

车道线检测

主 讲: Harry

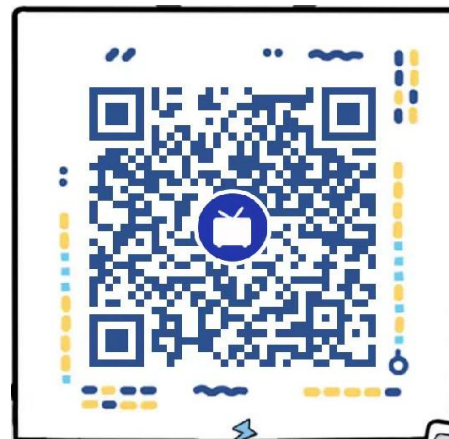
公众号: 自动驾驶之心

自动驾驶之心 全栈知识矩阵

自动驾驶之心，专注自动驾驶与AI



公众号
干货每日放送



B站
不定期直播+最新视频



知识星球
海量图文教程和Paper分享



视频号
技术视频一站式分享

主要内容

- ① 主讲人介绍
- ② 领域主流方法介绍
- ③ 个人工作/论文方法分享
- ④ 讨论与交流



主讲人介绍

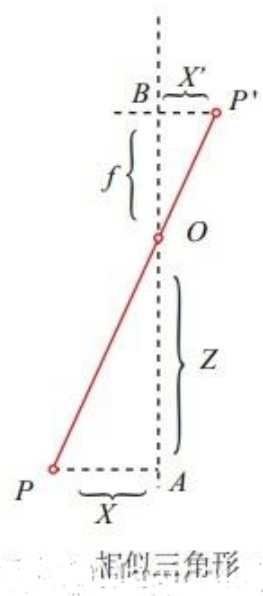
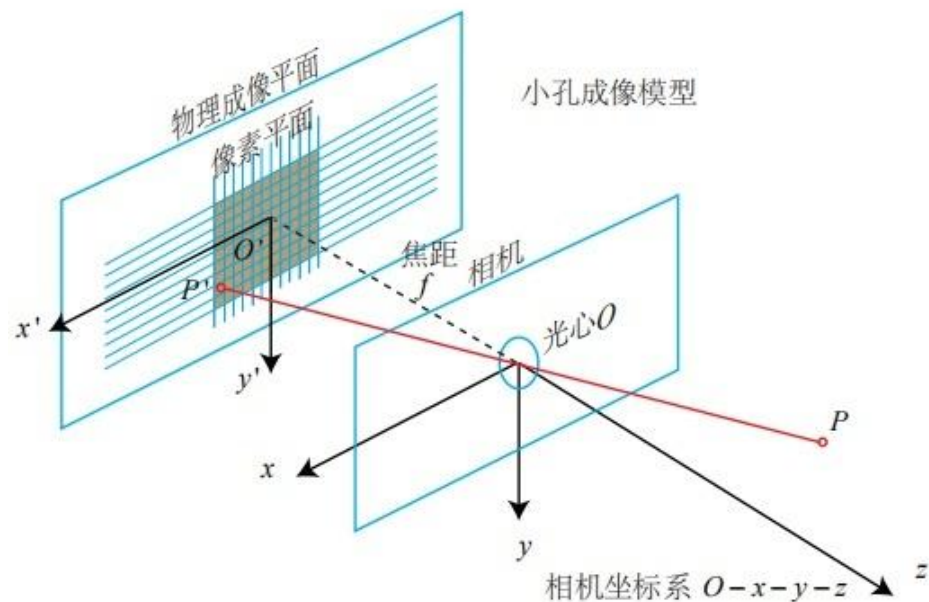
1. 毕业于国内某985
2. 目前主要研究方向：bev方向（地面元素检测，障碍物），nerf等
3. 研究成果：CVPR 一篇，AAAI两篇，SCI4篇
4. 分享内容简介： 车道线记





相机模型与IPM

相机模型 (相机内参)



$$\frac{Z}{f} = -\frac{X}{X'} = -\frac{Y}{Y'}.$$

$$\begin{cases} u = \alpha X' + c_x \\ v = \beta Y' + c_y \end{cases}.$$

将 $X' = f \frac{X}{Z}$ 和 $Y' = f \frac{Y}{Z}$ 带入, 并将 αf 合并为 f_x , βf 合并为 f_y 可得:

$$\begin{cases} u = f_x \frac{X}{Z} + c_x \\ v = f_y \frac{Y}{Z} + c_y \end{cases}.$$

