Lecture 8
Applications of CNNs

Lin ZHANG, PhD
School of Software Engineering
Tongji University
Spring 2020

Outline

- Vision-based Parking-slot Detection
- Human-body Keypoint Detection

Outline

- Vision-based Parking-slot Detection
  - Background Introduction
  - General Flowchart
  - Surround-view Synthesis
  - Parking-slot Detection from Surround-view
  - Experiments
- Human-body Keypoint Detection

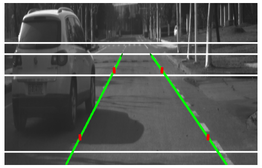
Background Introduction

- 同济大学智能型新能源协同创新中心（国家2011计划）

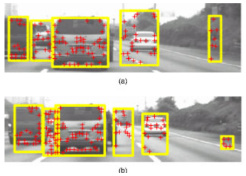
Background Introduction—ADAS Architecture

环境感知系统

毫米波雷达+前视相机+环视相机

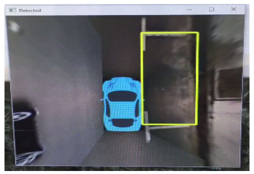


车道线检测

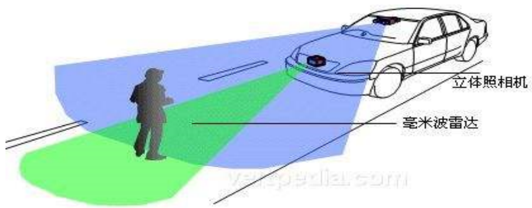


车辆及行人检测

交通标识检测



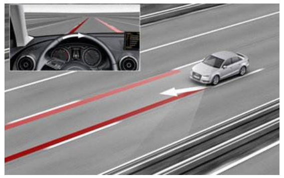
库位线检测



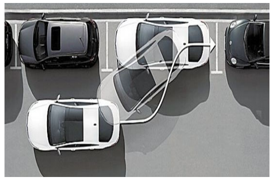
多源传感器信息融合

中央决策系统

中央决策控制器



车道保持

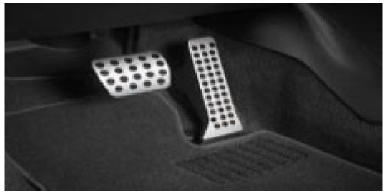


自动泊车

前向防撞

变道辅助

底层控制系统



驱/制动控制

转向控制

挡位控制

车身控制

Background Introduction

- Embarrassment in parking is one of the most difficult problems for drivers
- It is a challenge for a novice driver to park a car in a limited space

Automatic parking system is a hot research area in ADAS field

Background Introduction—ADAS Architecture

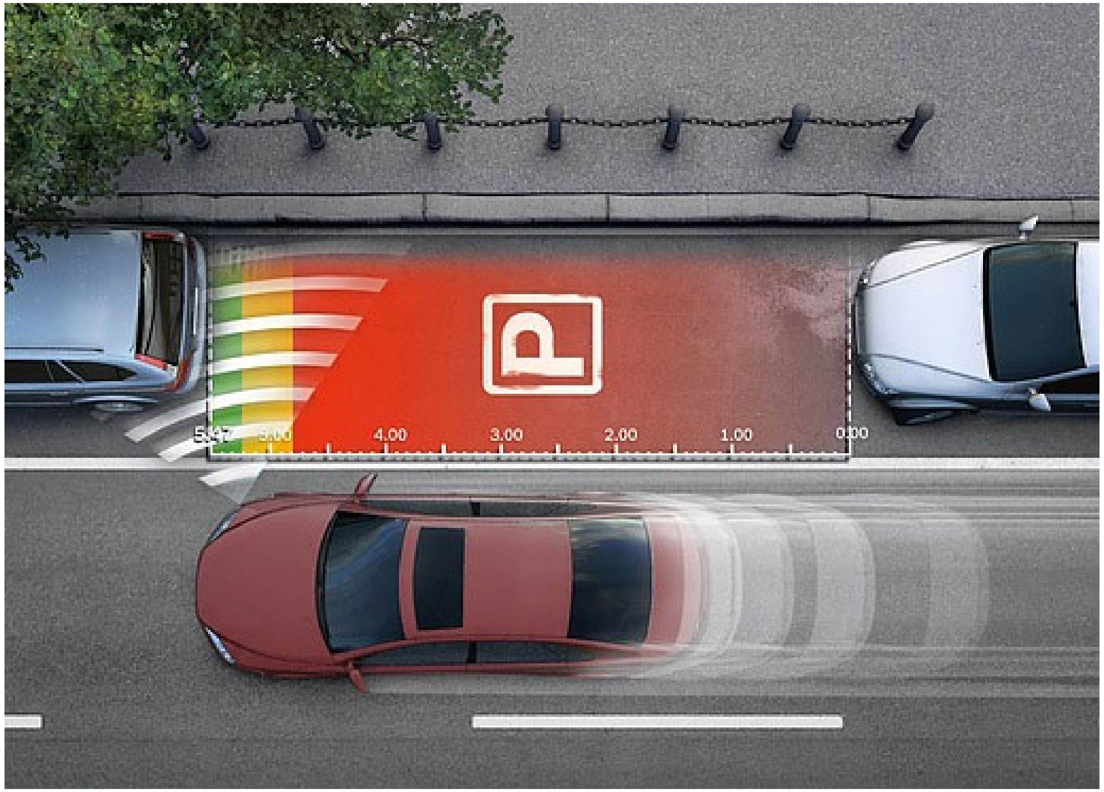
How to detect a parking-slot and return its position with respect to the vehicle coordinate system?

Different Ways to Locate a Parking-slot

- Infrastructure-based solutions
  - Need support from the parking site
  - Usually, the vehicle needs to communicate with the infrastructure

Different Ways to Locate a Parking-slot

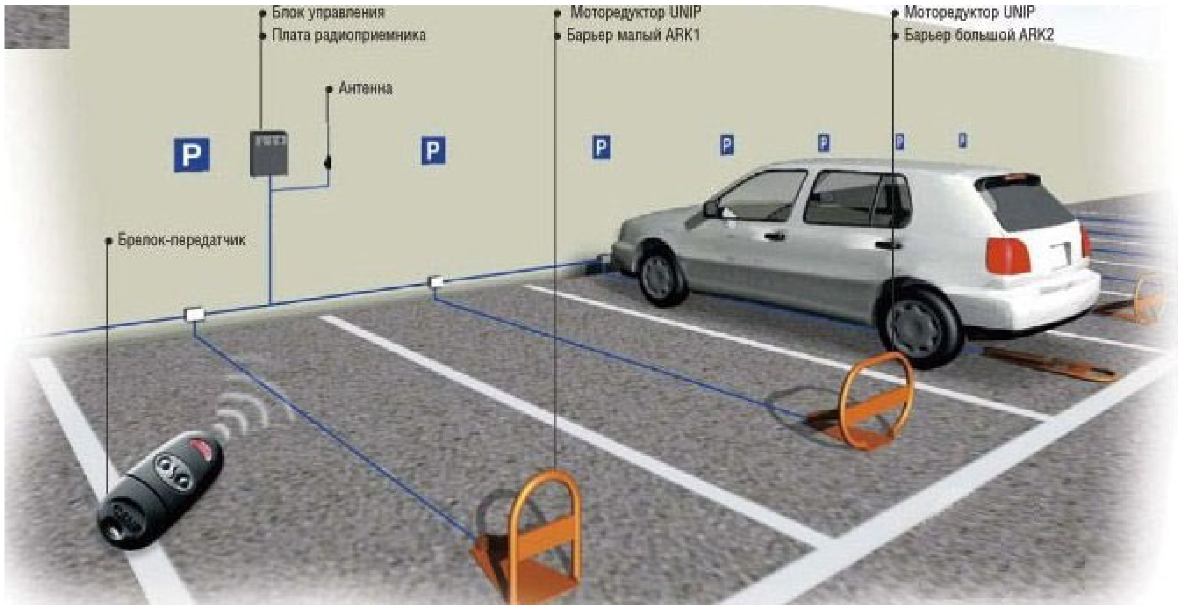
- Infrastructure-based solutions
- On-vehicle-sensor based solutions
    - Parking-vacancy detection
        - Ultrasonic radar
        - Stereo-vision
        - Depth camera



