自动驾驶系统中的视觉感知实践

张 林
同济大学
2018年12月

提纲

- 背景概述
- 泊车位检测与定位
- 减速带与行人的检测与测距
- 嵌入式平台实现
- 病态曝光图像的复原

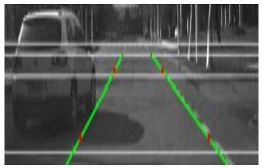
背景概述

- 同济大学智能型新能源协同创新中心（国家2011计划）

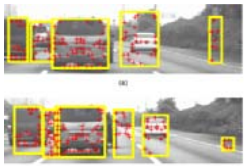
背景概述

环境感知系统

毫米波雷达+前视相机+环视相机



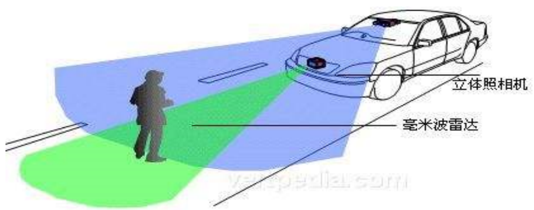
车道线检测



车辆及行人检测

交通标识检测

库位线检测

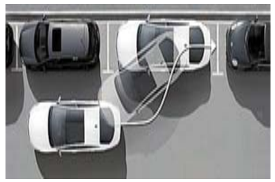


多源传感器信息融合

中央决策系统

中央决策控制器

车道保持

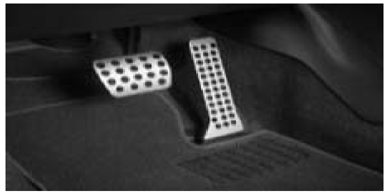


自动泊车

前向防撞

变道辅助

底层控制系统



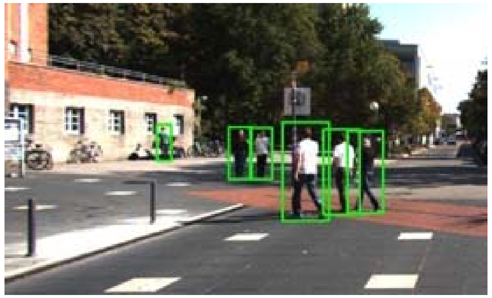
驱/制动控制

转向控制

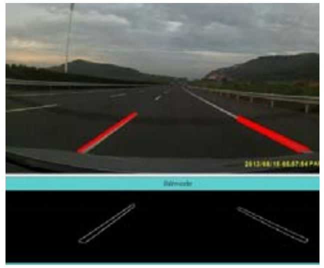
挡位控制

车身控制

背景概述


行人检测


车辆检测

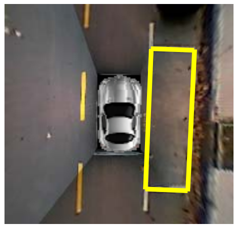

车道线检测


信号灯状态识别


CV在智能驾驶系统中


视觉SLAM

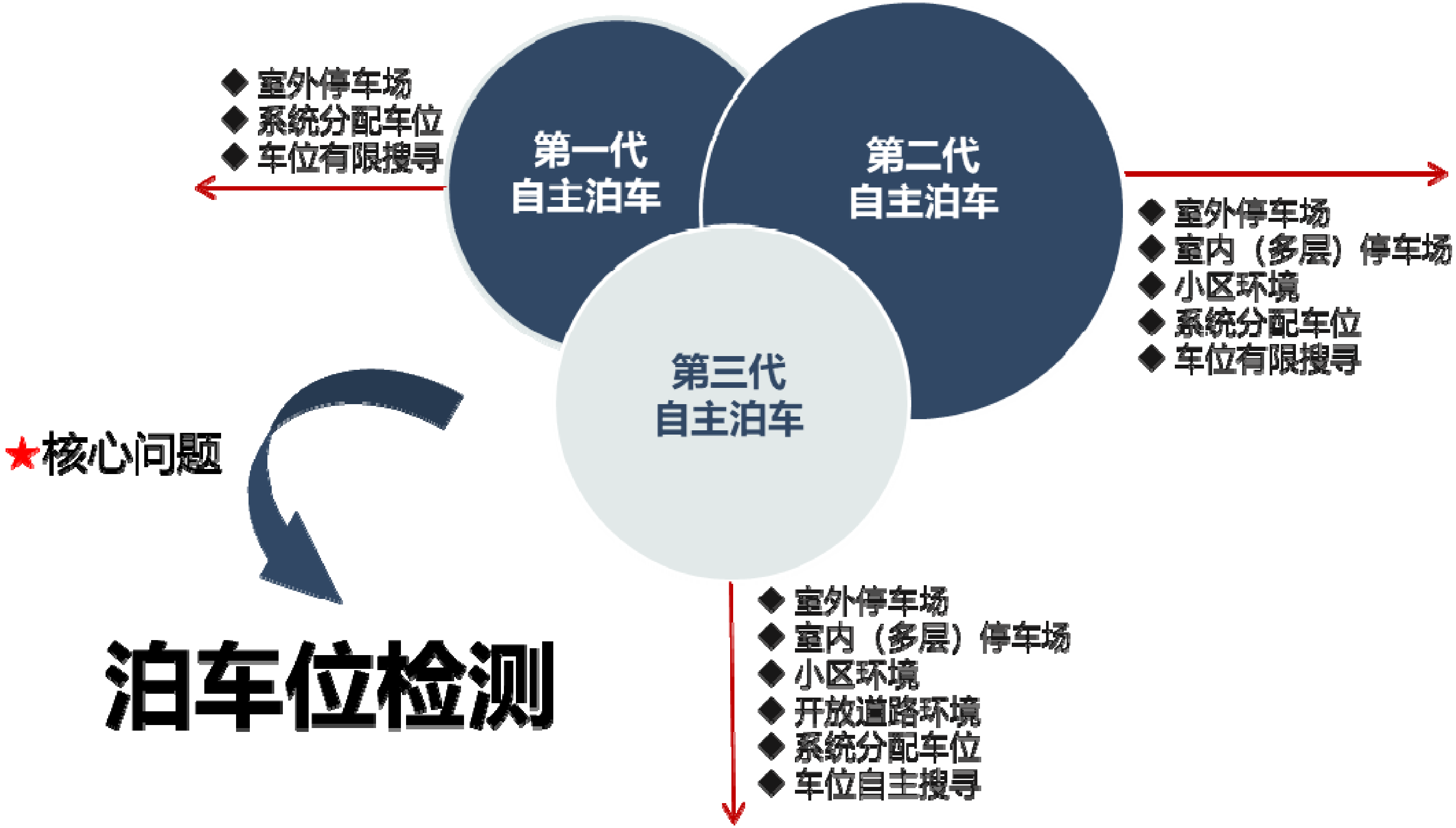

减速带识别


泊车位识别

提纲

- 背景概述
- 泊车位检测与定位
  - 背景
  - 总体流程
  - 环视图
  - 泊车位检测算法
  - 实验
- 减速带与行人的检测与测距
- 嵌入式平台实现
- 病态曝光图像的复原

背景



◆ 室外停车场
◆ 系统分配车位
◆ 车位有限搜寻
第一代
自主泊车
第二代
自主泊车
◆ 室外停车场
◆ 室内（多层）停车场
◆ 小区环境
◆ 系统分配车位
◆ 车位有限搜寻
第三代
自主泊车
★核心问题
泊车位检测
◆ 室外停车场
◆ 室内（多层）停车场
◆ 小区环境
◆ 开放道路环境
◆ 系统分配车位
◆ 车位自主搜寻

背景

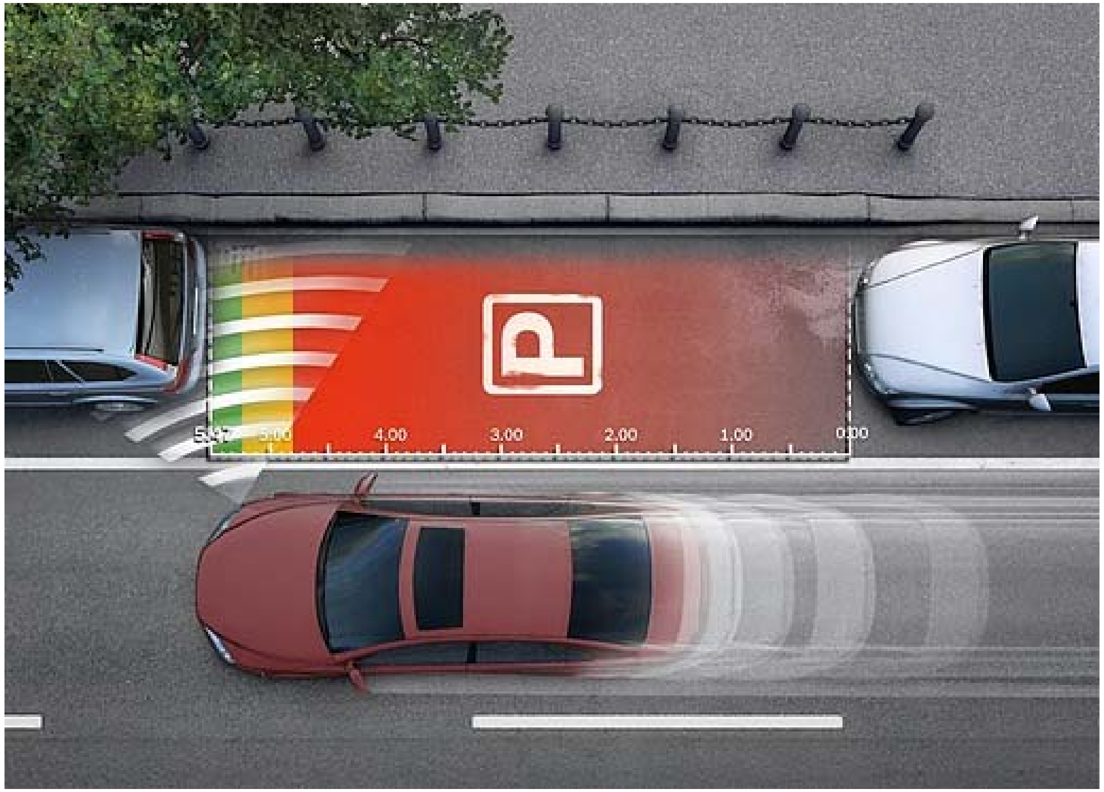
如何检测到泊车位并返回其在车辆坐标系下的坐标?

