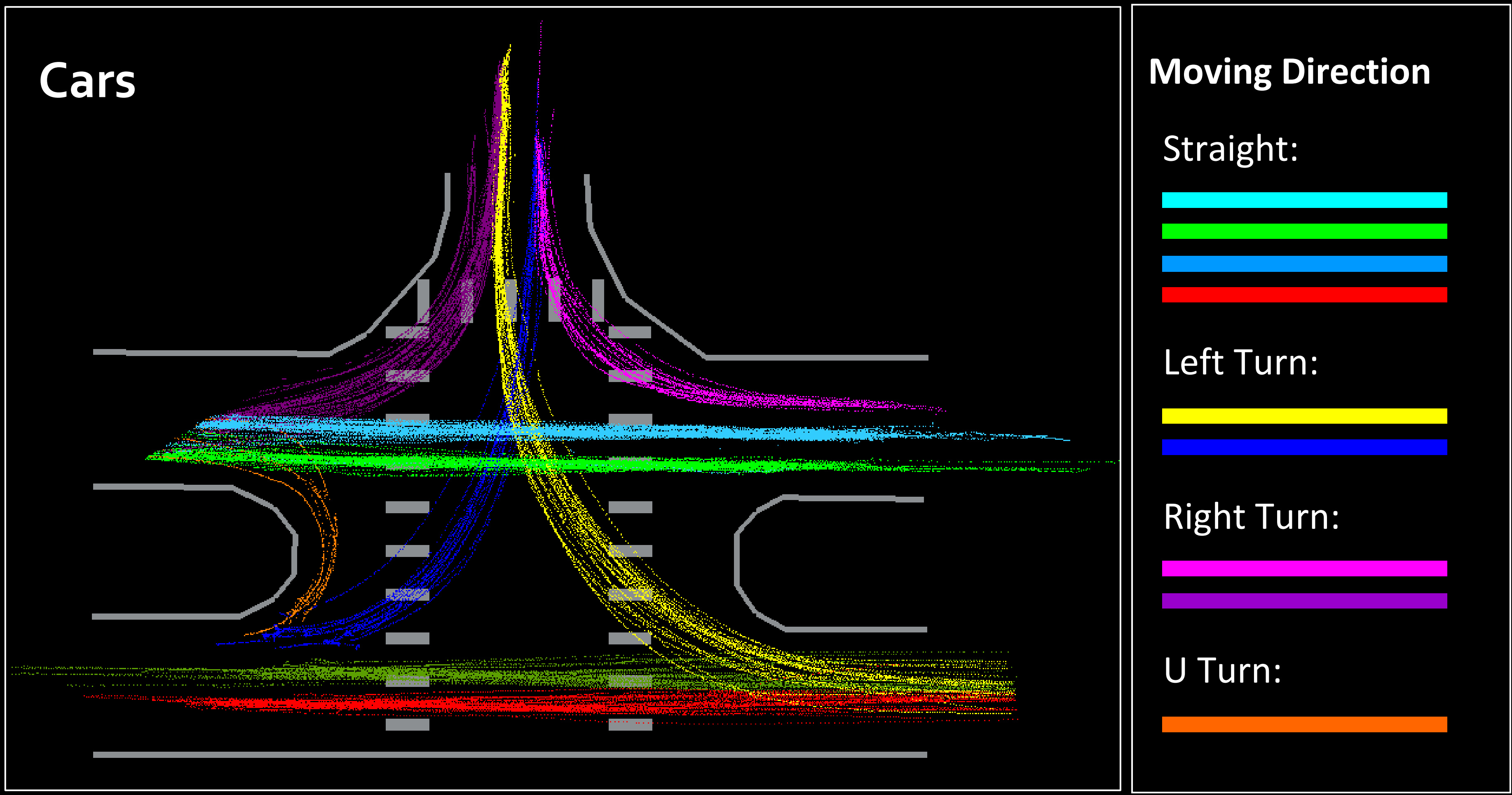
Detection Results

type	perfect	split	merge	none	total	d.ratio	p.ratio
car	6915	614	7	89	7625	0.988	0.907
bicycle	1571	82	0	24	1677	0.986	0.938
pedes.	799	13	508	130	1450	0.910	0.551
sum.	9285	709	515	243	10752	0.977	0.864