Different Ways to Locate a Parking-slot

- Infrastructure-based solutions
- On-vehicle-sensor based solutions
  - Parking-vacancy detection
  - Parking-slot (defined by lines, vision-based) detection

our focus



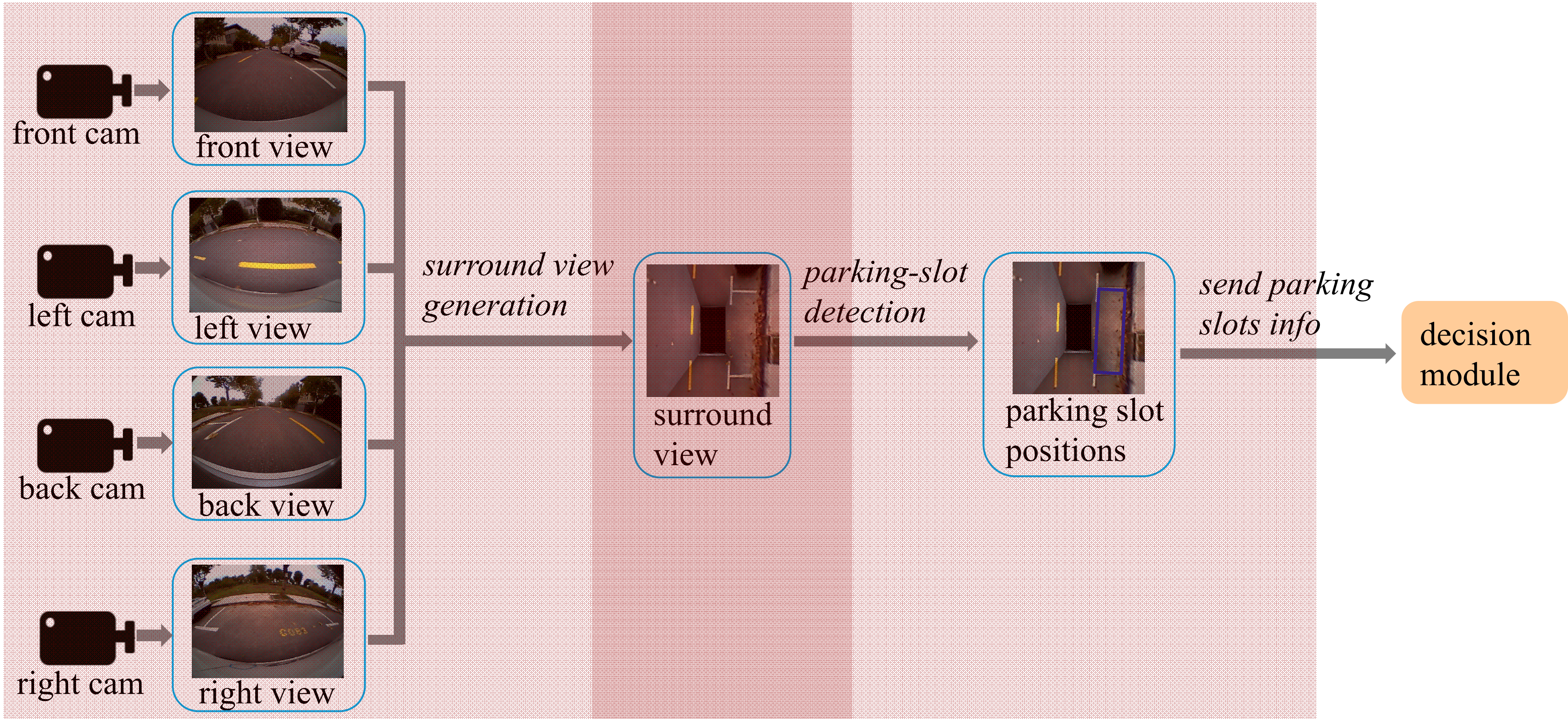
Research Gaps and Our Contributions

- Research Gaps
  - There is no publicly available dataset in this area
  - All the existing methods are based on low-level vision primitives (edges, corners, lines); large room for performance improvement
- Our contributions
  - ✓ Construct a large-scale labeled surround-view image dataset
  - ✓ Introduce machine learning theory into this field
  - ✓ Develop a real system that has been deployed on SAIC Roewe E50

Outline

- Vision-based Parking-slot Detection
  - Background Introduction
  - General Flowchart
  - Surround-view Synthesis
  - Parking-slot Detection from Surround-view
  - Experiments
- Human-body Keypoint Detection

General Flowchart



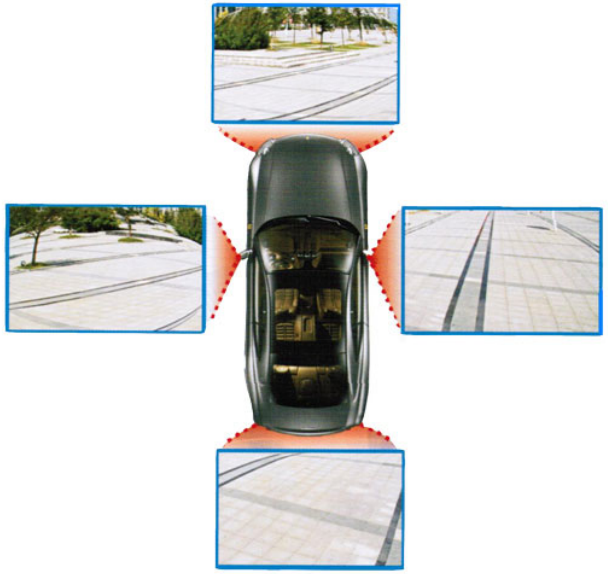
Overall flowchart of the vision-based parking slot detection system

Outline

- Vision-based Parking-slot Detection
  - Background Introduction
  - General Flowchart
  - Surround-view Synthesis
  - Parking-slot Detection from Surround-view
  - Experiments
- Human-body Keypoint Detection

Surround-view Synthesis

- Surround view camera system is an important ADAS technology allowing the driver to see a top-down view of the 360 degree surroundings of the vehicle
- Such a system normally consists of 4~6 wide-angle (fish-eye lens) cameras mounted around the vehicle, each facing a different direction

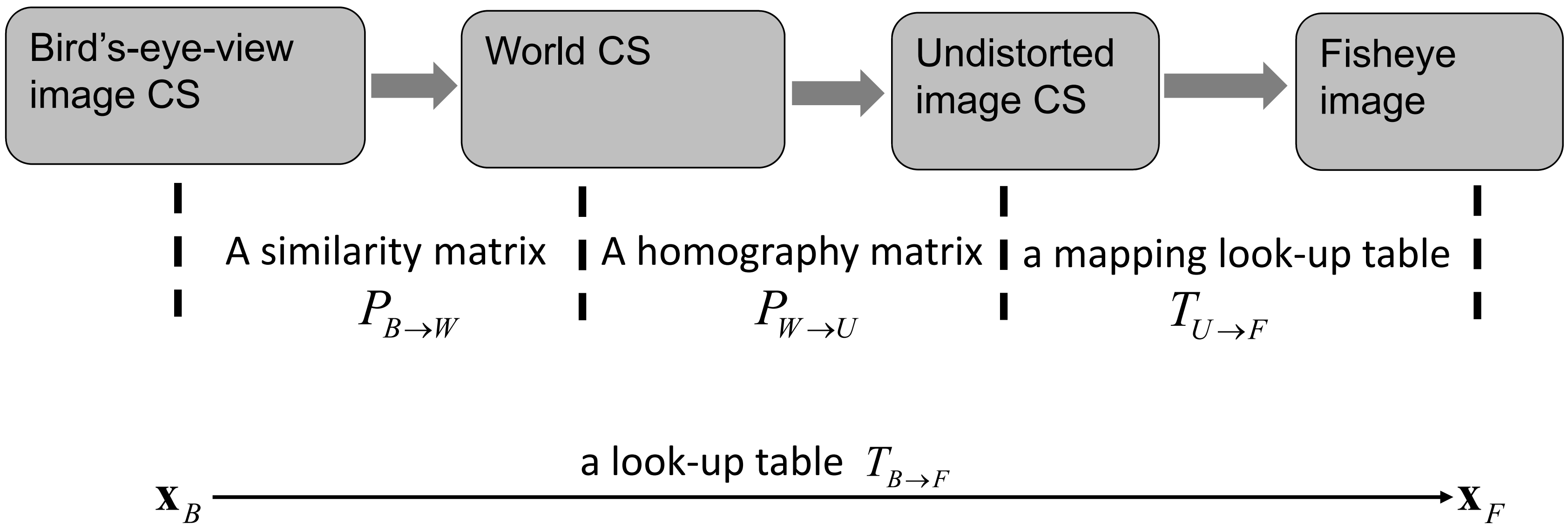


Surround-view Synthesis

- The surround-view is composed of the four bird's-eye views (front, left, back, and right)
- To get the bird's-eye view, the essence is generating a look-up table mapping a point on bird's-eye view to a point on the fish-eye image
  - Decide the similarity transformation matrix $P_{B\rightarrow W}$, mapping a point from the bird's-eye view coordinate system to the world coordinate system
  - Decide the projective transformation matrix $P_{W\rightarrow U}$, mapping a point from the world coordinate system to the undistorted image coordinate system
  - Decide the look-up table $T_{U\rightarrow F}$, mapping a point from the undistorted image coordinate system to the fish-eye image coordinate system