背景

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

背景

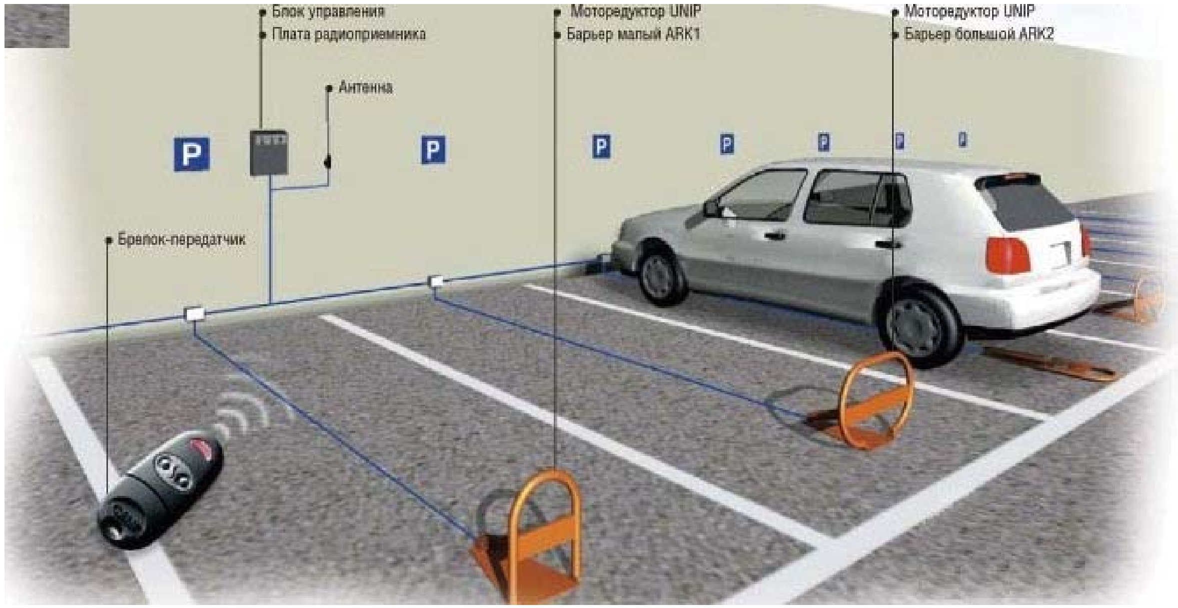
- Infrastructure-based solutions
- On-vehicle-sensor based solutions
  - Parking-vacancy detection
    - Ultrasonic radar
    - Stereo-vision
    - Depth camera



背景

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

our focus



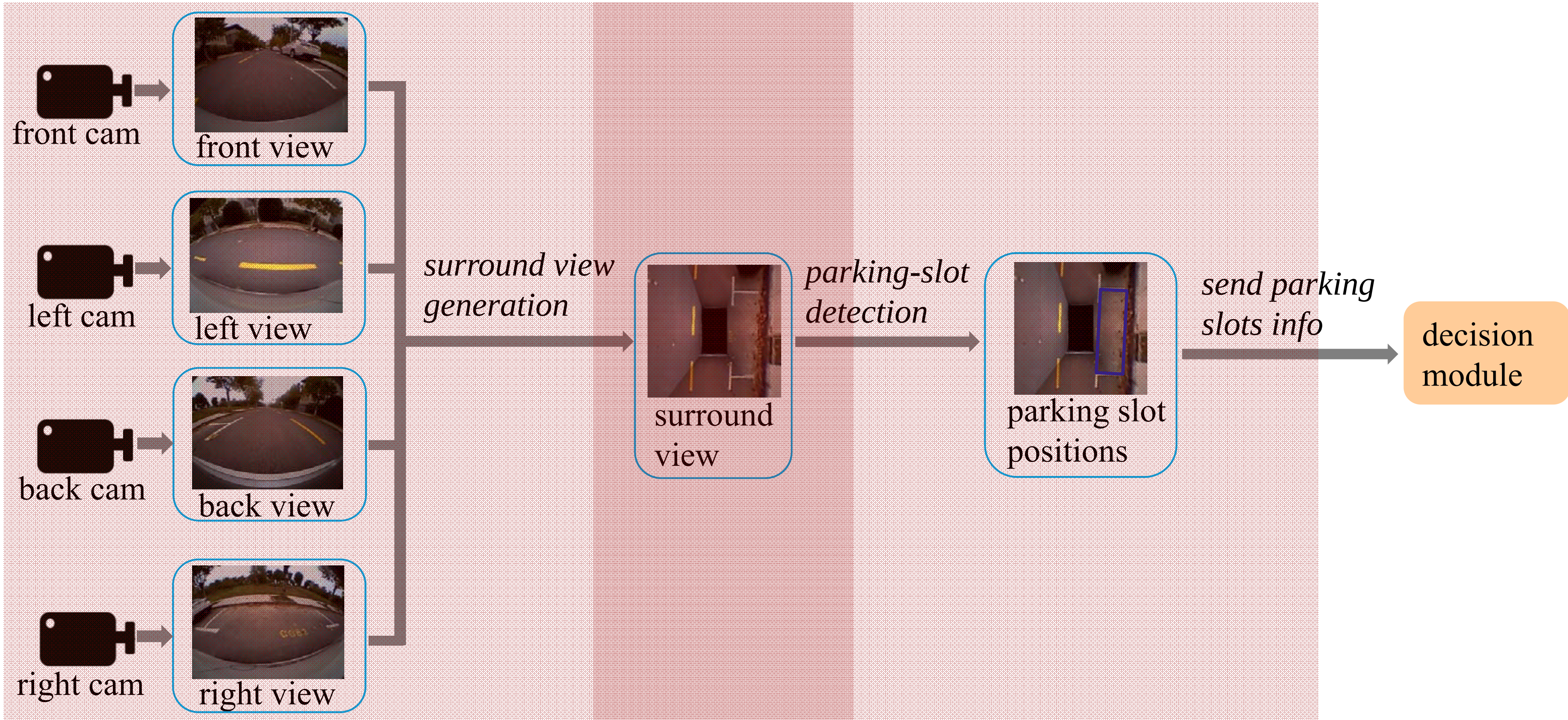
背景

- 研究现状的不足
  - 此领域没有公开数据集
  - 现有方法都是基于低层视觉特征的（边缘、角点、线等），鲁棒性和准确性都很有限
- 我们的贡献
  - ✓ 构建并公开了大规模带标注的环视图像数据集
  - ✓ 首次提出了基于机器学习理论的解决方案
  - ✓ 针对荣威E50车型开发实际系统并完成实车自主泊车系统验证

提纲

- 背景概述
- 泊车位检测与定位
  - 背景
  - 总体流程
  - 环视图
  - 泊车位检测算法
  - 实验
- 减速带与行人的检测与测距
- 嵌入式平台实现
- 病态曝光图像的复原

总体流程



基于视觉的自主泊车系统工作流程

提纲

- 背景概述
- 泊车位检测与定位
  - 背景
  - 总体流程
  - 环视图
  - 泊车位检测算法
  - 实验
- 减速带与行人的检测与测距
- 嵌入式平台实现
- 病态曝光图像的复原

环视图

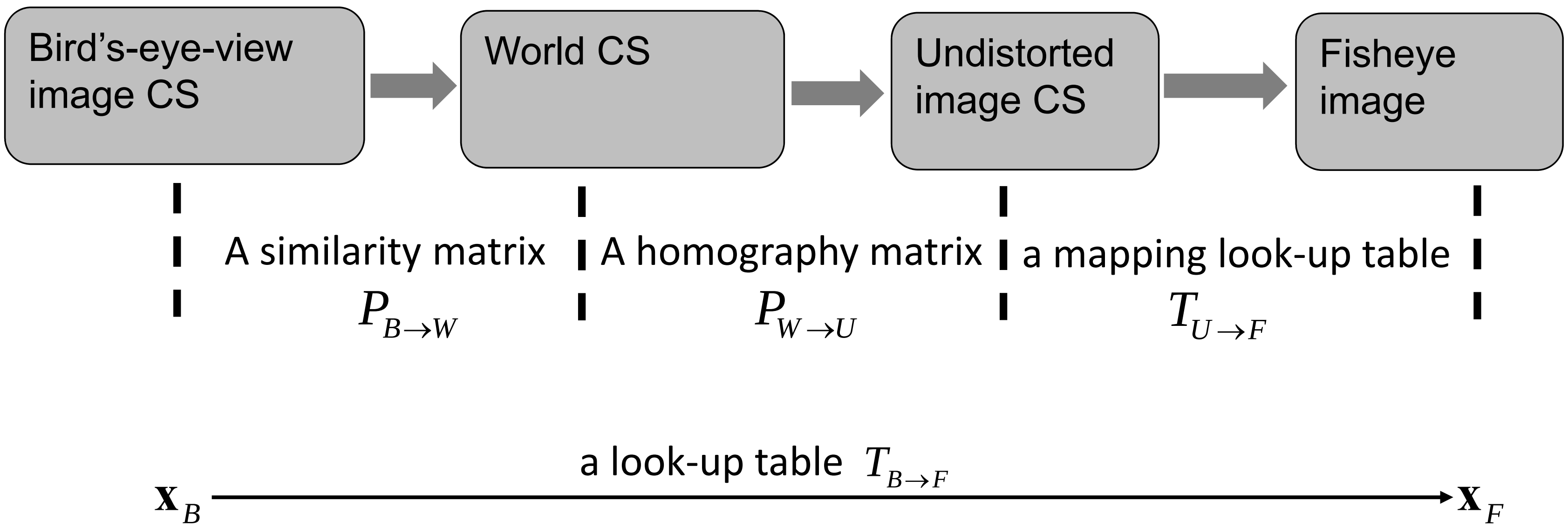
- Surround view camera system is an important ADAS technology allowing the driver to see a top-down view of the 360 degree surroundings of the vehicle
- Such a system normally consists of 4~6 wide-angle (fish-eye lens) cameras mounted around the vehicle, each facing a different direction