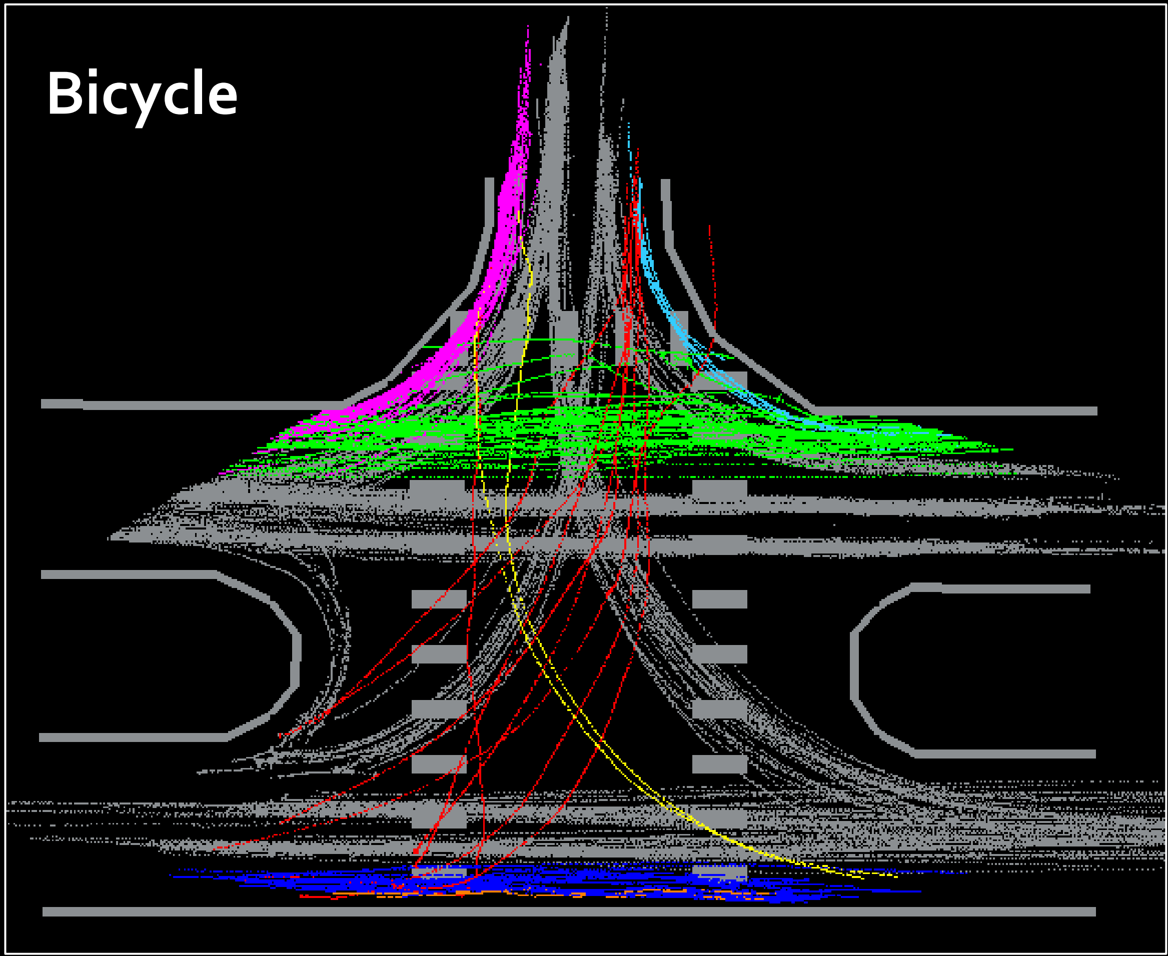
Tracking Results

type	perfect	broken	error	total	t.ratio	p.ratio
car	636	22	17	675	0.975	0.942
bicycle	322	8	21	351	0.940	0.917
pedes.	30	2	5	37	0.865	0.811
sum.	988	32	43	1063	0.960	0.929