Surround-view Synthesis

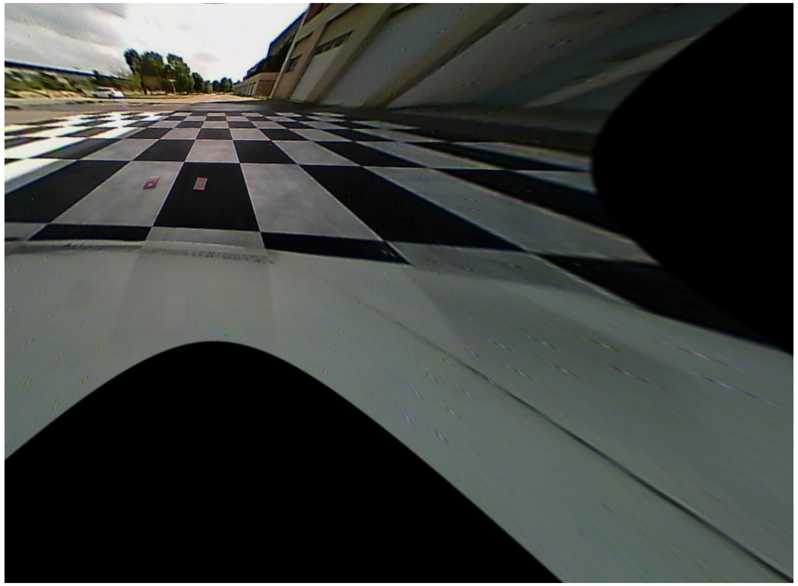
- Process to get the bird's-eye view



Surround-view Synthesis

- Process to get the bird's-eye view
  - Distortion coefficients of a fish-eye camera and also the mapping look-up table $T_{U \to F}$ can be determined by the calibration routines provided in openCV3.0

fisheye image



undistorted image

Surround-view Synthesis

- Process to get the bird's-eye view
  - Determine $P_{W\to U}$

    The physical plane (in WCS) and the undistorted image plane can be linked via a homography matrix $P_{W\to U}$

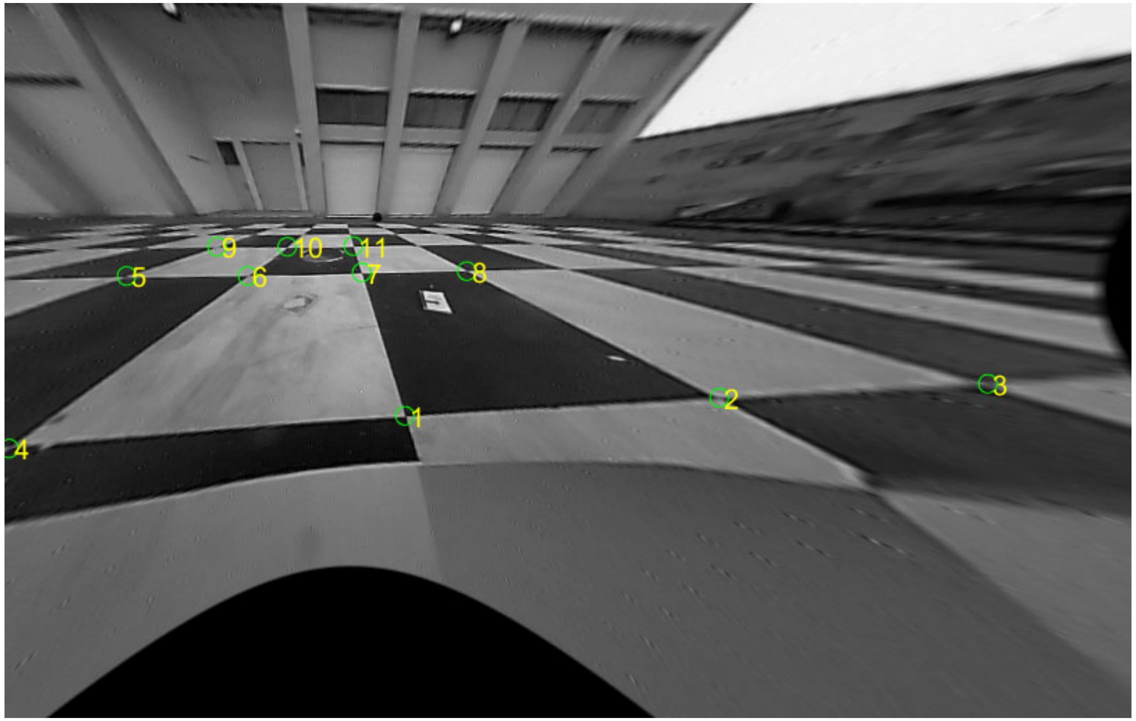
$$\mathbf{x}_U = P_{W\to U}\mathbf{x}_W$$

If we know a set of correspondence pairs $\left\{\mathbf{x}_{Ui}, \mathbf{x}_{Wi}\right\}_{i=1}^{N}$,

$P_{W\to U}$ can be estimated using the least-square method

Surround-view Synthesis

- Process to get the bird's-eye view
  - Determine $P_{W \to U}$



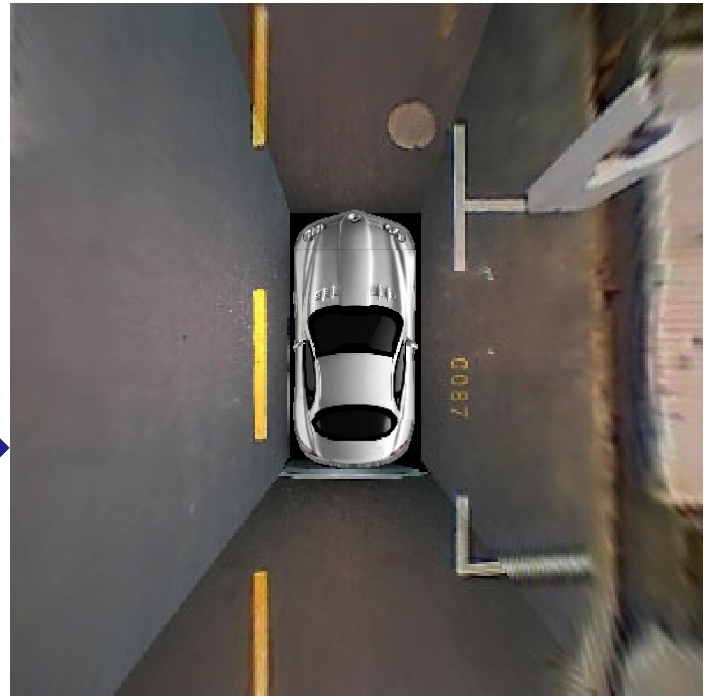
Surround-view Synthesis

(a)

(b)

(c)

(d)