环视图

- 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 $P_{B \to W}$, mapping a point from the bird's-eye view coordinate system to the world coordinate system
  - Decide the projective transformation matrix $P_{W \to U}$, mapping a point from the world coordinate system to the undistorted image coordinate system
  - Decide the look-up table $T_{U \to F}$, mapping a point from the undistorted image coordinate system to the fish-eye image coordinate system

环视图

- 鸟瞰视图生成流程



环视图

- 鸟瞰视图生成流程
  - 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

环视图

- 鸟瞰视图生成流程
  - Determine $P_{W \to U}$

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

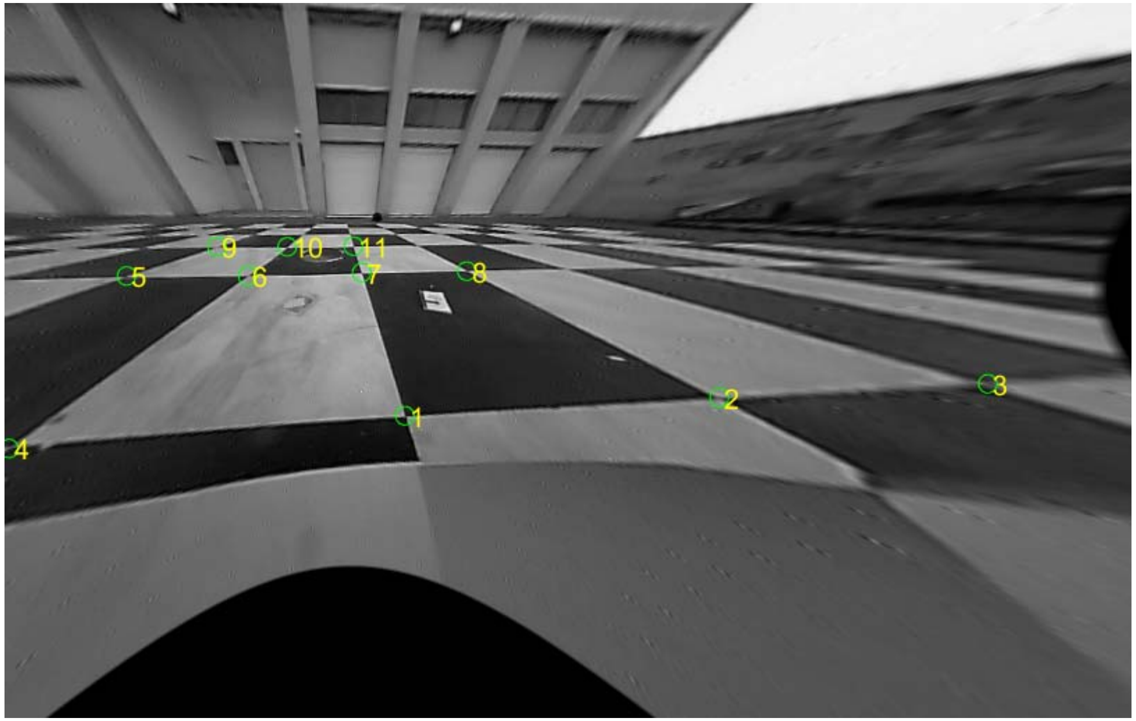
$$\mathbf{x}_U = P_{W \to U} \mathbf{x}_W$$

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

环视图

- 鸟瞰视图生成流程
  - Determine $P_{W \to U}$



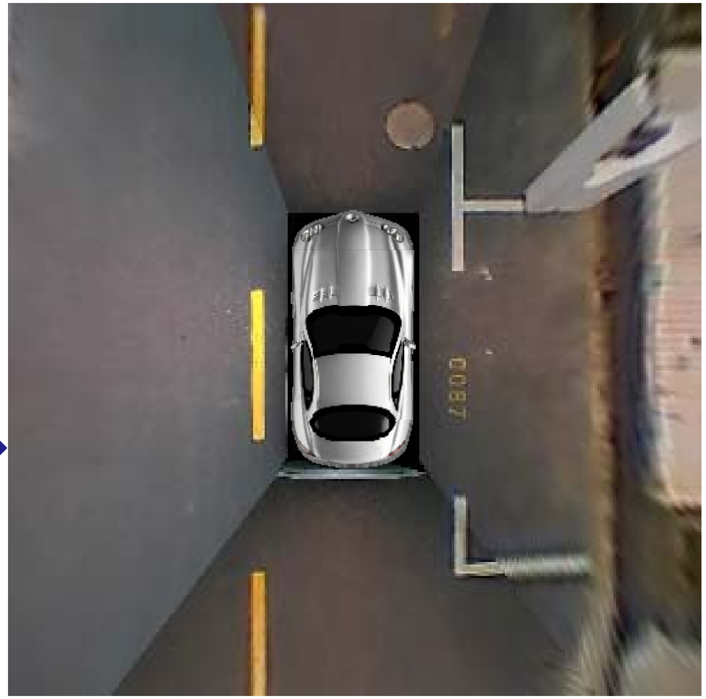
环视图

(a)

(b)

(c)

(d)

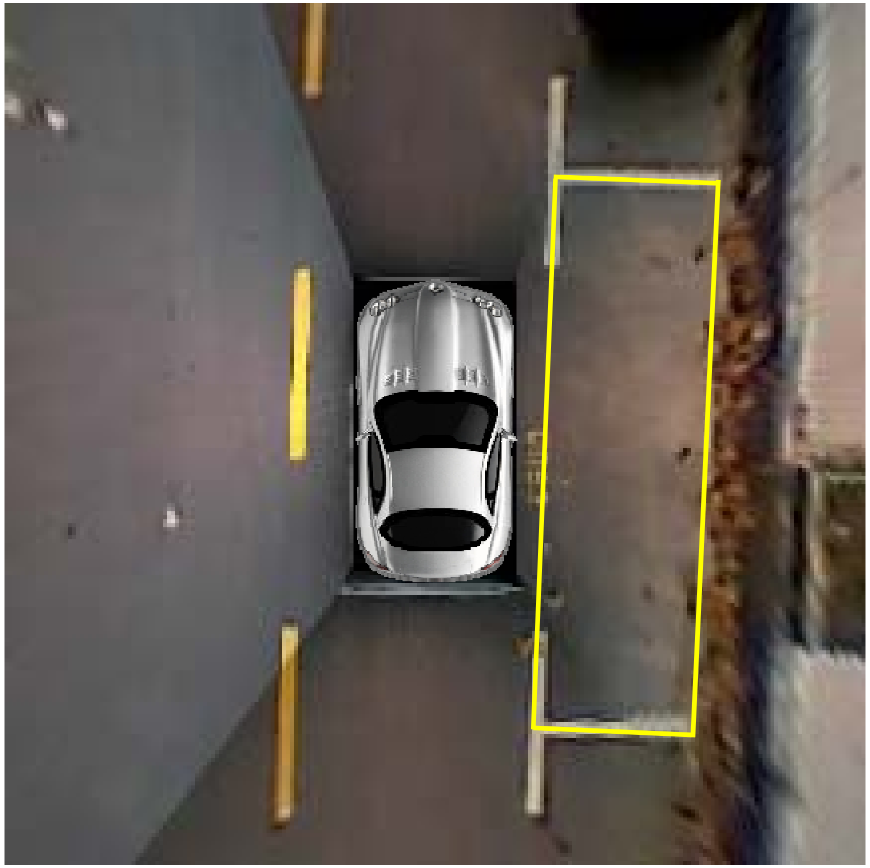


(e)

Image is of the size 600×600

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

环视图



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

提纲

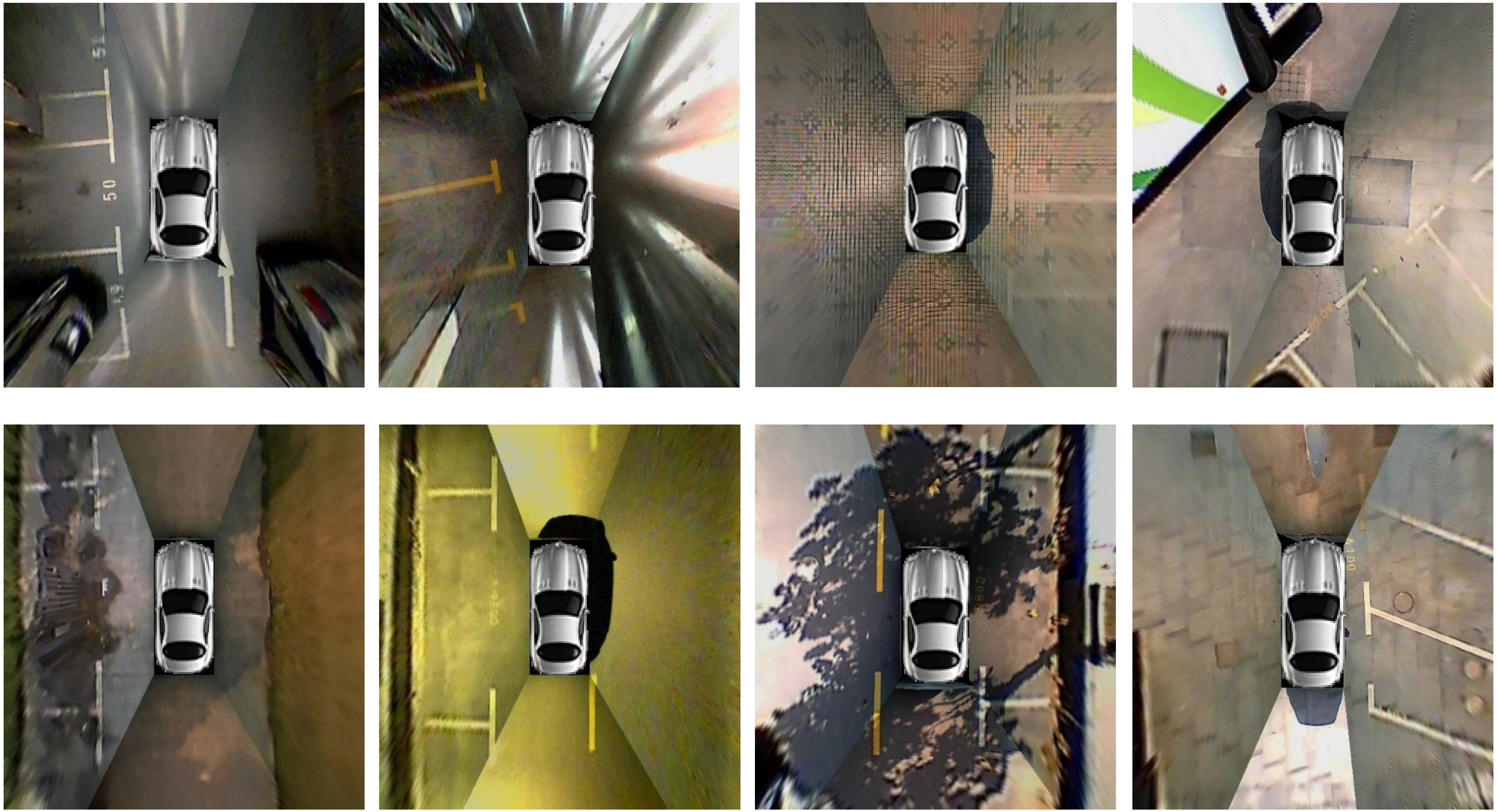
- 背景概述
- 泊车位检测与定位
  - 背景
  - 总体流程
  - 环视图
  - 泊车位检测算法
  - 实验
- 减速带与行人的检测与测距
- 嵌入式平台实现
- 病态曝光图像的复原