Trajectory Clustering



Trajectory Clustering



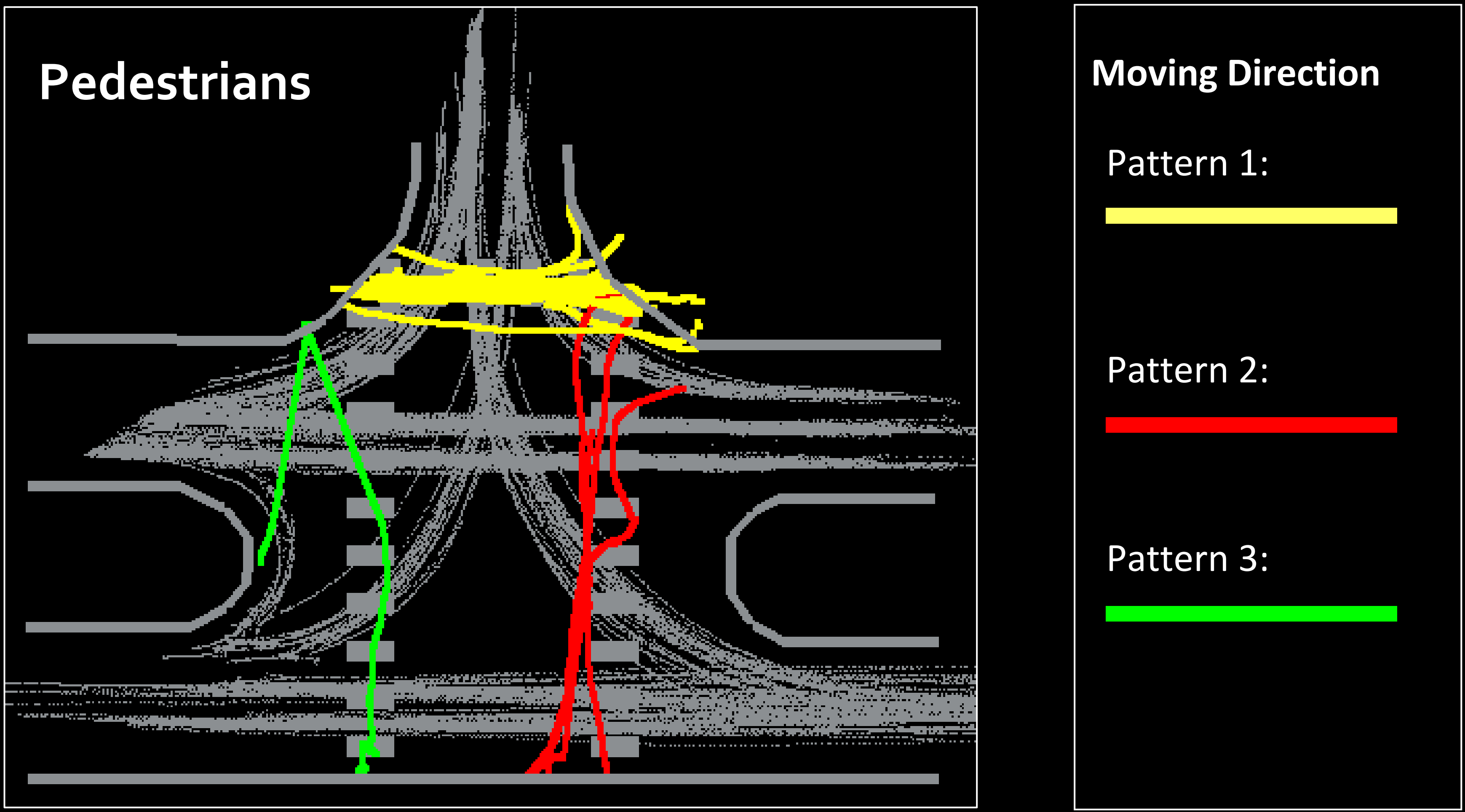
Moving Direction

Straight:

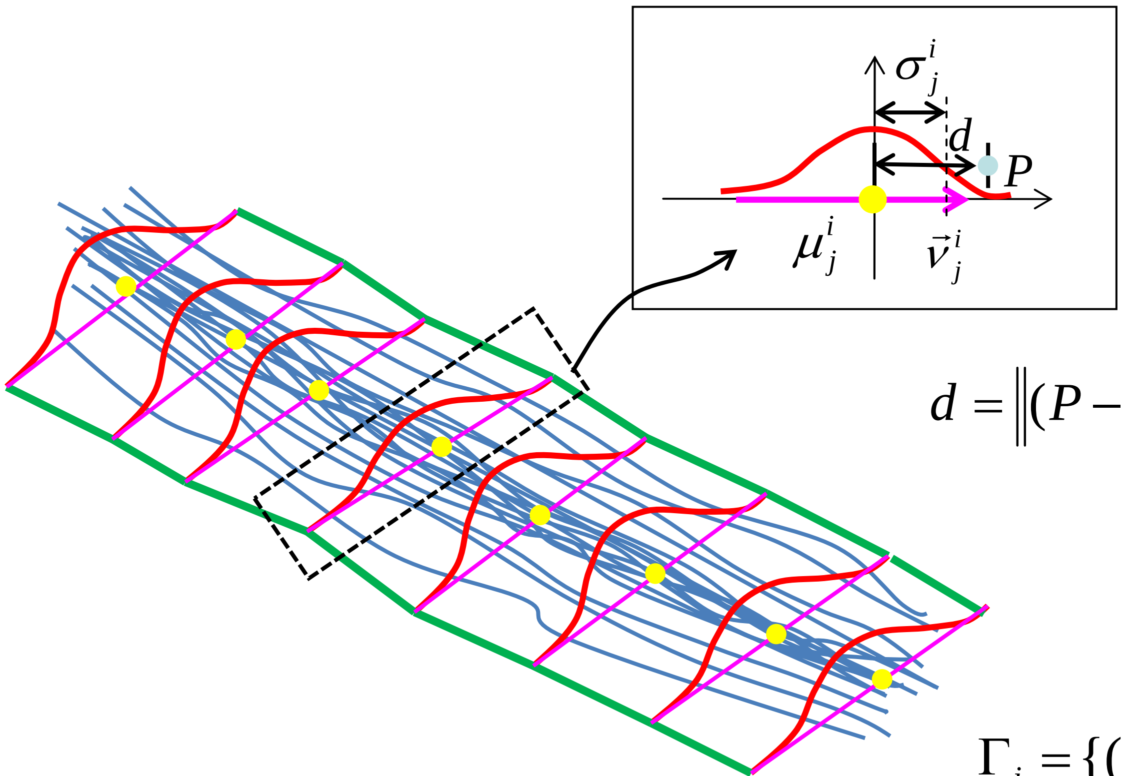
Crossing:

Turn along road:

Trajectory Clustering



Path Model



$$d = \left\|(P-\mu_j^i)\cdot\vec{v}_j^i\right\| \sim N(0,\sigma_j^i)$$

$$\Gamma_j = \{(\mu_j^i,\vec{v}_j^i,\sigma_j^i) \mid i = 1,2,\ldots,n_j\}$$

Each cross-section is represented by a Gaussian
A path is model as a sequence of Gausians

Trajectory Evaluation

Likelihood between a trajectory T_k and a path Γ_j is evaluated

$$P(T_k \mid \Gamma_j) \propto \prod_{p=1}^{L_k} P(T_k^p \mid \Gamma_j) \quad // \quad T_k^p \text{ is a trajectory point}$$

$$= \prod_{p=1}^{L_k} P(T_k^p \mid \Gamma_j^{i'}) \quad // \quad \Gamma_j^{i'} \text{ is the nearest cross-section to } T_k^p$$

$$P(T_k^p \mid \Gamma_j^{i'}) = \frac{1}{\sqrt{2\pi}\sigma_j^{i'}} e^{-d^2 / 2\sigma_j^{i'2}}$$

Trajectory Classification

Given a trajectory T_* , the objective is to classify it with in

$$class = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_n, \boxed{\Gamma_a}\}$$