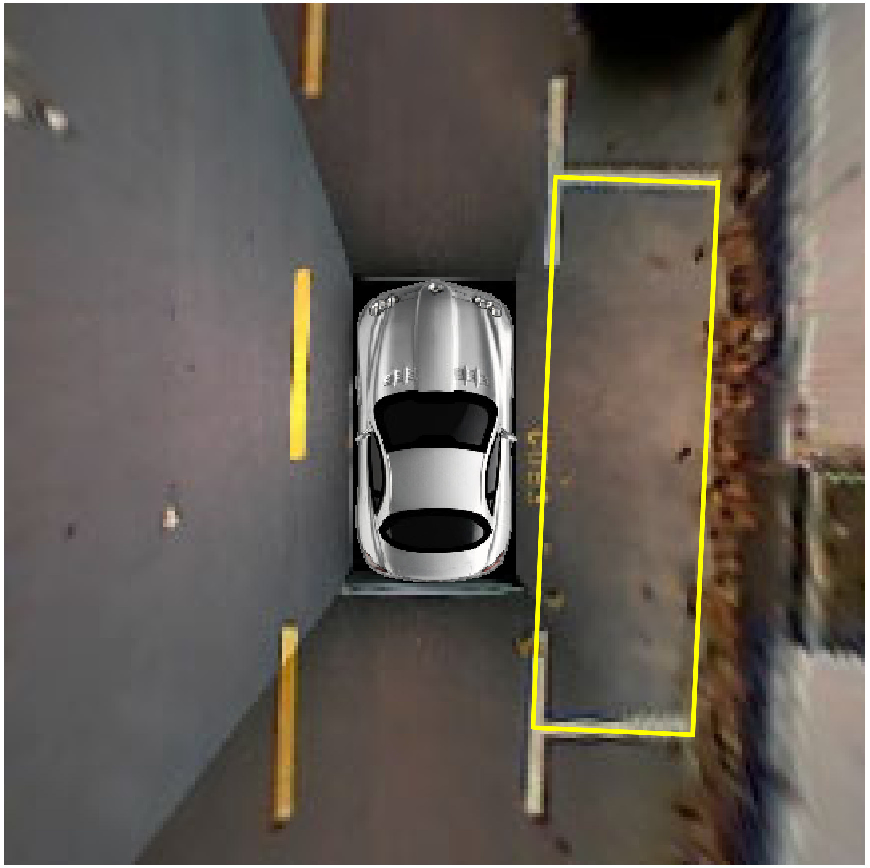


(e)

Image is of the size 600×600

$\Leftrightarrow$ $10m \times 10m$ physical region

Surround-view Synthesis



How to detect the parking-slot given a surround-view image?

Outline

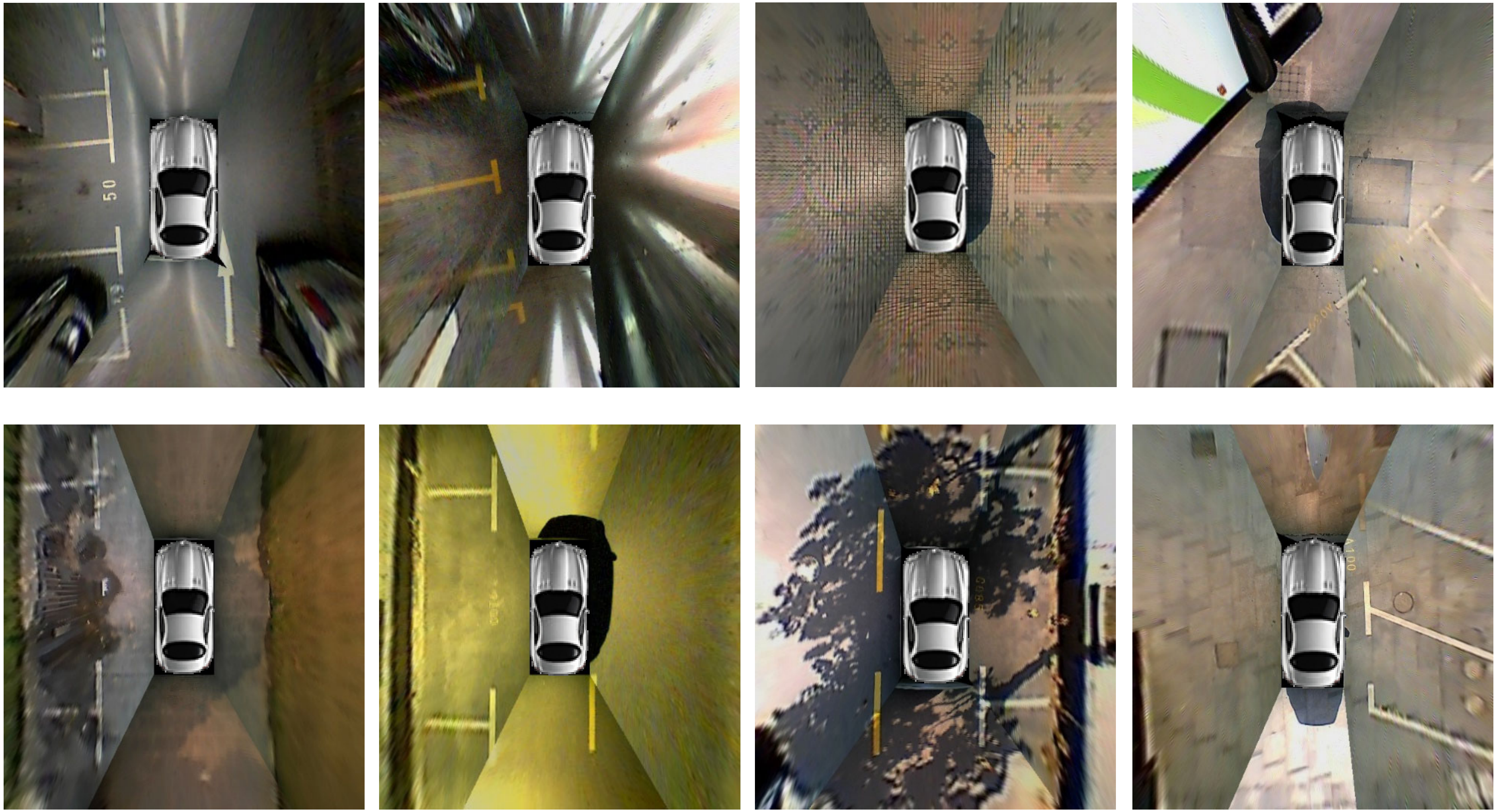
- Vision-based Parking-slot Detection
  - Background Introduction
  - General Flowchart
  - Surround-view Synthesis
  - Parking-slot Detection from Surround-view
  - Experiments
- Human-body Keypoint Detection

Challenges

- It is not an easy task due to the existence of
  - ✓ Various types of road textures
  - ✓ Various types of parking-slots
  - ✓ Illumination variation
  - ✓ Partially damaged parking-lines
  - ✓ Non-uniform shadow

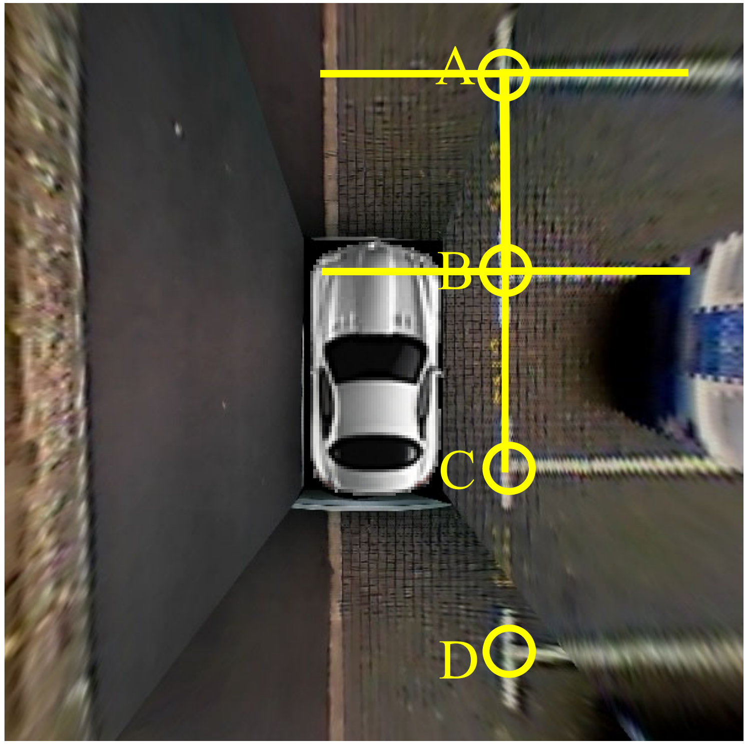
Making the low-level vision based algorithms difficult to succeed