面临的挑战

- 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

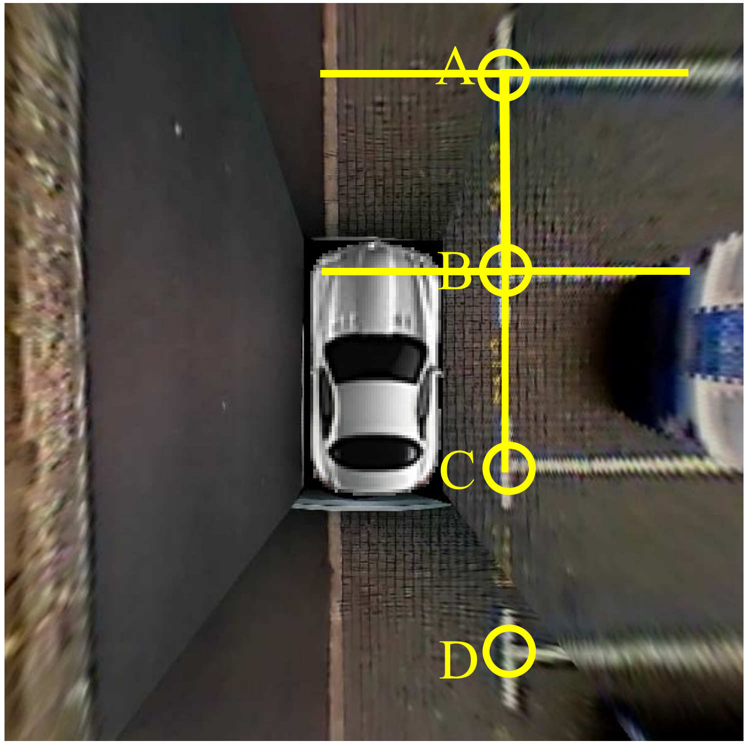
Making the low-level vision based algorithms difficult to succeed

面临的挑战



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)

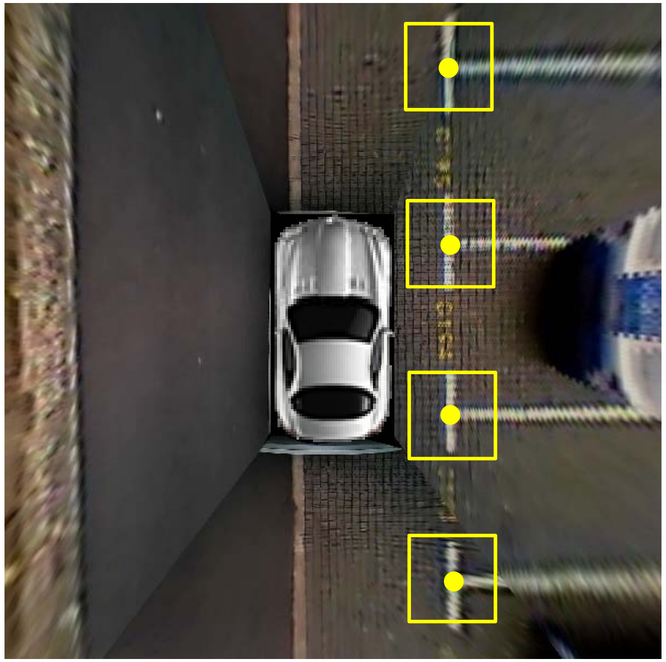
Both of them can be solved by DCNN-based techniques

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!*

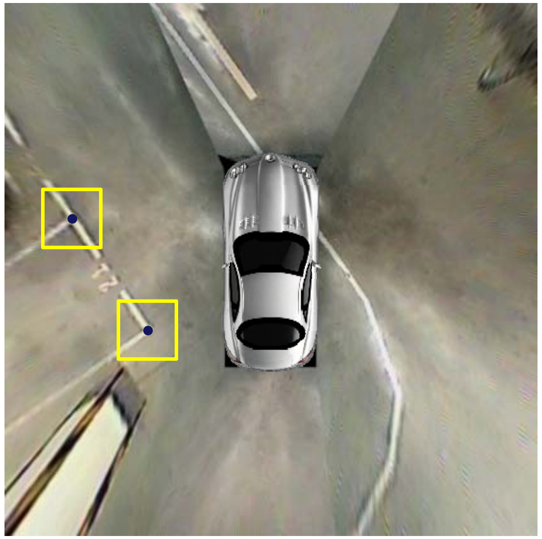
DeepPS: A DCNN-based Approach

- 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”



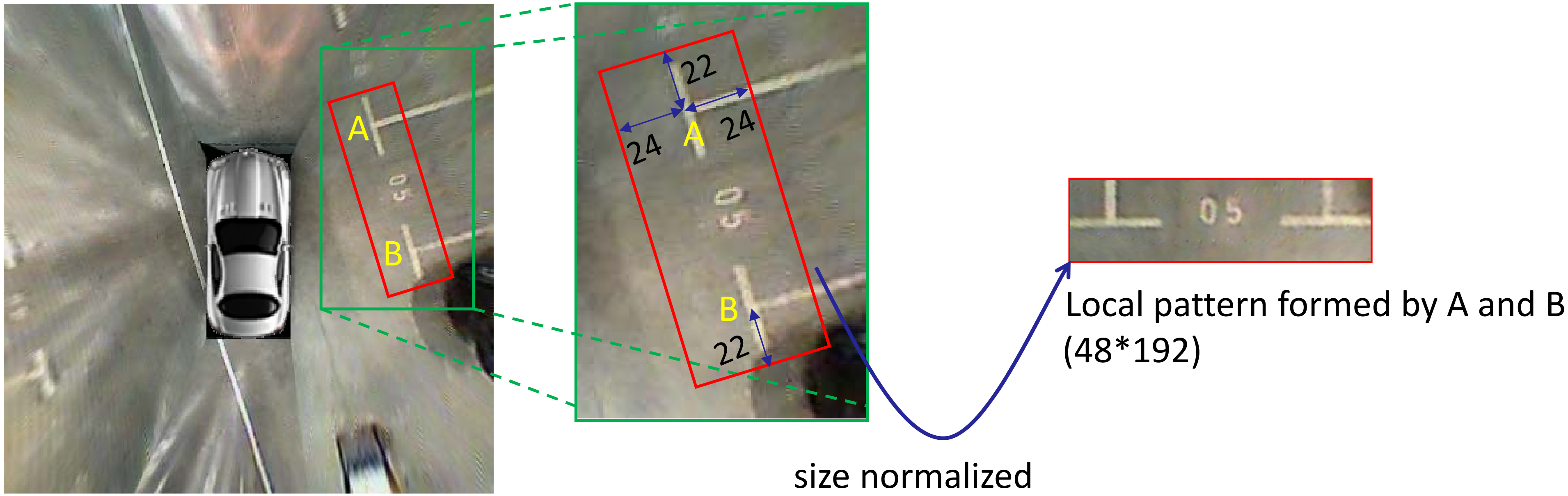
DeepPS: A DCNN-based Approach

- 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

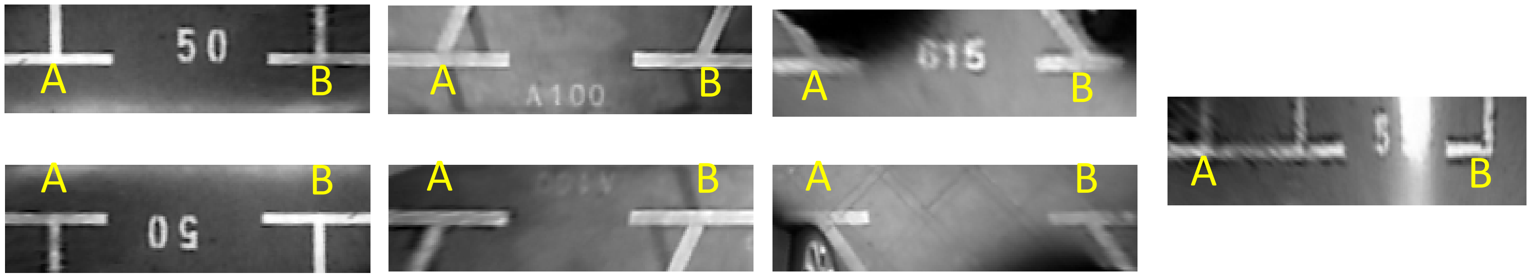
- Given two marking points A and B, classify the local pattern formed by A and B for two purposes
  - Judge whether “AB” is a valid entrance-line
  - If it is, decide the type of this entrance-line