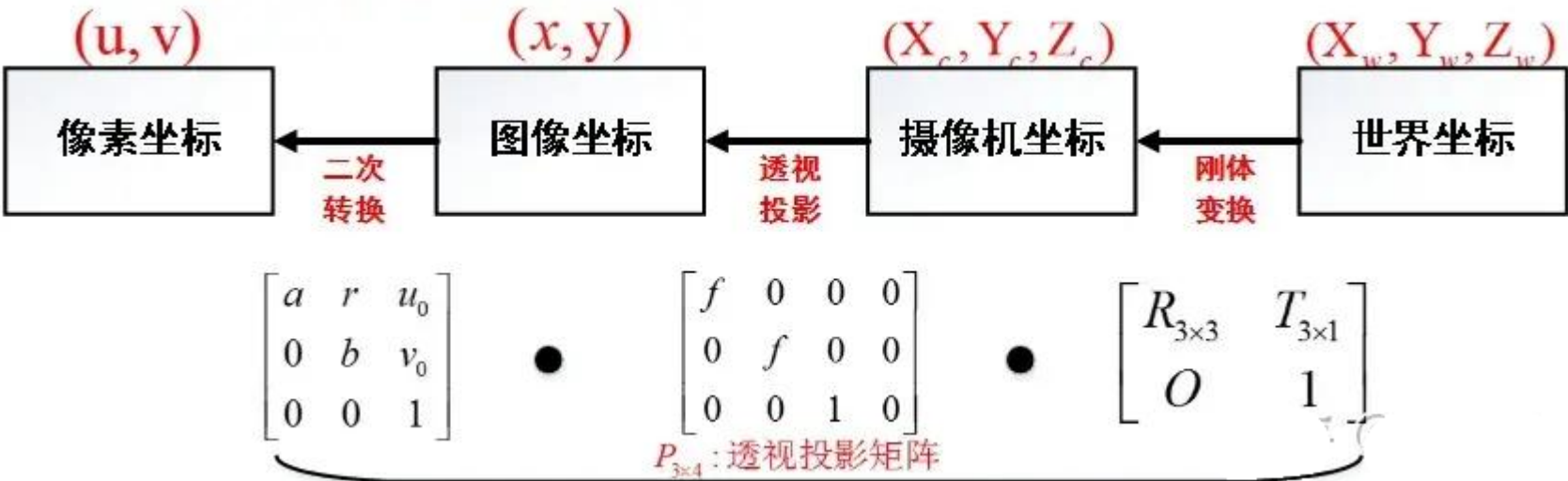
f 的单位为米, α, β 的单位为像素/米, 所以 f_x, f_y 的单位为像素。

利用齐次坐标可写为矩阵形式:

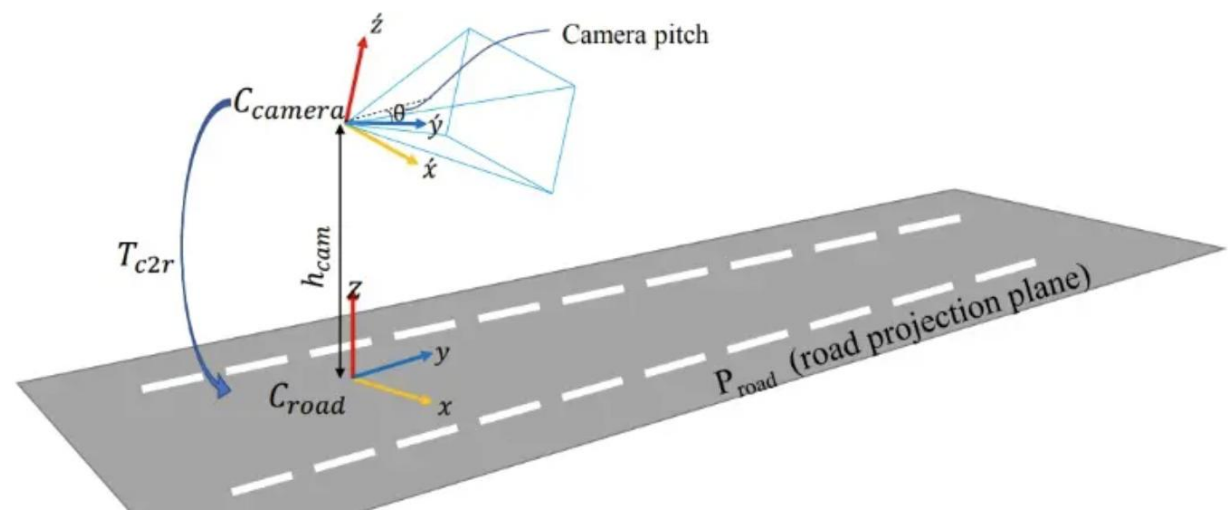
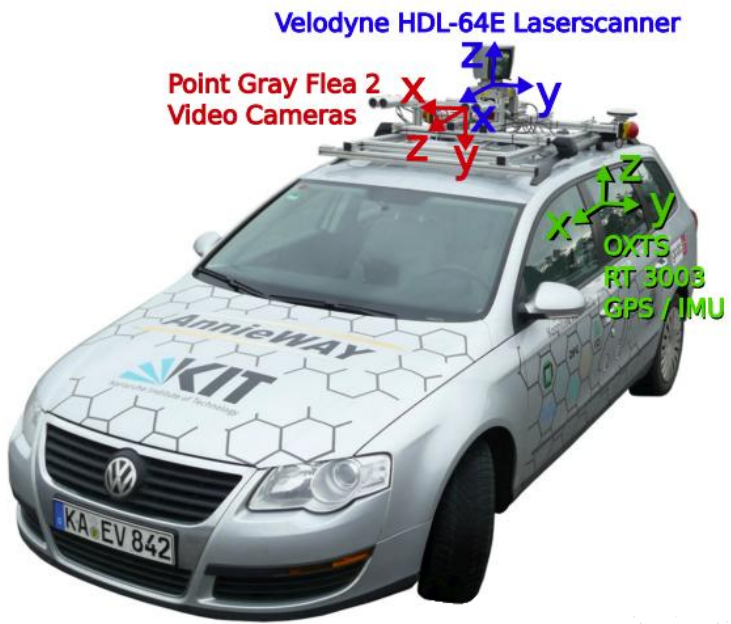
$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \triangleq \frac{1}{Z} K P.$$



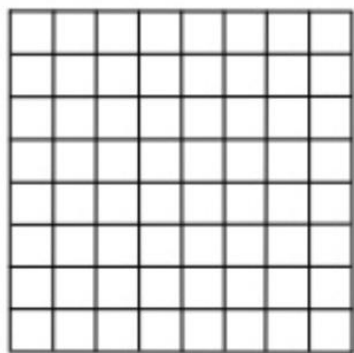
相机模型 (相机内外参)



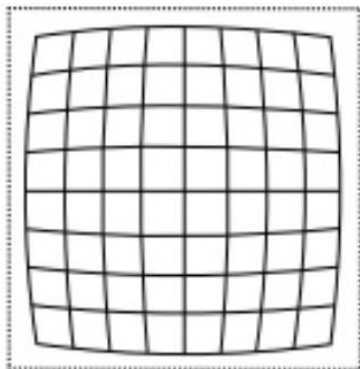
$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K(R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T)$$