Challenges



DeepPS: A DCNN-based Approach

- Motivation



✓ Detect marking-points

✓ Decide the validity of entrance-lines and their types (can be solved as a classification problem)

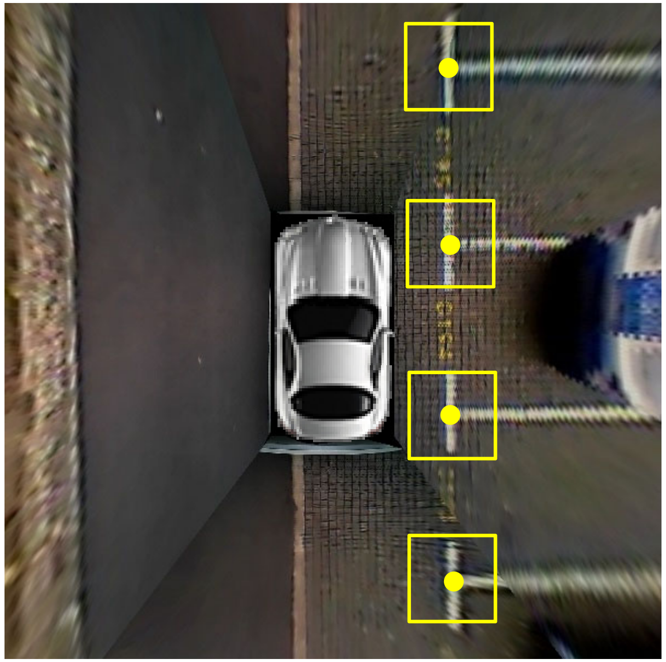
Both of them can be solved by DCNN-based techniques

DeepPS: A DCNN-based Approach

- Marking-point detection by using a DCNN-based framework
  - We adopt YoloV2 as the detection framework
    - R-CNN (Region-baed convolutional neural networks) (CVPR 2014)
    - SPPNet (Spatial Pyramid Pooling Network) (T-PAMI 2015)
    - Fast-RCNN (ICCV 2015)
    - Faster-RCNN (NIPS 2015)
    - Yolo (You Only Look Once) (CVPR 2016)
    - SSD (Single Shot Multibox Detector) (ECCV 1016)
    - Yolov2 (ArXiv 2016) *Accurate enough, fastest!*

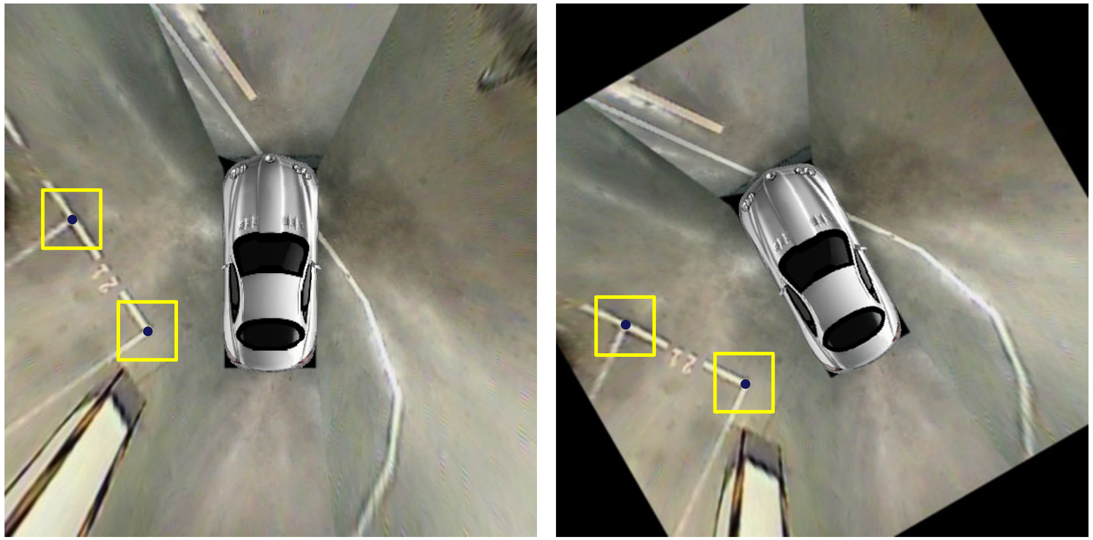
DeepPS: A DCNN-based Approach

- Marking-point detection by using a DCNN-based framework
  - We adopt YoloV2 as the detection framework
  - Manually mark the positions of marking-points and define regions with fixed size centered at marking-points as “marking-point patterns”



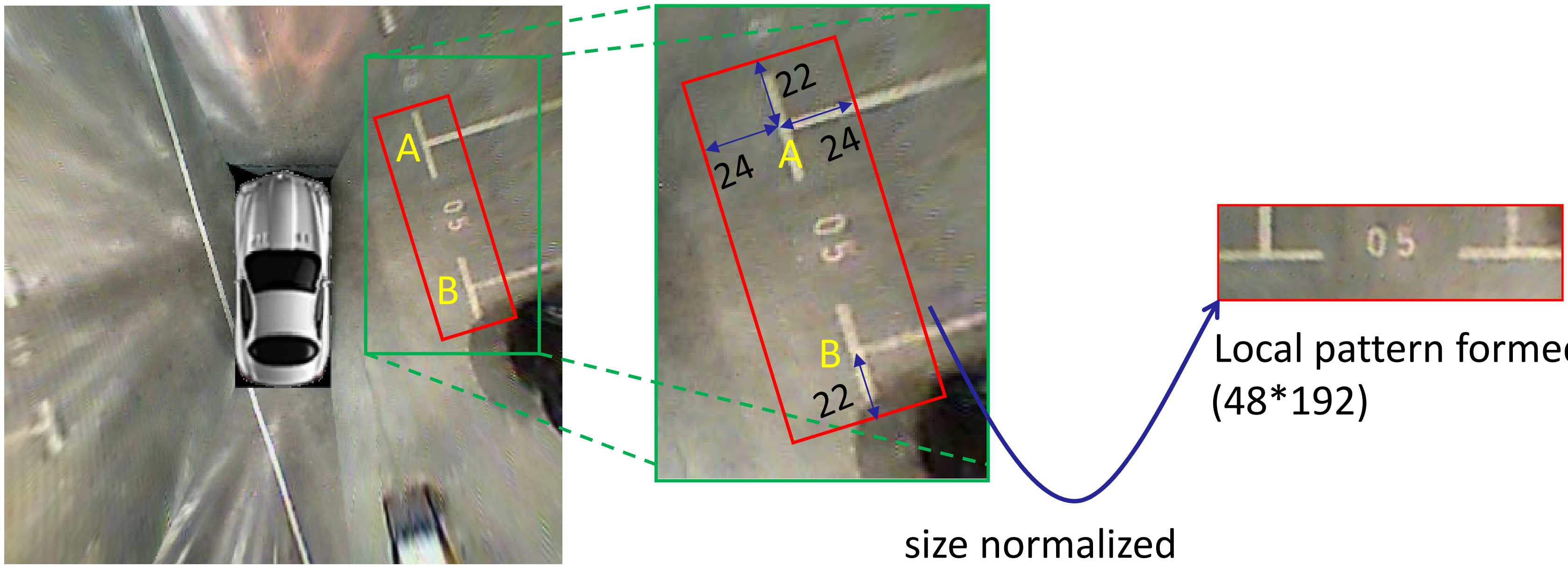
DeepPS: A DCNN-based Approach

- Marking-point detection by using a DCNN-based framework
  - We adopt YoloV2 as the detection framework
  - Manually mark the positions of marking-points and define regions with fixed size centered at marking-points as “marking-point patterns”
  - To make the detector rotation-invariant, we rotate the training images (and the associated labeling information) to augment the training dataset



DeepPS: A DCNN-based Approach

- Given two marking points A and B, classify the local pattern formed by A and B for two purposes
  - Judge whether “AB” is a valid entrance-line
  - If it is, decide the type of this entrance-line