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

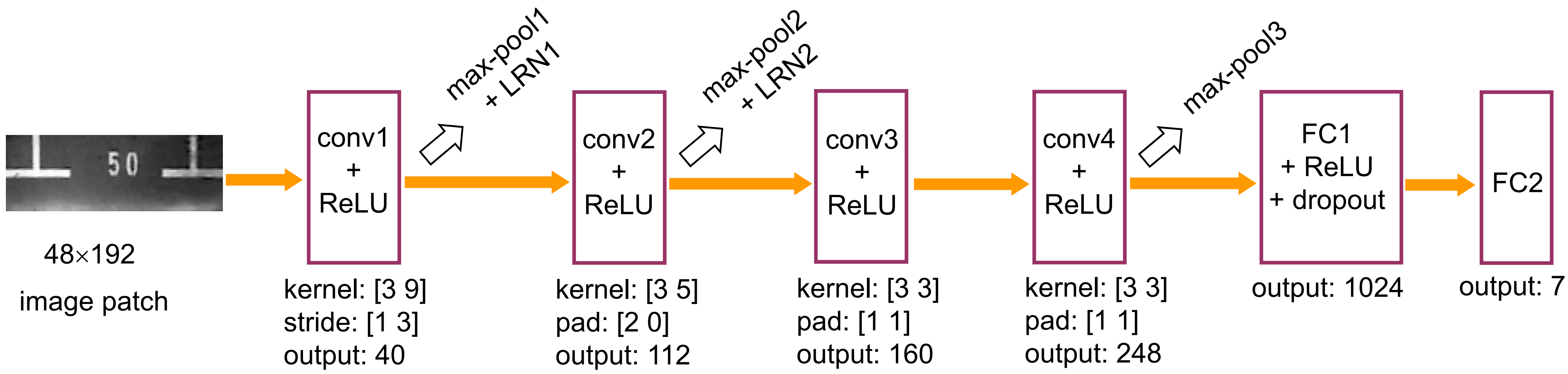
We define 7 types of local patterns formed by two marking-points



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

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

提纲

- 背景概述
- 泊车位检测与定位
  - 背景
  - 总体流程
  - 环视图
  - 泊车位检测算法
  - 实验
- 减速带与行人的检测与测距
- 嵌入式平台实现
- 病态曝光图像的复原

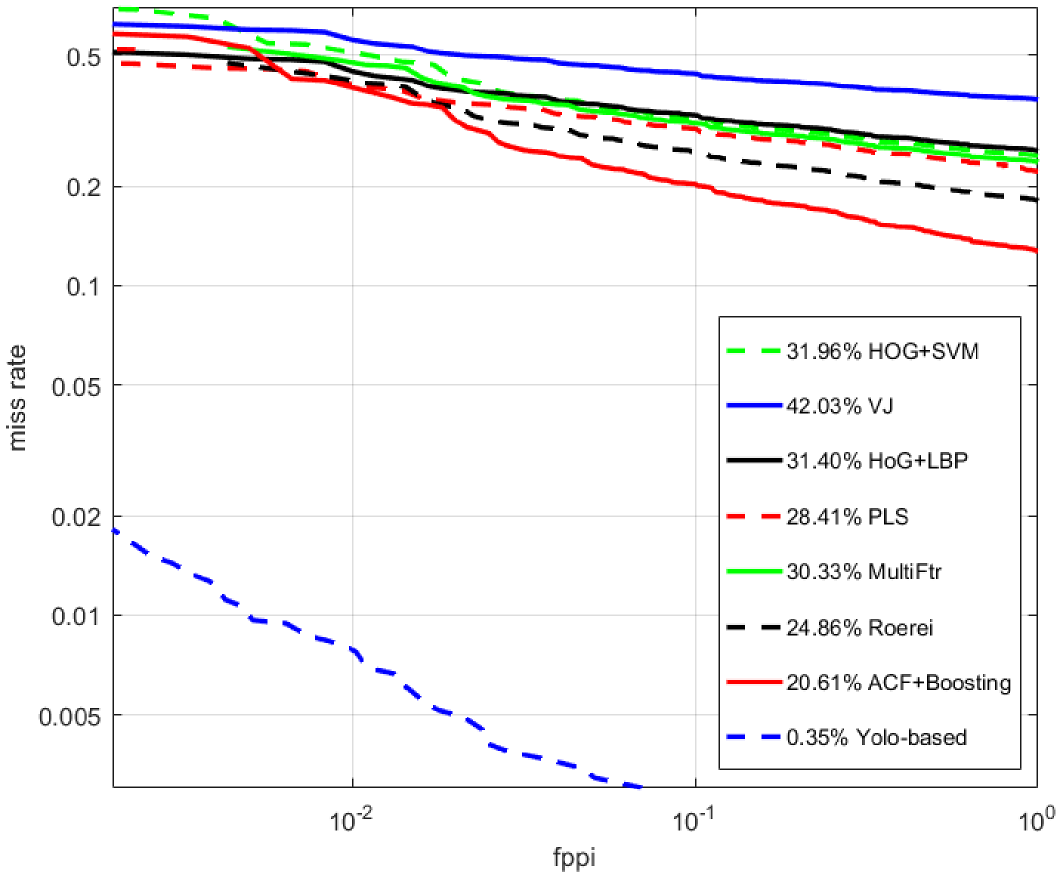
数据集

- 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

标志点检测准确性

- Missing rates VS FPPI curves on the entire test set



标志点定位准确性

- 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

泊车位检测准确性

- 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%**

泊车位检测准确性

- 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%)