as

$$\Gamma_* = \max_{\Gamma} \arg P(\Gamma \mid T_*)$$

where

$$P(\Gamma_i \mid T_*) = \frac{1}{\eta} P(T_* \mid \Gamma_i) P(\Gamma_i)$$

$$P(\Gamma_a \mid T_*) = \frac{1}{\eta} P(T_* \mid \Gamma_a) P(\Gamma_a)$$

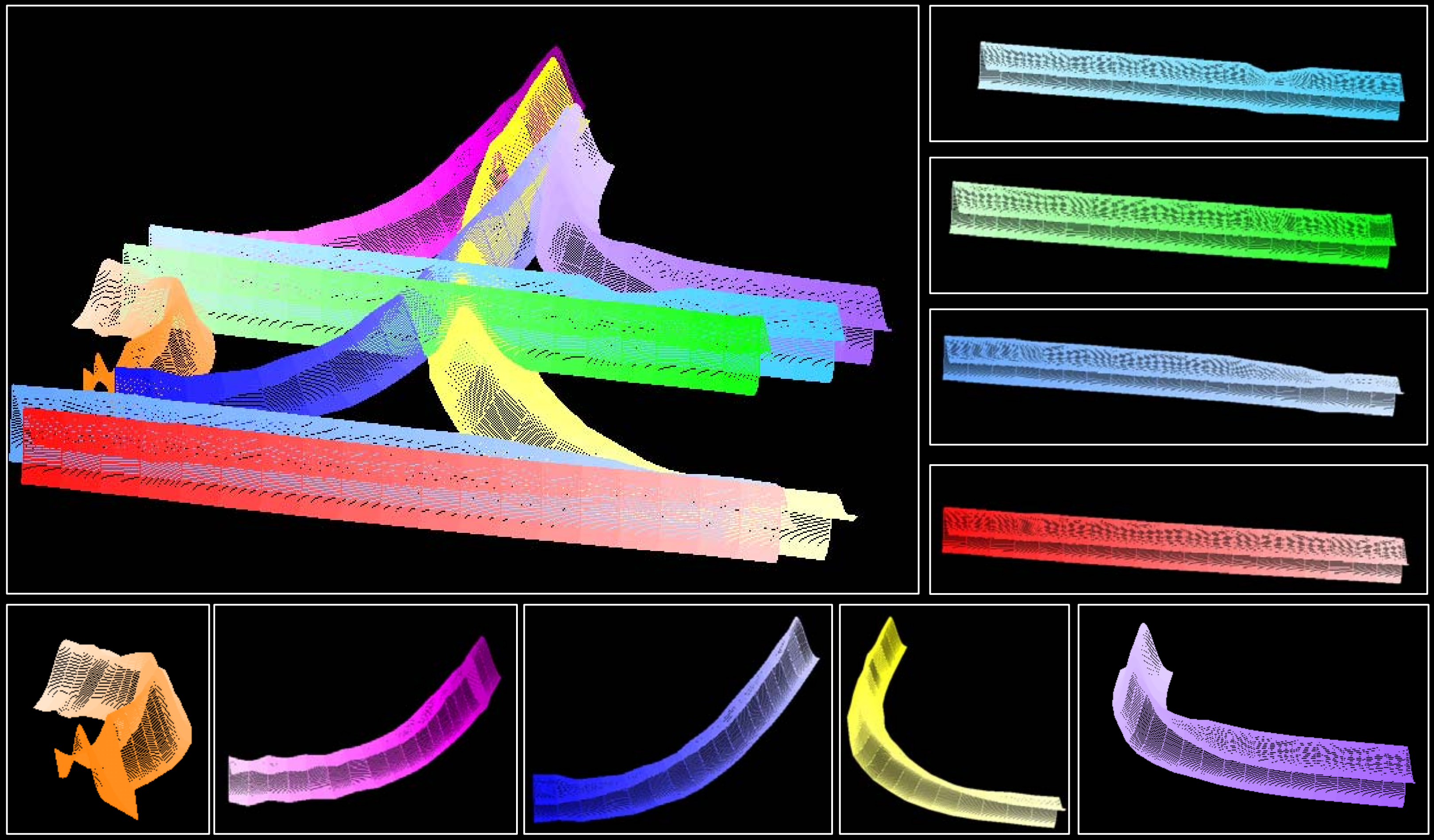
and

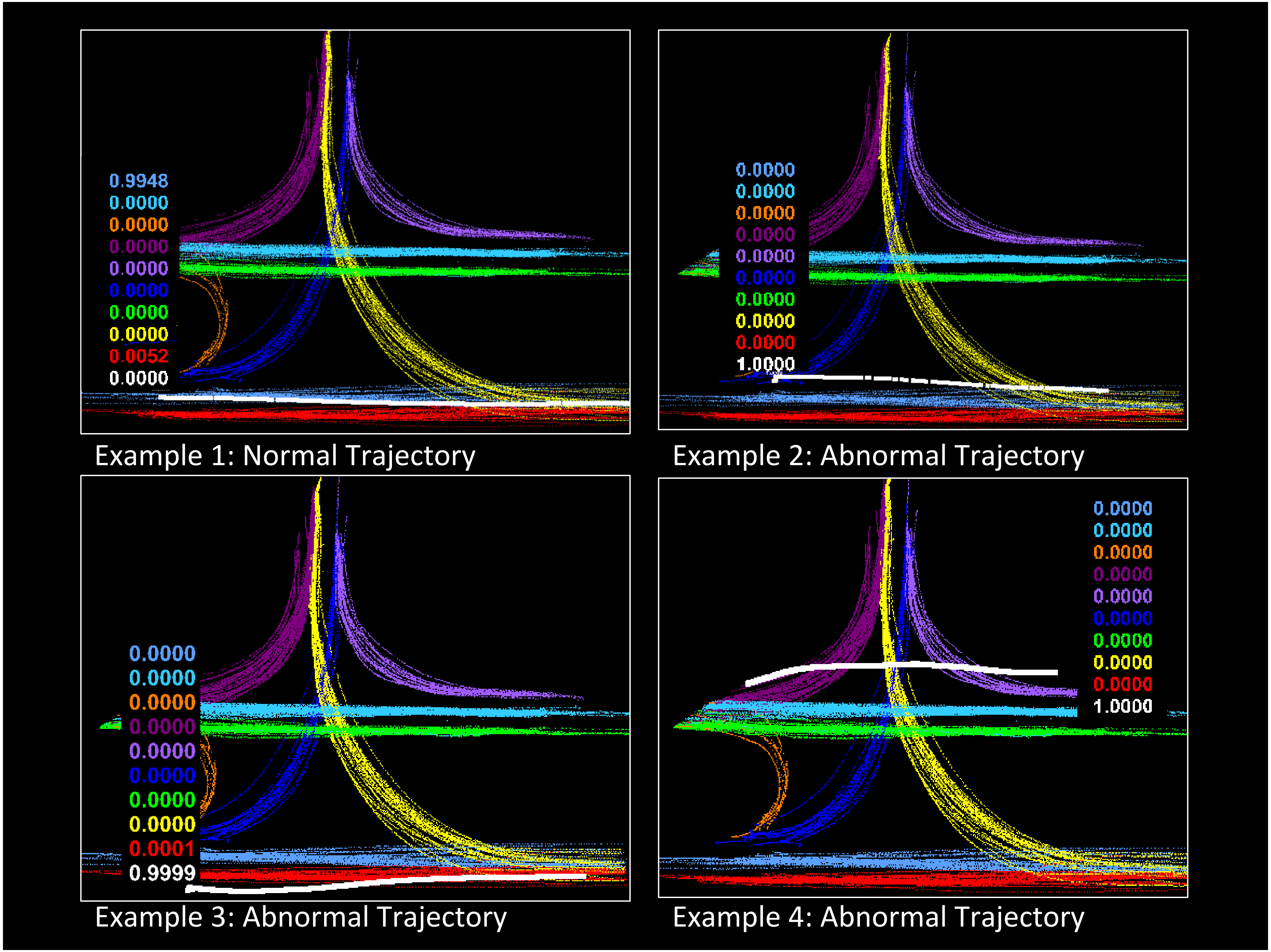
$$P(T_* \mid \Gamma_a) = \prod_{i=1}^{n} (P(T_* \mid \Gamma_i) < \varepsilon_i), \in \{0,1\}$$