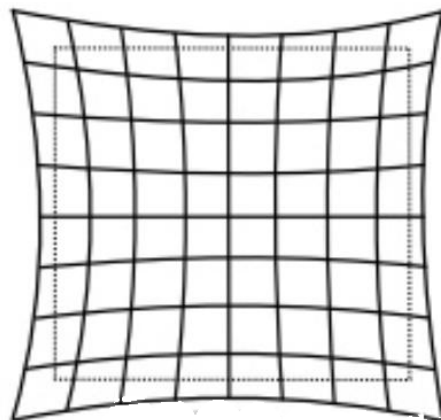
相机模型（相机标定简介）



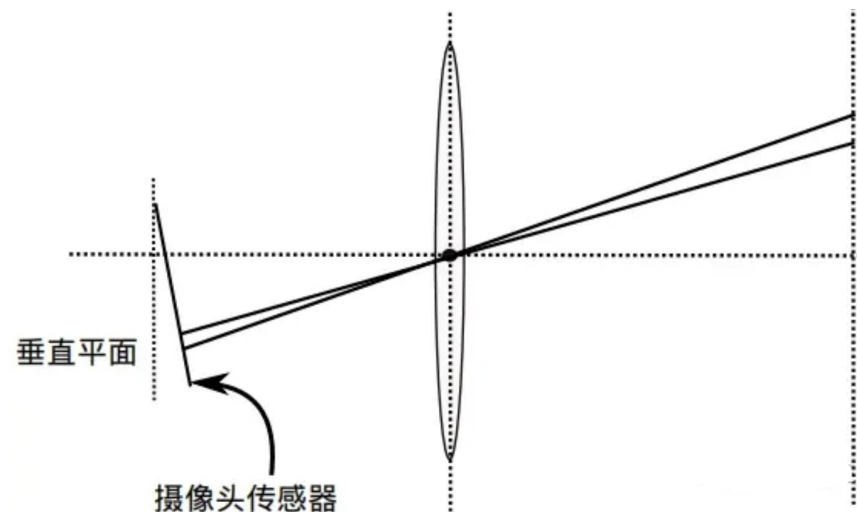
正常图像



桶形失真



枕形失真



$$x_{corrected} = x + x(k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$y_{corrected} = y + y(k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$x_{corrected} = x + 2p_1 xy + p_2(r^2 + 2x^2)$$

$$y_{corrected} = y + 2p_2 xy + p_1(r^2 + 2y^2)$$

$$x_{corrected} = x + x(k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 xy + p_2(r^2 + 2x^2)$$

$$y_{corrected} = y + y(k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_2 xy + p_1(r^2 + 2y^2)$$

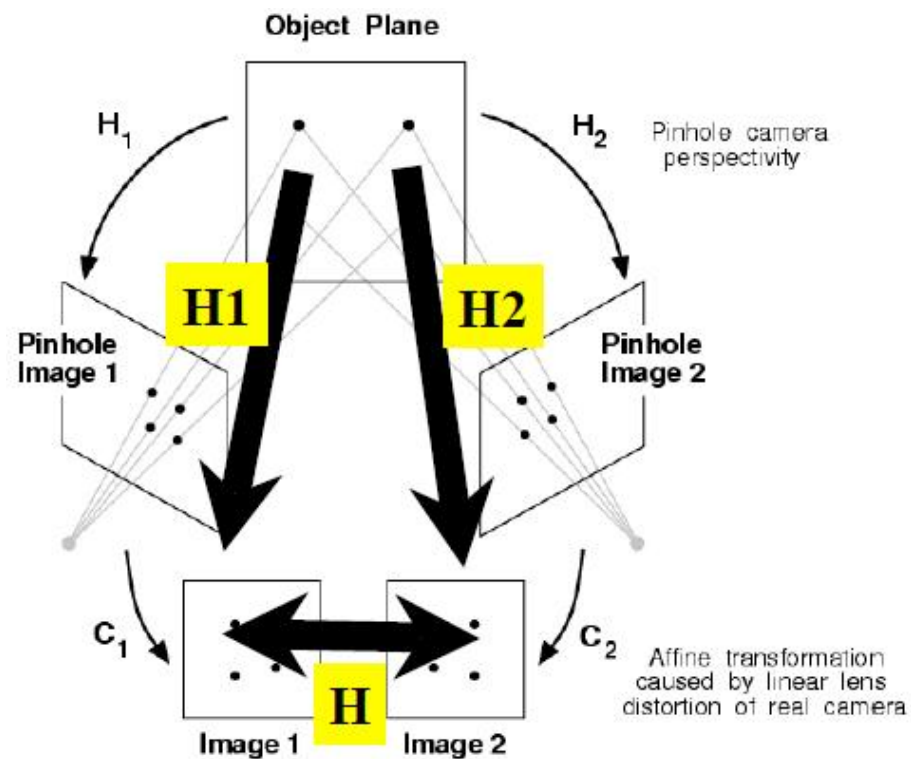


相机模型（相机标定简介）

标定方法	优点	缺点	常用方法
传统相机标定法	可使用于任意的相机模型、精度高	需要标定物、算法复杂	Tsai两步法、张氏标定法
主动视觉相机标定法	不需要标定物、算法简单、鲁棒性高	成本高、设备昂贵	主动系统控制相机做特定运动
相机自标定法	灵活性强、可在线标定	精度低、鲁棒性差	分层逐步标定、基于Kruppa方程