基于视觉的泊车位检测

同济大学软件学院
计算视觉课题组

张林 李曦媛 黄君豪 李林申

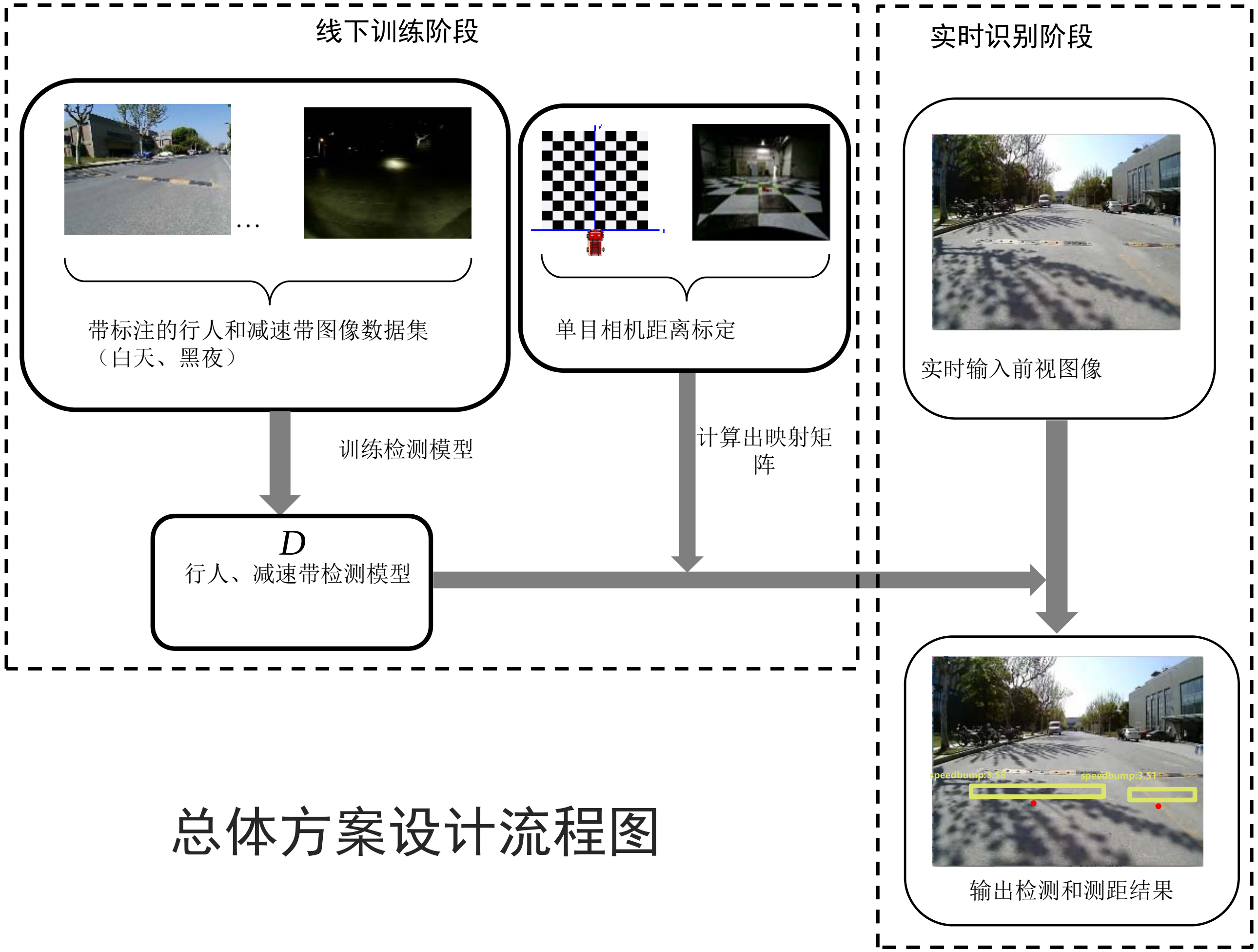
提纲

- 背景概述
- 泊车位检测与定位
- 减速带与行人的检测与测距
- 嵌入式平台实现
- 病态曝光图像的复原

减速带与行人的检测与测距

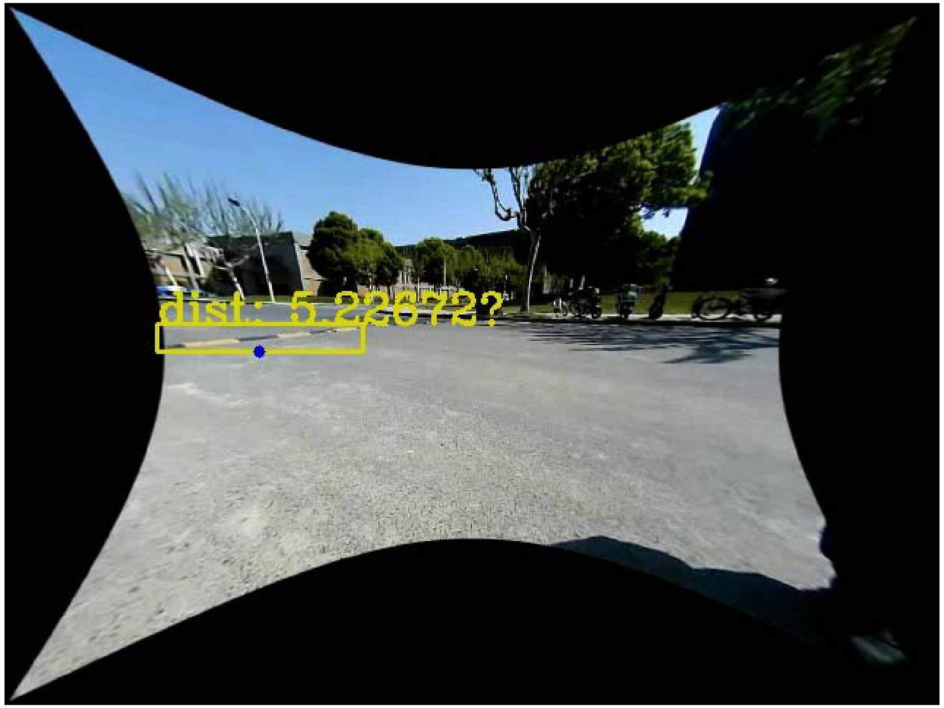
- 面向无人清扫车
- 基于单目视觉
- 实时检测到前方行人与减速带，并能够反馈其距离

减速带与行人的检测与测距



总体方案设计流程图

演示视频


dist: 5.22672?


面向智能清扫车的行人
及减速带检测与测距
同济大学软件学院
计算视觉课题组
张林 赵世雨 李曦媛 邵玄

提纲

- 背景概述
- 泊车位检测与定位
- 减速带与行人的检测与测距
- 嵌入式平台实现
- 病态曝光图像的复原

嵌入式平台实现

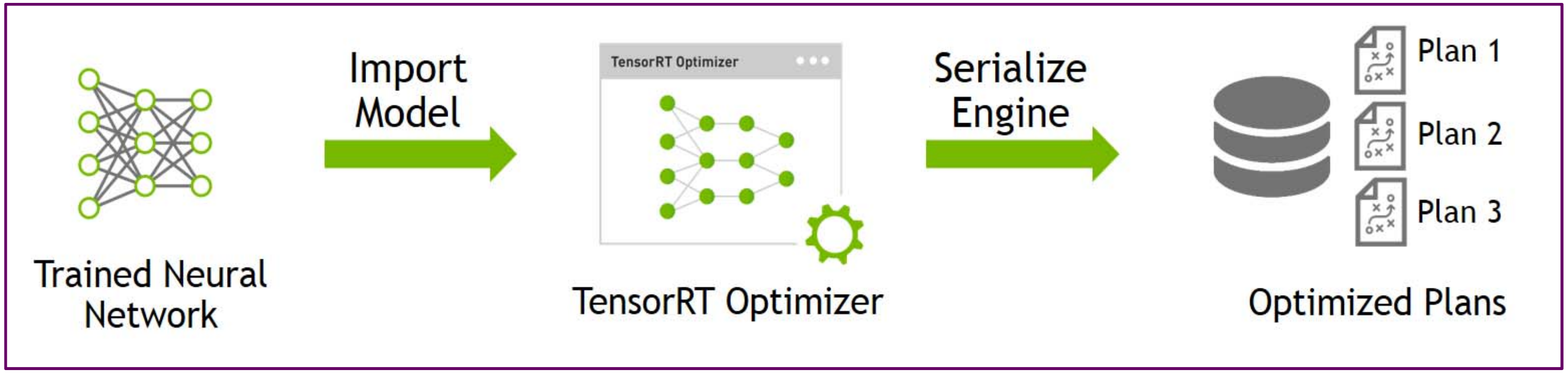
- 深度模型训练往往在workstation + 高端GPU上进行

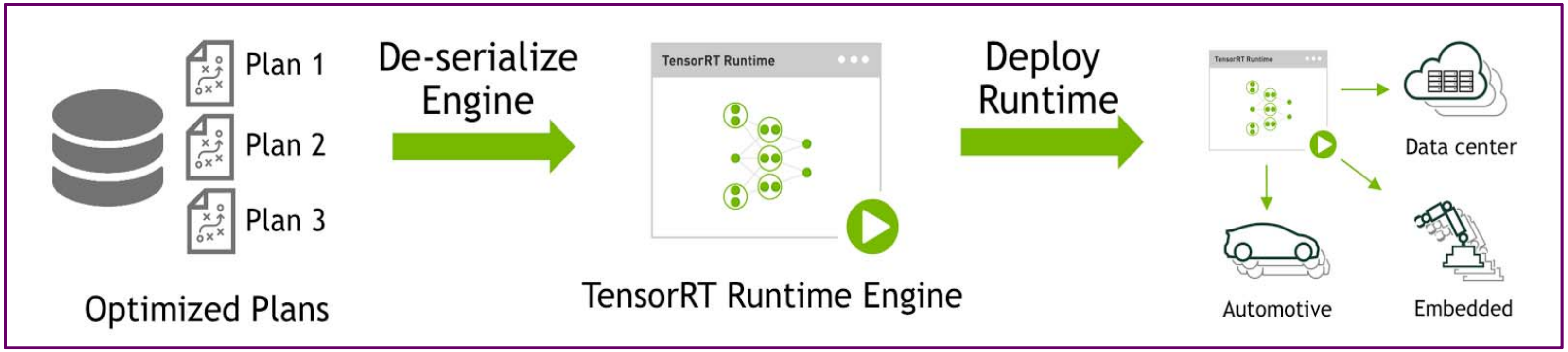
- 而客户终端的计算能力受限，比如Nvidia Jetson TX2

- 需要针对终端平台进行代码加速

嵌入式平台实现

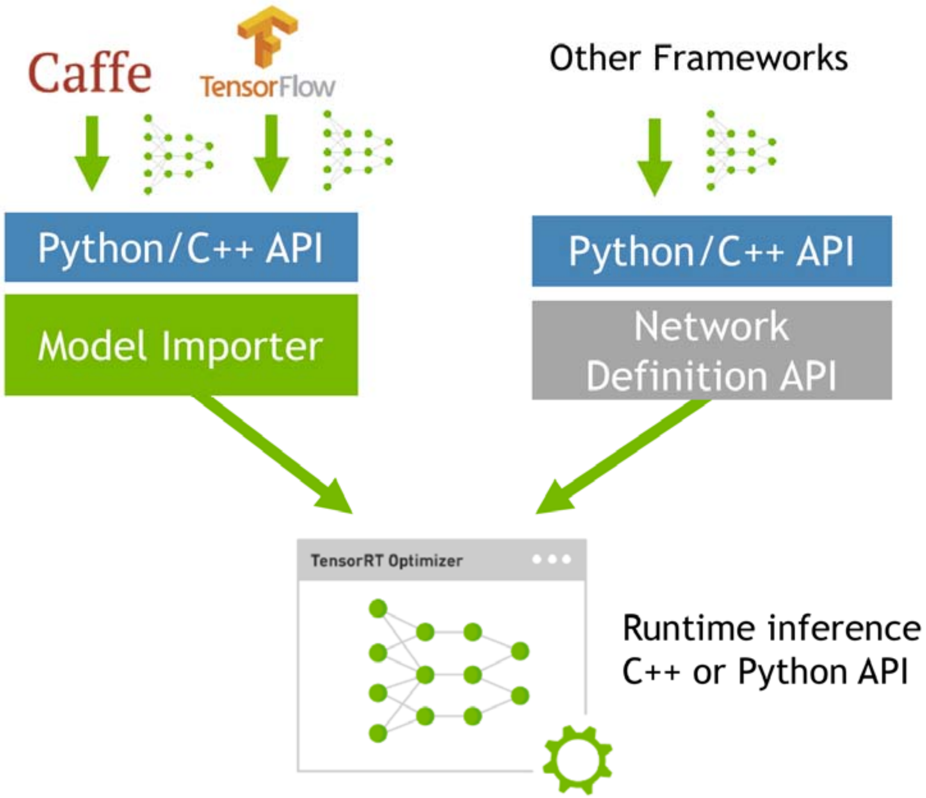
- 我们目前采用Nvidia Jetson TX2平台
- 用TensorRT作为运行时推断引擎





嵌入式平台实现

- 我们目前采用Nvidia Jetson TX2平台
- 用TensorRT作为运行时推断引擎



嵌入式平台实现

将Caffe-SSD模型转换为TensorRT推断引擎　　使用的TensortRT版本：3.0

TensorRT中不支持优化的网络层以Plugin Layer的形式参与转换

SSD中使用到的Plugin Layer:

1. DetectionOutput
2. PriorBox
3. Normalize
4. Permute
5. Concat (axis=2)

使用TensorRT提供的plugin API生成

6. Flatten
7. Reshape
8. Softmax (axis=2)

自主实现TensorRT的plugin Interface

嵌入式平台实现

Plugin Interface

接口名称	实现细节	调用阶段
getNbOutputs()	获取网络输出的Tensor的数量	网络定义阶段
getOutputDimensions()	获取输出Tensor的维度	网络定义阶段
configure()	设置网络层参数，调整算法	引擎构造阶段
getWorkspaceSize()	根据Batch Size设置GPU显存	引擎构造阶段
initialize()	运行初始化	执行环境初始化阶段
enqueue()	加入流执行队列	执行阶段
terminate()	结束网络层执行	执行环境终止阶段
getSerializationSize()	获取序列化空间大小	序列化阶段
serialize()	序列化	序列化阶段

嵌入式平台实现

- 最优秀目标检测算法在TX2上的表现

算法名称	未用TensorRT优化	用TensorRT优化
YoloV2 (416x416)	约160ms	-
VGG-SSD (300x300)	约189ms	约70ms
MobileNet-SSD (300x300)	约131ms	约22ms
Pelee-SSD (304x304)	约118ms	约20ms