$P(T_* \mid \Gamma_a)$ can be either 0 or 1.

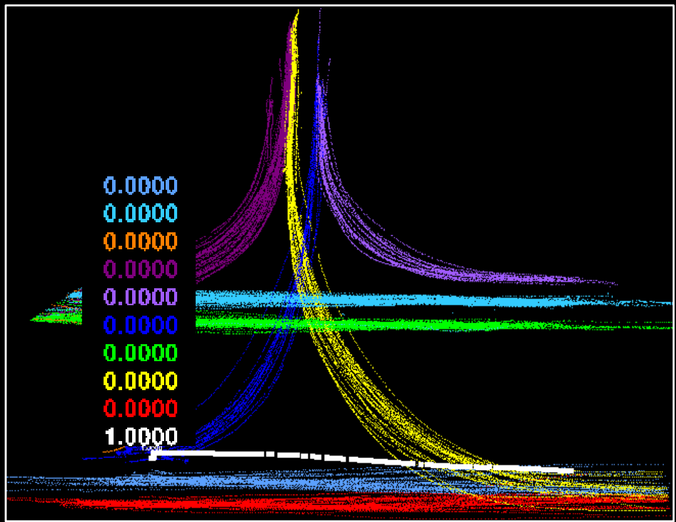
$P(T_* \mid \Gamma_a) = 1$ if and only if $\forall i \quad P(T_* \mid \Gamma_i) < \varepsilon_i$