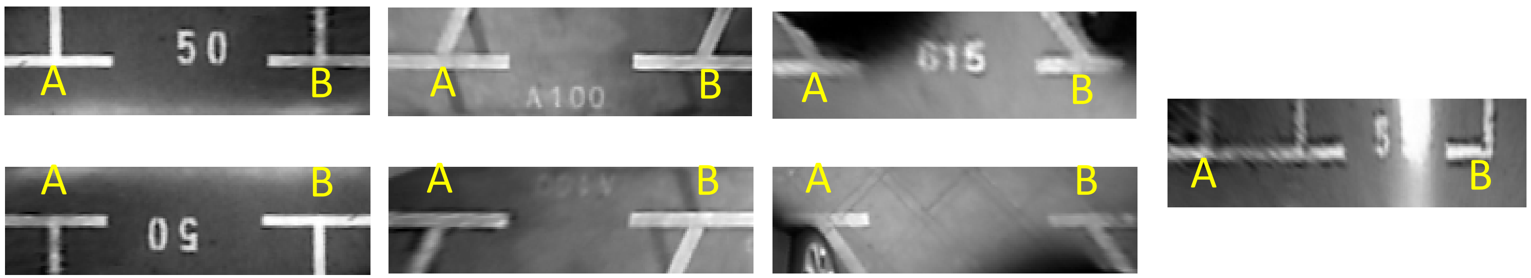
size normalized

Local pattern formed by A and B (48*192)

DeepPS: A DCNN-based Approach

- Given two marking points A and B, classify the local pattern formed by A and B for two purposes
  - Judge whether “AB” is a valid entrance-line
  - If it is, decide the type of this entrance-line

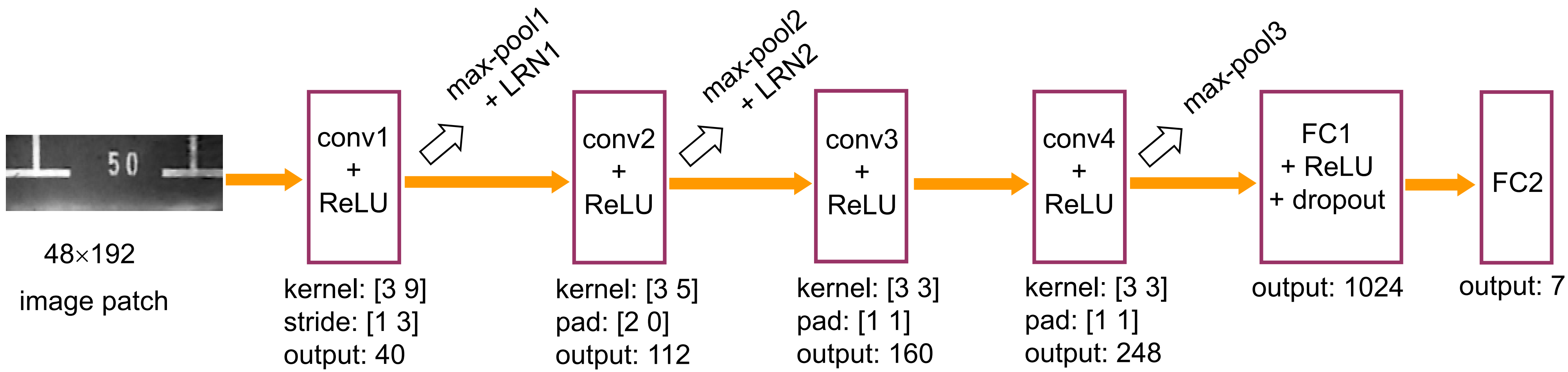
We define 7 types of local patterns formed by two marking-points



Typical samples of 7 types of local patterns

DeepPS: A DCNN-based Approach

- To solve the local pattern classification problem, we design a DCNN model which is a simplified version of AlexNet



- Samples for slant parking-slots were quite rare, we use SMOTE[1] strategy to create more virtual samples

[1] N.V. Chawla *et al.*, SMOTE: Synthetic Minority Over-sampling Technique, J. Artificial Intelligence Research 16: 321-357, 2002

DeepPS: A DCNN-based Approach

- For a slant parking-slot, how to obtain the angle between its entrance-line and its separating lines?

Prepare a set of templates $\left\{T_{\theta_j}\right\}$ having different angles

Extract the two patches I_A and I_B around A and B after the direction is normalized

$$\alpha = \arg\max_{\theta_j} \left\{ I_A * T_{\theta_j} + I_B * T_{\theta_j} \right\}$$

Outline

- Vision-based Parking-slot Detection
  - Background Introduction
  - General Flowchart
  - Surround-view Synthesis
  - Parking-slot Detection from Surround-view
  - Experiments
- Human-body Keypoint Detection

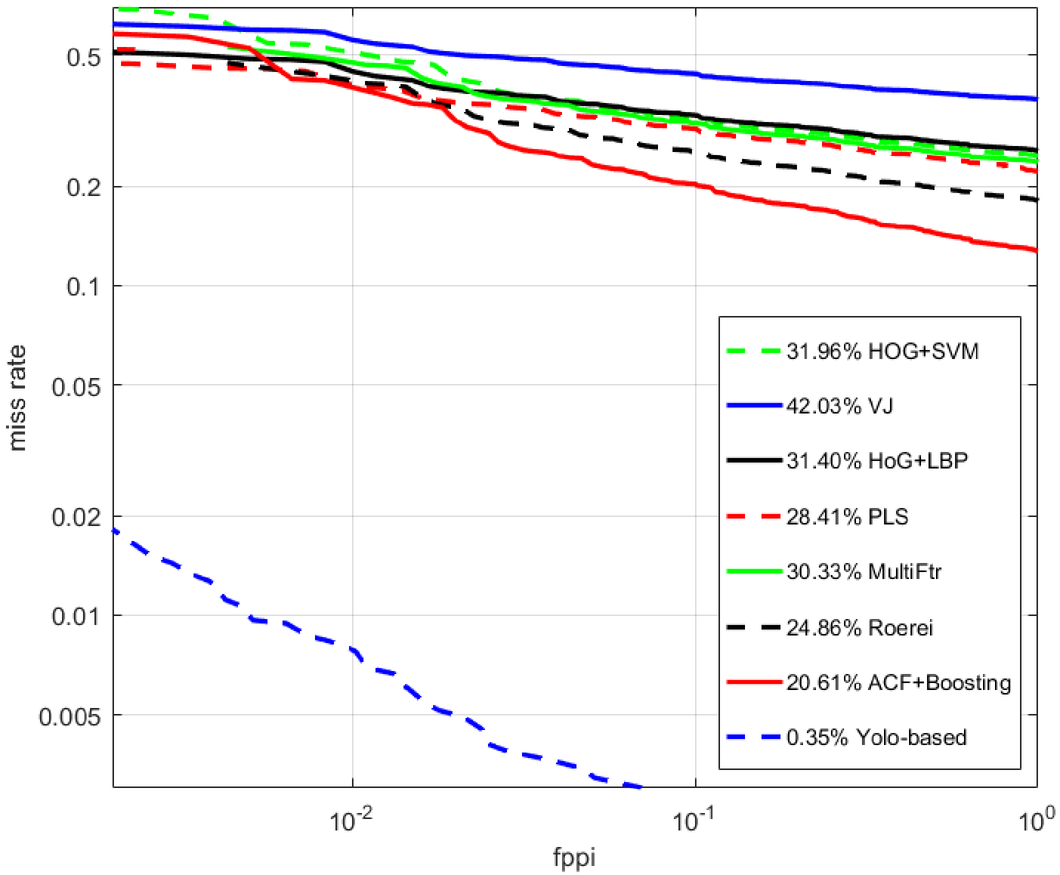
Dataset

- We collected and labeled a large-scale dataset
  - It covers vertical ones, parallel ones, and slant ones
  - Typical illumination conditions were considered
  - Various road textures were included
  - 9827 training images
  - 2338 test images
- Test set is separated into several subsets

Subset Name	Number of image samples
indoor parking lot	226
outdoor normal daylight	546
outdoor rainy	244
outdoor shadow	1127
outdoor street light	147
outdoor slanted	48

Marking-point detection accuracy

- Missing rates VS FPPI curves on the entire test set



Marking-point localization accuracy

- Statistics of the distances of the detected marking-points with the matched labeled ones

detection methods	mean and std (in pixels)	mean and std (in cm)
ACF + Boosting	2.86 ± 1.54	4.77 ± 2.57
YoloV2-based	1.55 ± 1.05	2.58 ± 1.75

Parking-slot detection accuracy

- Precision-Recall rates of different parking-slot detection methods

method	precision	recall
Jung *et al.*'s method	98.38%	52.39%
Wang *et al.*'s method	98.27%	56.16%
Hamada *et al.*'s method	98.29%	60.41%
Suhr&Jung's method	98.38%	70.96%
PSD_L	98.55%	84.64%
DeepPS	**99.67%**	**98.76%**