提纲

- 背景概述
- 泊车位检测与定位
- 减速带与行人的检测与测距
- 嵌入式平台实现
- 病态曝光图像的复原

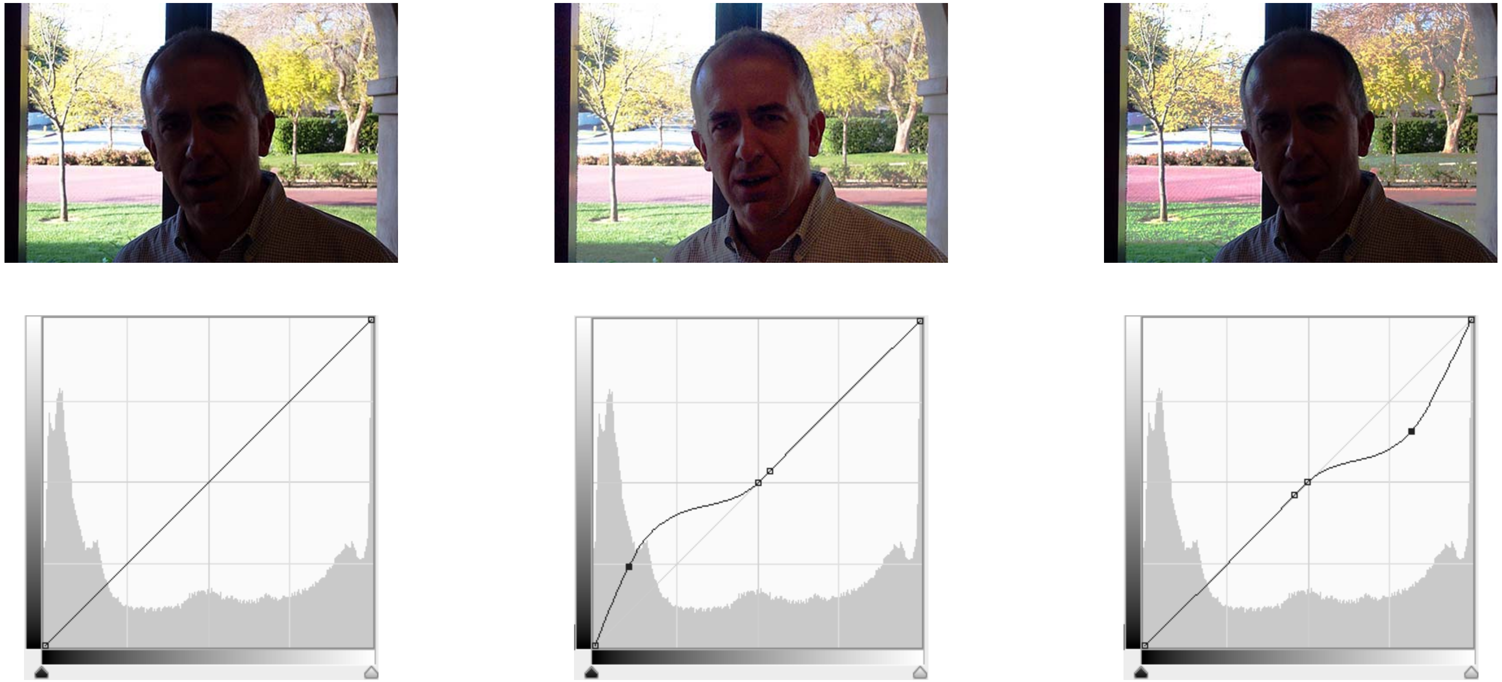
背景概述

- 逆光和暗光图像信号给视觉信息处理带来不便


Our goal

背景概述-相关方法

- 人工校正，比如photoshop调整图像的S-curve



背景概述-相关方法

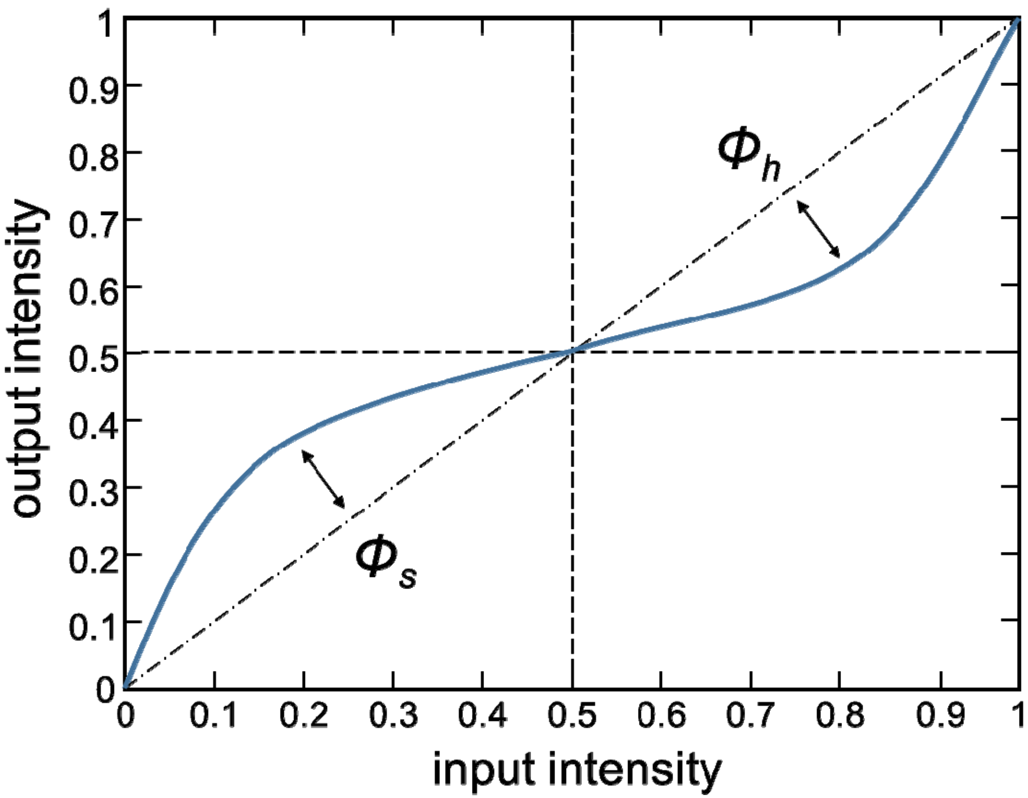
- 人工校正，比如photoshop调整图像的S-curve
- 自动化算法
  - 传统图像增加算法（HE，CLAHE，Retinex）：不针对背光图像校正问题，无法完全适用
  - 启发式算法：试图划分背光与迎光区域，然而手段粗糙不精确
  - 基于机器学习的算法：基于训练，需要大量数据

我们的方法：ExCNet

- EXCNet has the following merits
  - It is the first unsupervised CNN-based ill-exposed image restoration approach
  - It does not require pre-training and is image specific. Thus, it can be widely applicable to different shooting scenes and kinds of lighting conditions
- EXCNet's core idea
  - The optimal S-curve of the input image is estimated by using ExCNet, a CNN
  - Motivated by MRF, the loss function of ExCNet is designed as a block-based loss function, which tends to maximize the visibility of all blocks while keeping the relative difference between neighboring blocks
  - With the optimal S-curve, the input image can be restored straightforwardly

方法基础：S-curve

- S-curve：调整图像像素亮度，使之映射到曝光适宜的水平
  - 典型的S-curve：ϕ_s和ϕ_h分别是亮部和暗部的调节量



方法基础：S-curve

- S-curve：调整图像像素亮度，使之映射到曝光适宜的水平
  - 典型的S-curve：ϕ_s和ϕ_h分别是亮部和暗部的调节量
  - S-curve参数化：

$$f(x{:}\,\phi_s, \phi_h) = x + \phi_s \times f_\Delta(x) - \phi_h \times f_\Delta(1-x)$$

$$f_\Delta(t) = k_1 \cdot t \cdot \exp\left(-k_2 \cdot t^{k_3}\right) (k_1 = 5, k_2 = 14, k_3 = 1.6)$$

方法基础：S-curve

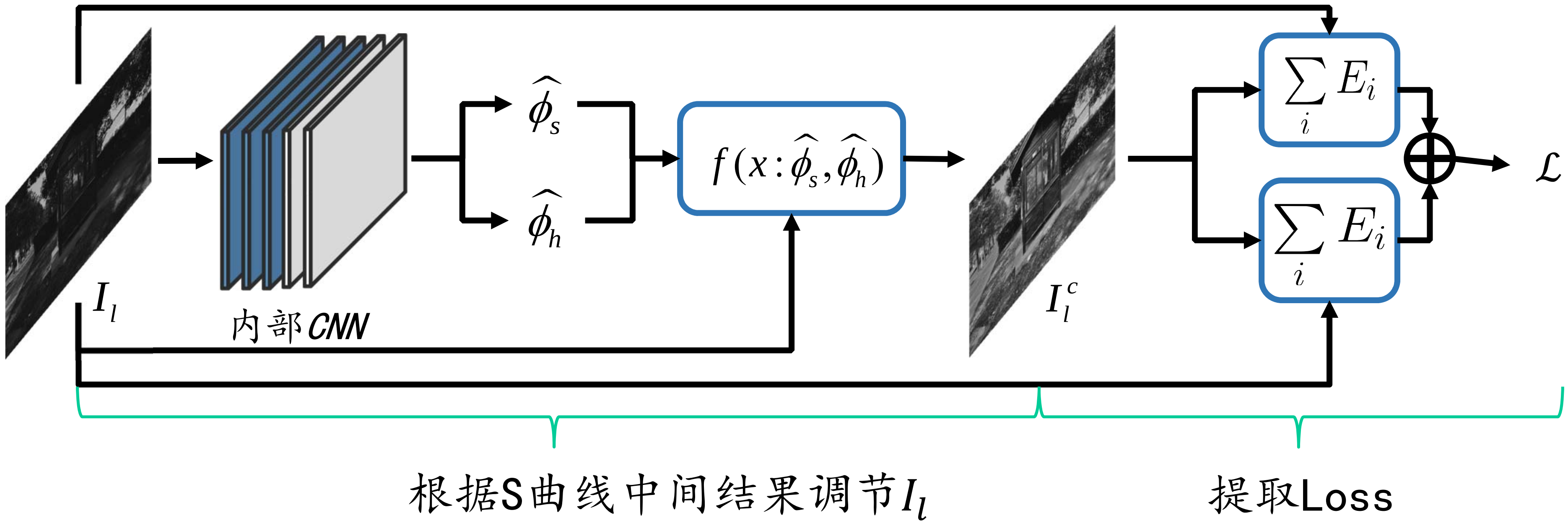
- S-curve：调整图像像素亮度，使之映射到曝光适宜的水平
  - 典型的S-curve：ϕ_s和ϕ_h分别是亮部和暗部的调节量
  - S-curve参数化：