IPM (透视变换)



$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \leftarrow$$

$$\begin{cases} X' = \frac{X}{Z} \\ Y' = \frac{Y}{Z} \\ Z' = \frac{Z}{Z} \end{cases}$$

$$\begin{cases} X' = \frac{a_{11}x + a_{12}y + a_{13}}{a_{31}x + a_{32}y + a_{33}} \\ Y' = \frac{a_{21}x + a_{22}y + a_{23}}{a_{31}x + a_{32}y + a_{33}} \\ Z' = 1 \end{cases} \quad \leftarrow$$

```
pts1 = np.float32([[50,50],[200,50],[50,200]])
```

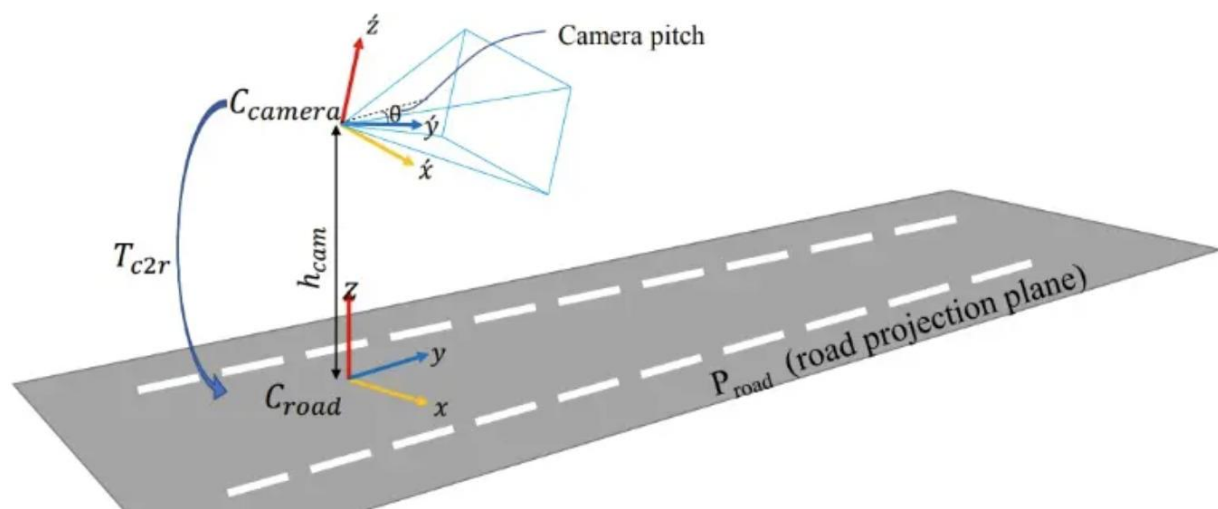
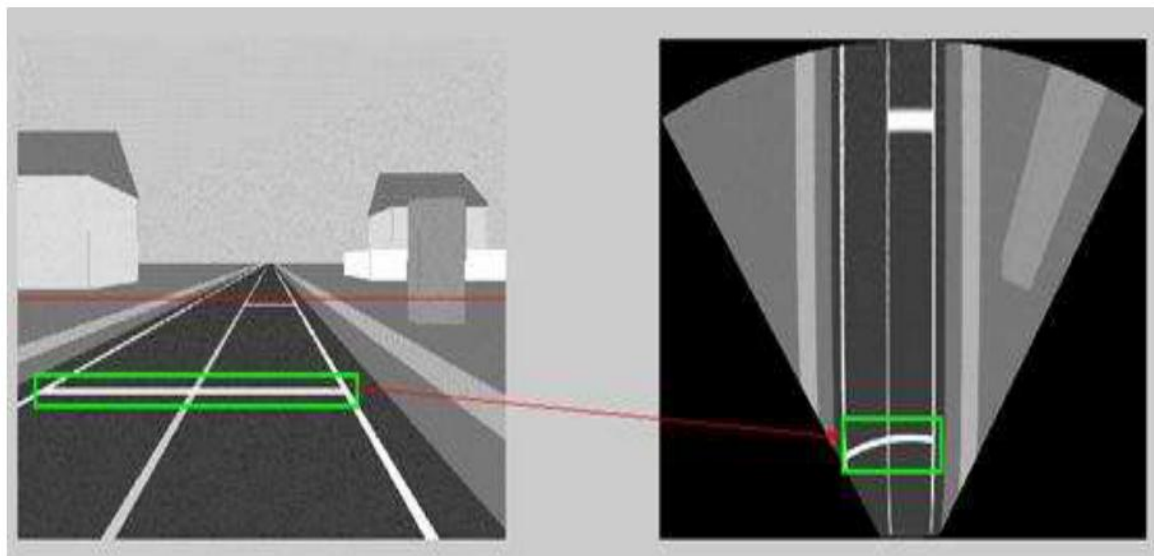
```
pts2 = np.float32([[10,100],[200,50],[100,250]])
```

```
M = cv2.getAffineTransform(pts1,pts2)
```

```
dst = cv2.warpAffine(img,M,(cols,rows))
```