Parking-slot detection accuracy

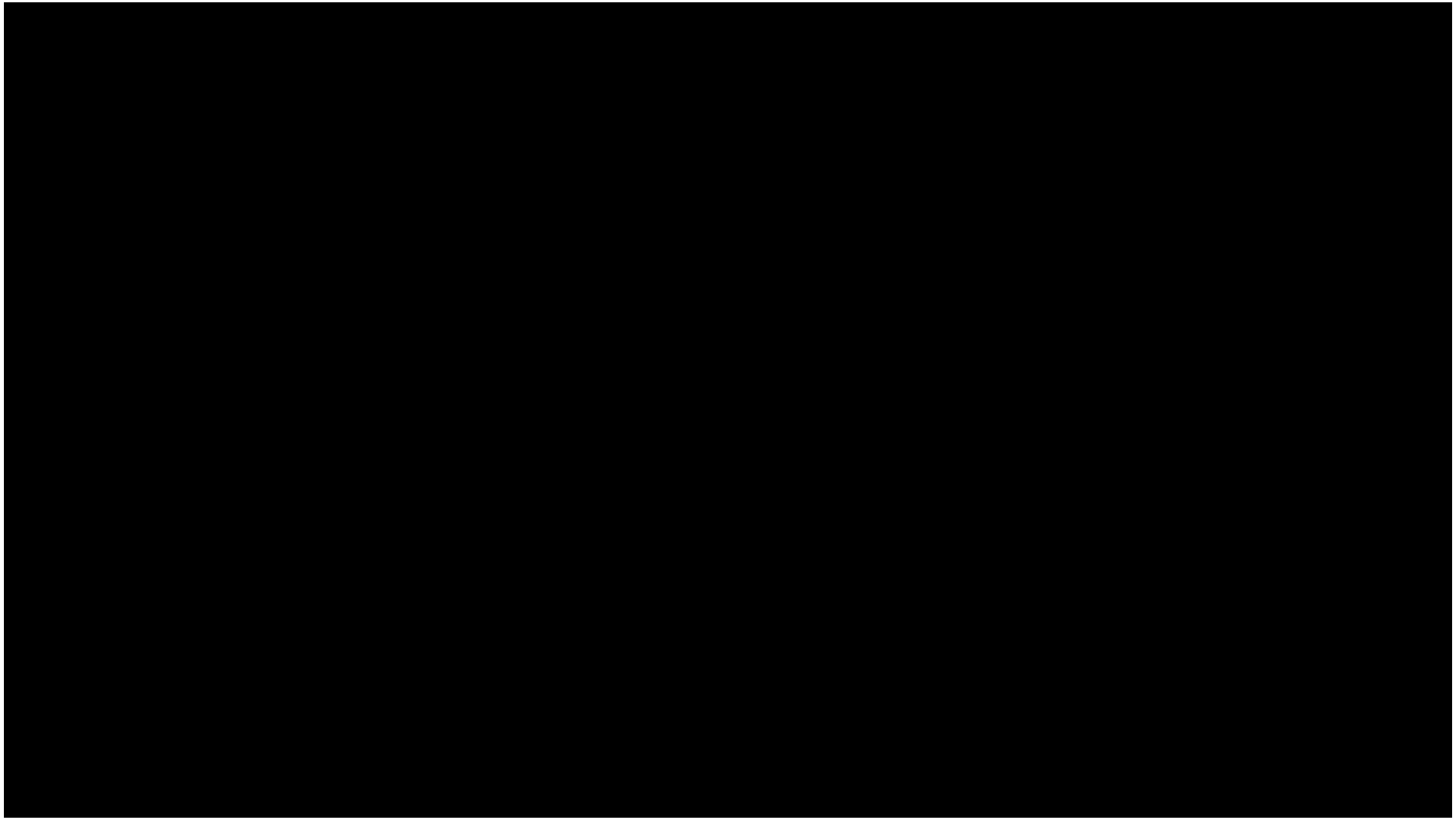
- Precision-Recall rates of two best performing methods on subsets

subset	PSD_L (precision, recall)	DeepPS (precision, recall)
indoor-parking lot	(99.34%, 87.46%)	(100%, 97.67%)
outdoor-normal daylight	(99.44%, 91.65%)	(99.61%, 99.23%)
outdoor-rainy	(98.68%, 87.72%)	(100%, 99.42%)
outdoor-shadow	(97.52%, 73.67%)	(99.86%, 99.14%)
outdoor-street light	(98.92%, 92.00%)	(100%, 100%)
outdoor-slanted	(93.15%, 83.95%)	(96.15%, 92.59%)

About the computational cost

- Workstation configuration
  - GPU: Nvidia Pascal Titan X
  - CPU: 2.4GHZ Intel Xeon E5-2630V3
  - RAM: 32GB
- It can process one frame within 25ms

Demo Video for PS Detection



Demo Video for Our Self-parking System

• 2017年5月17日，上海市委书记韩正调研同济期间，参观了“短程自主泊车系统”，其中的基于视觉的泊车位检测技术由本课题组完成

Outline

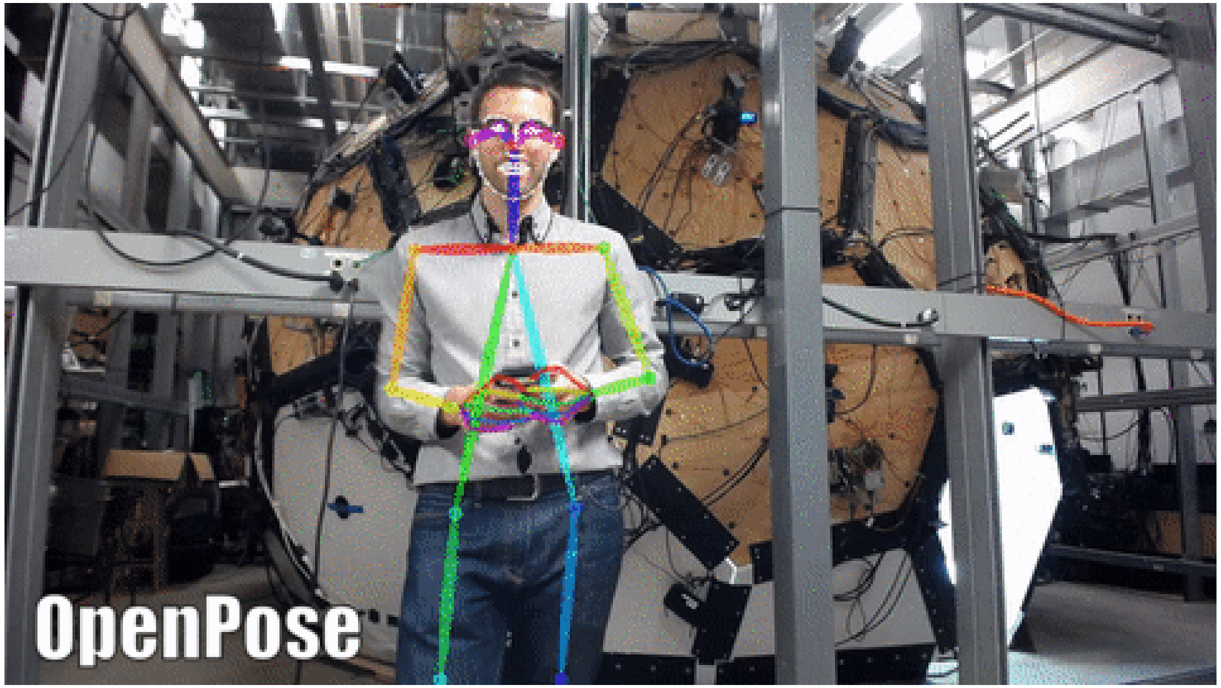
- Vision-based Parking-slot Detection
- Human-body Keypoint Detection

Outline

- Vision-based Parking-slot Detection
- Human-body Keypoint Detection
  - Problem definition
  - OpenPose