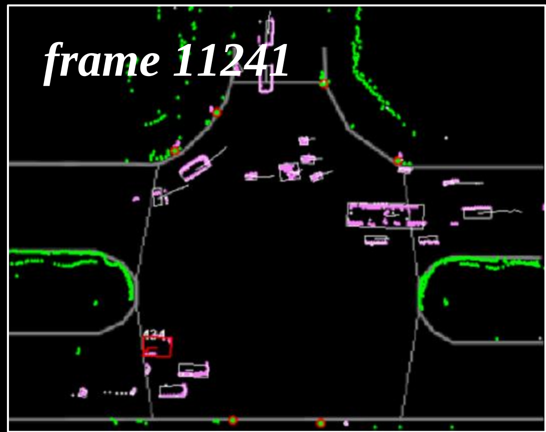
Path models

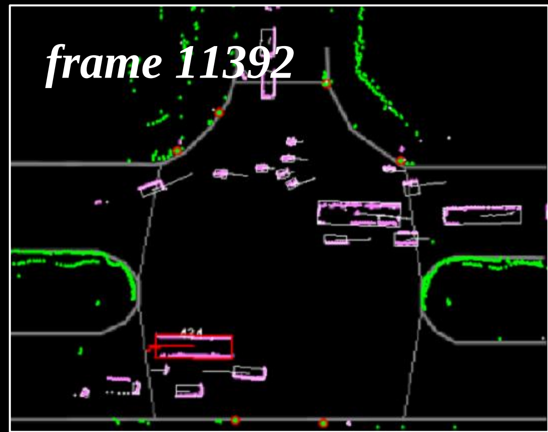


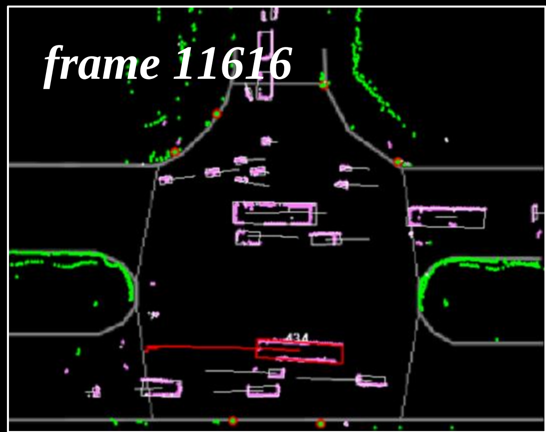

0.9948
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0052
0.0000
Example 1: Normal Trajectory
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
1.0000
Example 2: Abnormal Trajectory
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0001
0.9999
Example 3: Abnormal Trajectory
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
1.0000
Example 4: Abnormal Trajectory