$$f(x:\phi_s,\phi_h)=x+\phi_s\times f_\Delta(x)-\phi_h\times f_\Delta(1-x)$$

  - 最优S-curve：最优参数对$\{\phi_s^*,\phi_h^*\}$

方法核心：ExCNet

- 设计目的：对任意一张图像I，从其亮度通道I_l中估计出最优S-curve参数
- 网络框架：两部分



方法核心：ExCNet

- 设计目的：对任意一张图像I，从其亮度通道I_l中估计出最优S-curve参数
- 网络框架：两部分
- 架构核心：Loss函数--Block-based energy minimization problem

$$\mathcal{L} = \sum_i (E_i + \lambda \sum_{j \in \Omega(i)} E_{ij})$$

- E_i为一元项，E_{ij}为二元项，λ 为常数
- 最小化E_i可提高第i块的视觉细节；E_{ij}表示相邻两块i和j的相对对比度变化，第j块是第i块的4个相邻块
- 最小化loss函数可以在增加图像块的细节可见度的同时，尽可能保证相邻图像块之间的对比度

方法核心：ExCNet

- 设计目的：对任意一张图像I，从其亮度通道I_l中估计出最优S-curve参数
- 网络框架：两部分
- 架构核心：Loss函数---一元项E_i

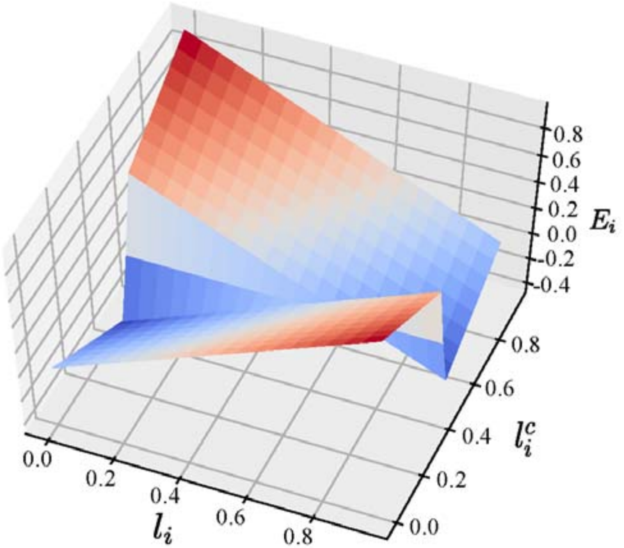
$$E_i = sign(l_i^c - 0.5) \cdot (l_i^c - l_i)$$

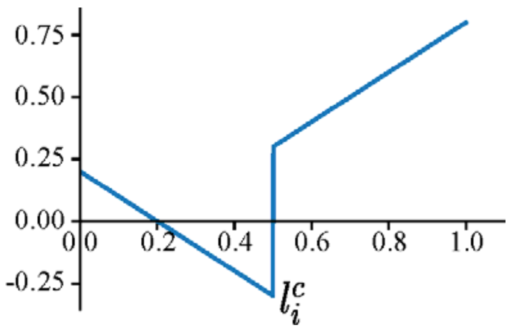
- l_i和l_i^c分别是原图I_i和校正后图像I_i^c第i块的平均亮度
- 使l_i^c尽可能接近0.5，以调整过曝和欠曝区域至曝光适宜
- 若l_i大于（小于）0.5，l_i^c也一定大于（小于）0.5，以使原亮区/暗区调整后仍然为亮区/暗区

方法核心：ExCNet

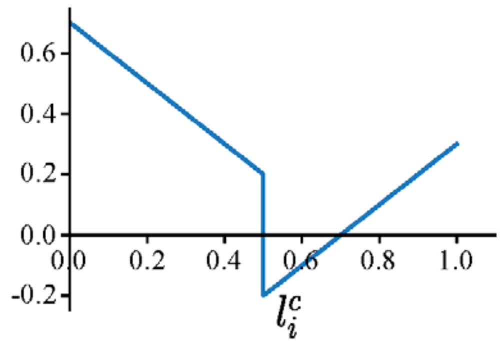
- 设计目的：对任意一张图像I，从其亮度通道I_l中估计出最优S-curve参数
- 网络框架：两部分
- 架构核心：Loss函数---一元项E_i

$$E_i = sign(l_i^c - 0.5) \cdot (l_i^c - l_i)$$





when l_i=0.2



when l_i=0.7

方法核心：ExCNet

- 设计目的：对任意一张图像I，从其亮度通道I_l中估计出最优S-curve参数
- 网络框架：两部分
- 架构核心：Loss函数--二元项E_{ij}

$$E_{ij} = ((l_j^c - l_i^c) - (l_j - l_i))^2$$

- l_j (l_j^c) 和l_i (l_i^c) 是块i和j的在原图I_i和校正后图像I_i^c中的平均亮度值
- 最小化二元项以保证相邻区域的相对对比度不变

病态曝光图像的复原

- 复原流程：
  - ExCNet估计待复原图像最优S-curve
  - 利用S-curve复原图像亮度通道
  - 对图像色彩通道成比例调整
- 复原代价：
  - 对于任意一张4032×3024分辨率的图像，在3.0GHZ Intel Core i7-5960X CPU 和 NVidia Titan X GPU工作站上，大约需要1.0s

图像复原后的对比度

- CDIQA is a no-reference quality metric for contrast-distorted images, which can be considered as a metric for richness of image details. Higher CDIQA value roughly corresponds to higher contrast.

Methods	CDIQA
HE	2.8757
CLAHE	3.0602
Retinex	3.2021
Picasa	3.0667
Wlsfilter	2.7608
Lapfilter	2.7790
Yuan and Sun	2.9451
Li and Wu	3.2494
ExCNet	**3.2616**

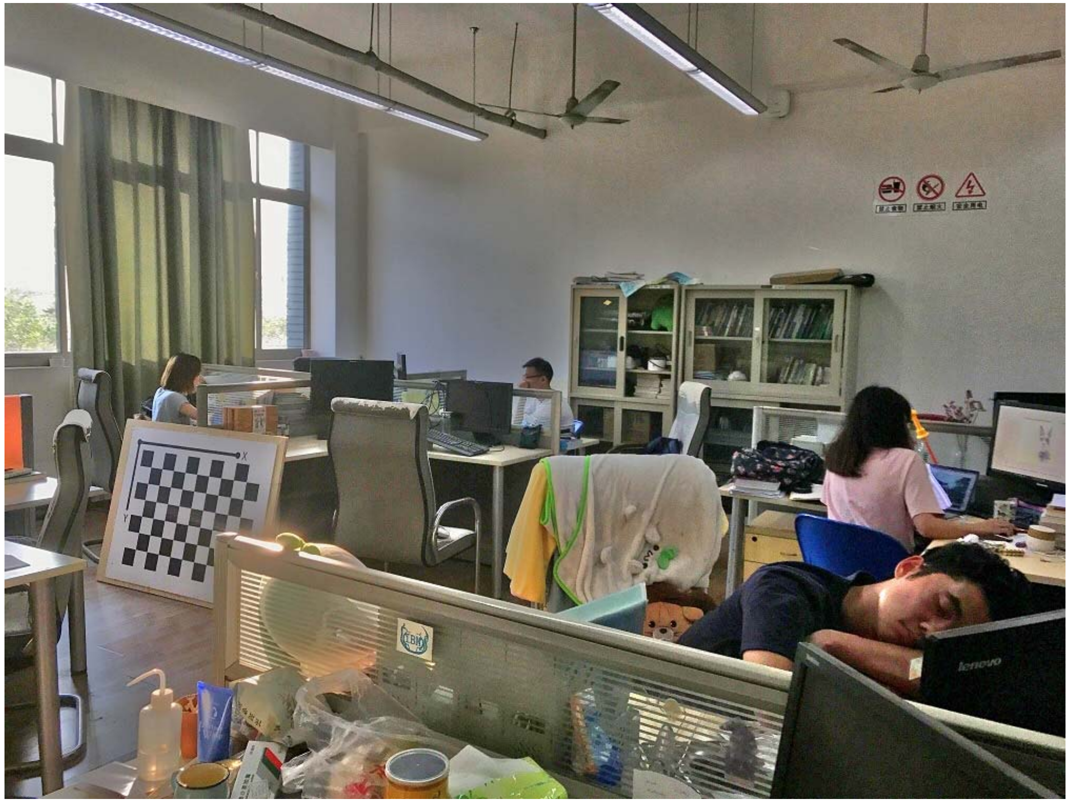
图像复原后的保序性

- Ideally, if the restoration approach does not violate the order statistics of pixel values of the input image, the associated LOD (luminance ordinal distortion) measure would be zero.

Methods	LOD
HE	4.4820
CLAHE	3.5214
Retinex	3.9602
Picasa	**2.2694**
Wlsfilter	3.7365
Lapfilter	5.0398
Yuan and Sun	4.6261
Li and Wu	4.9643
ExCNet	**2.8030**

示例

示例



示例

示例

示例

示例

示例

示例

视频展示

同济大学
软件学院

张 林

副教授，博士生导师
媒体艺术与科学教研室主任
CCF高级会员
IEEE Senior Member
CCF计算机视觉专委会委员

中国上海曹安公路4800号
同济大学济事楼408室 邮编：201804
电话：021-69589976 手机：13482179554
Email: cslinzhang@tongji.edu.cn
主页：http://sse.tongji.edu.cn/linzhang/

敬请批评指正!