Problem Definition

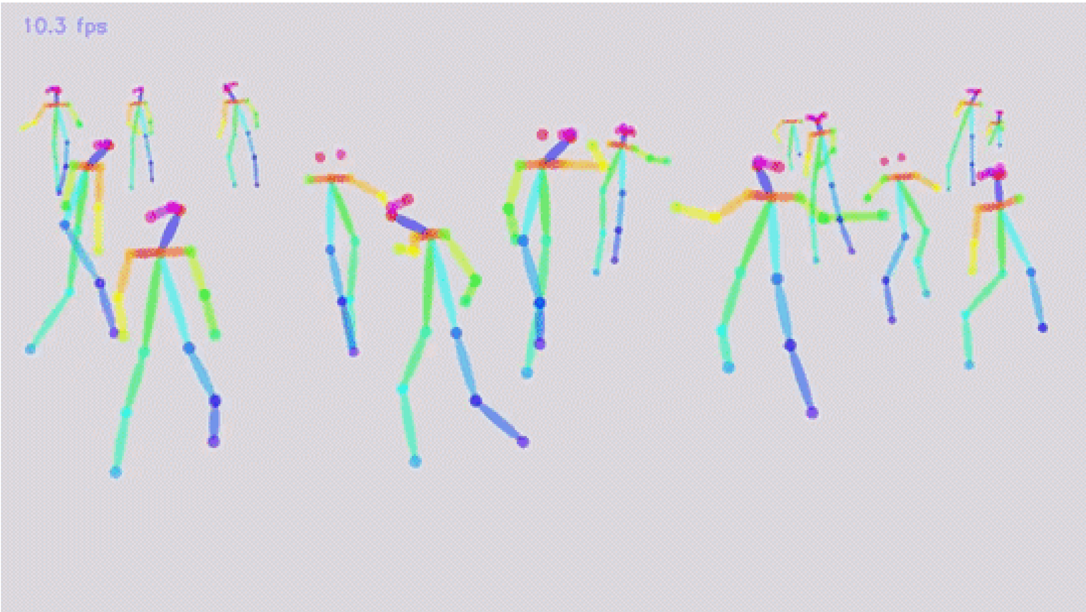
- Human-body Keypoints

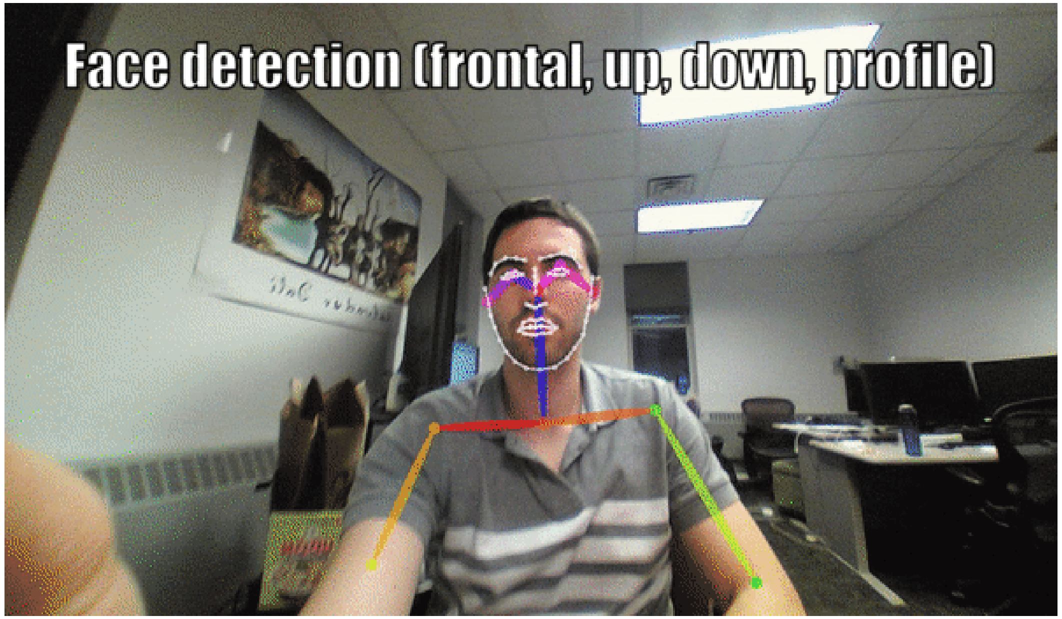


- Potential applications
  - Behavior analysis

OpenPose[1]

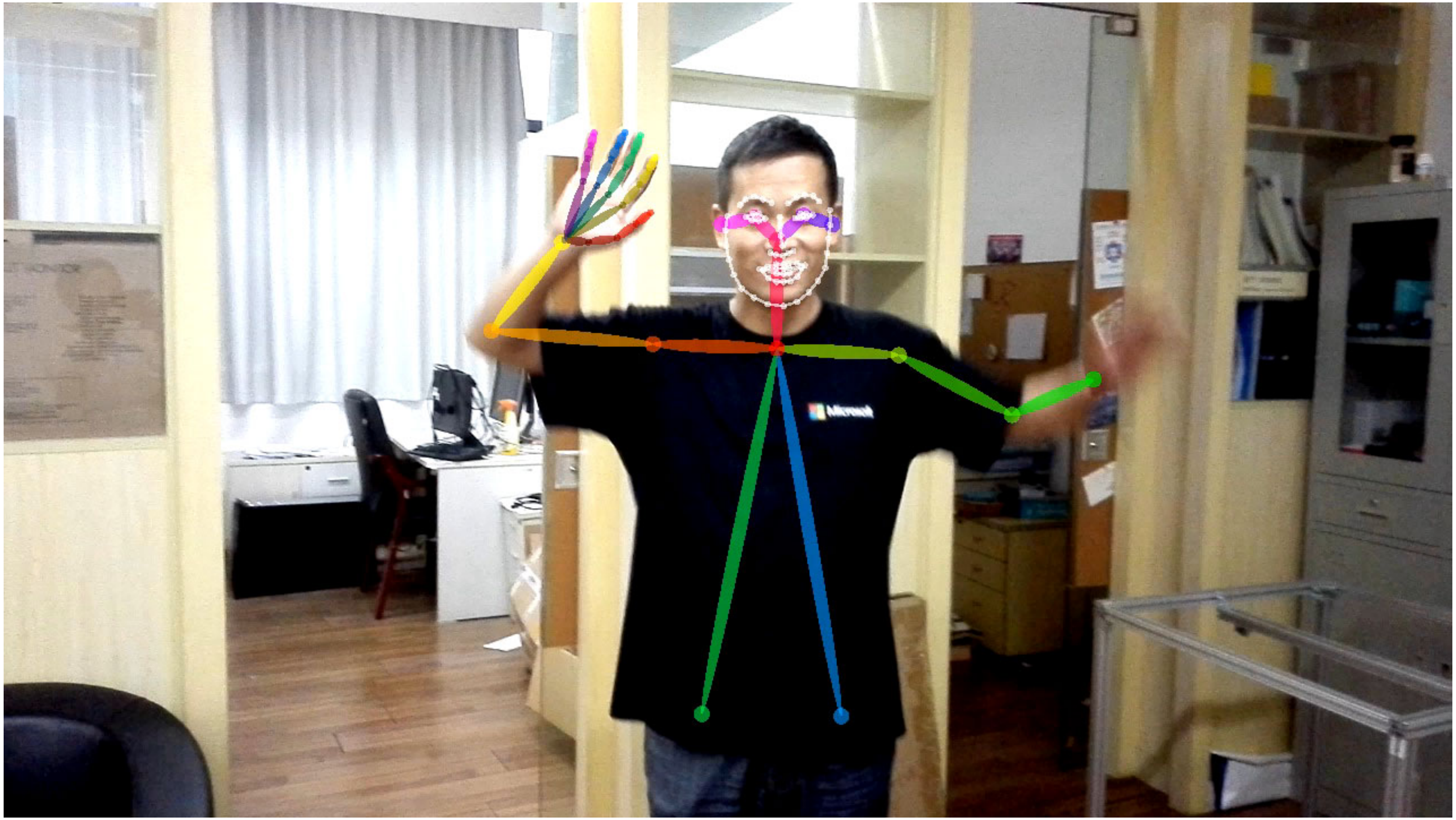
- OpenPose
  - A CNN-based library for human-body keypoint detection
  - With Nividia Titan XP GPU, its frame rate is about 15 fps
  - Support both Windows and Ubuntu





[1] Z. Cao et al., Realtime multi-person 2D pose estimation using part affinity fields, CVPR, 2017

OpenPose



Thanks!