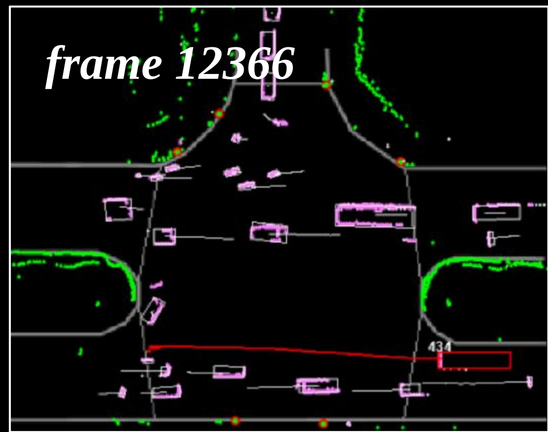
Real Abnormal Trajectory



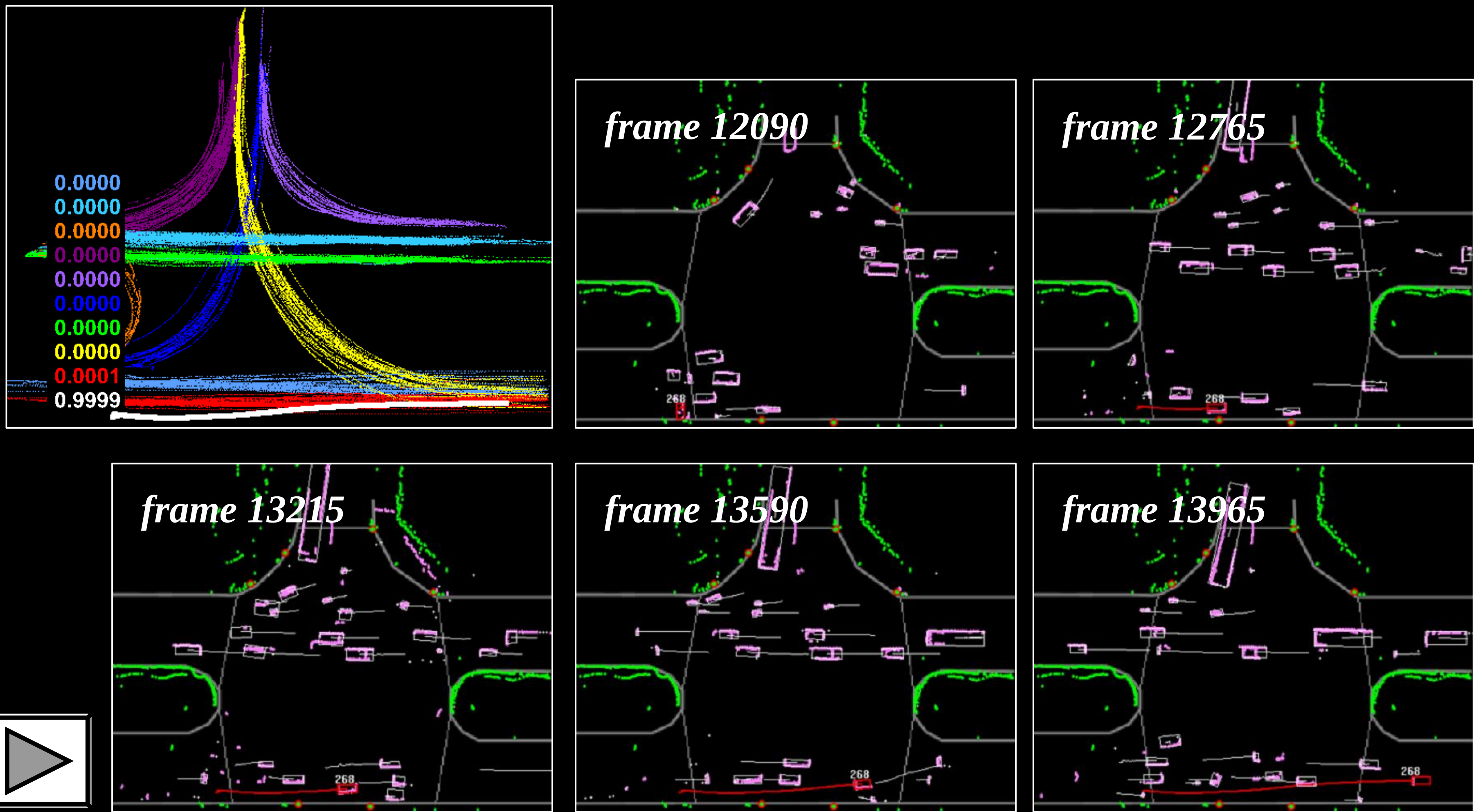




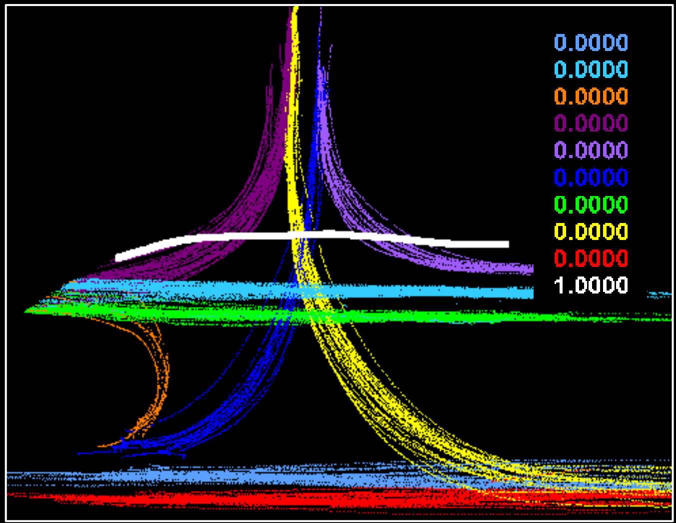


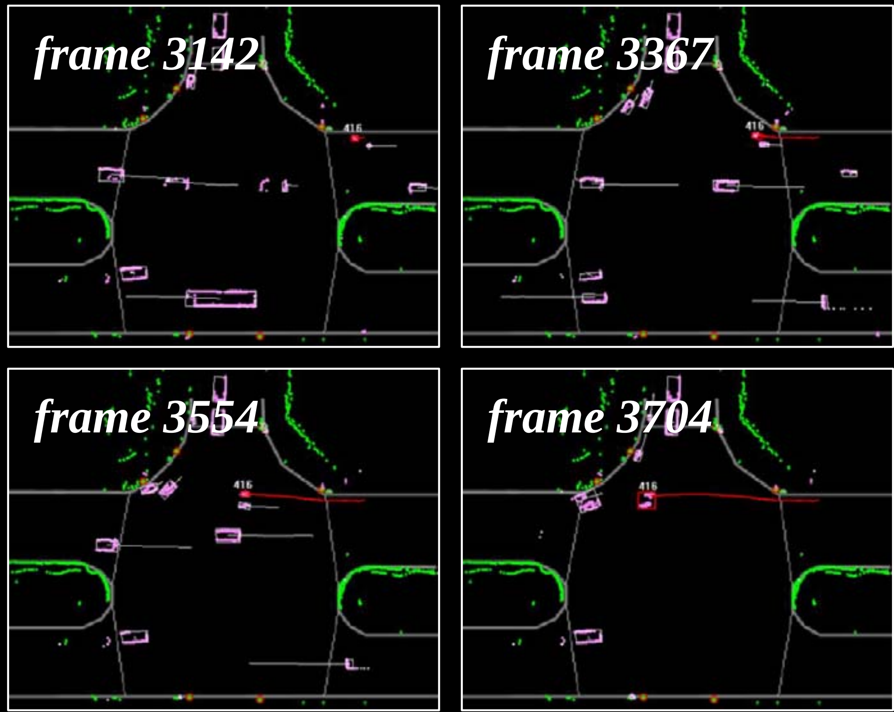


Real Abnormal Trajectory



False Abnormal Trajectory





Thank You!

Contact Info:

赵卉菁（Huijing Zhao）Tel：010 62757458
Email：zhaohj@cis.pku.edu.cn
http://www.cis.pku.edu.cn/faculty/vision/zhaohj/