IPM (逆透视变换) -- 地平假说



$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K(R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T)$$

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = R^{-1} \left(K^{-1} Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - T \right) = R^{-1} K^{-1} Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - R^{-1} T$$

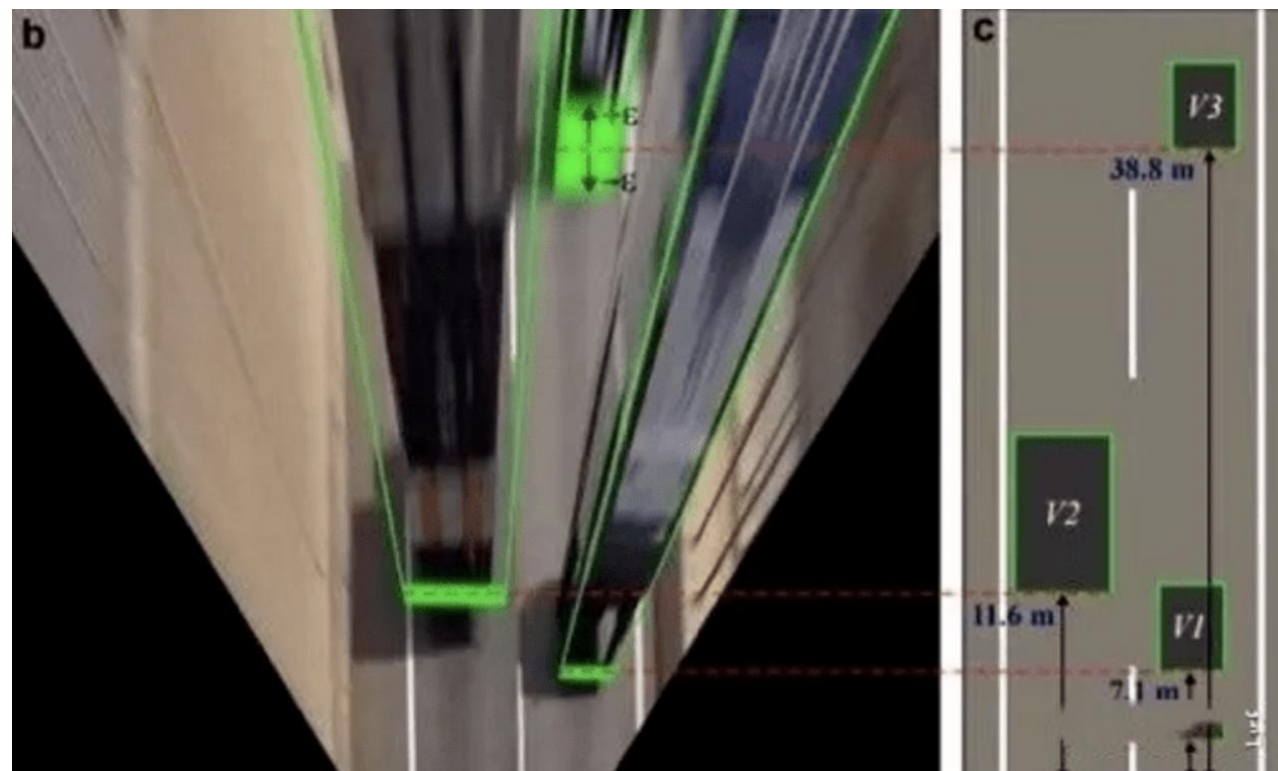
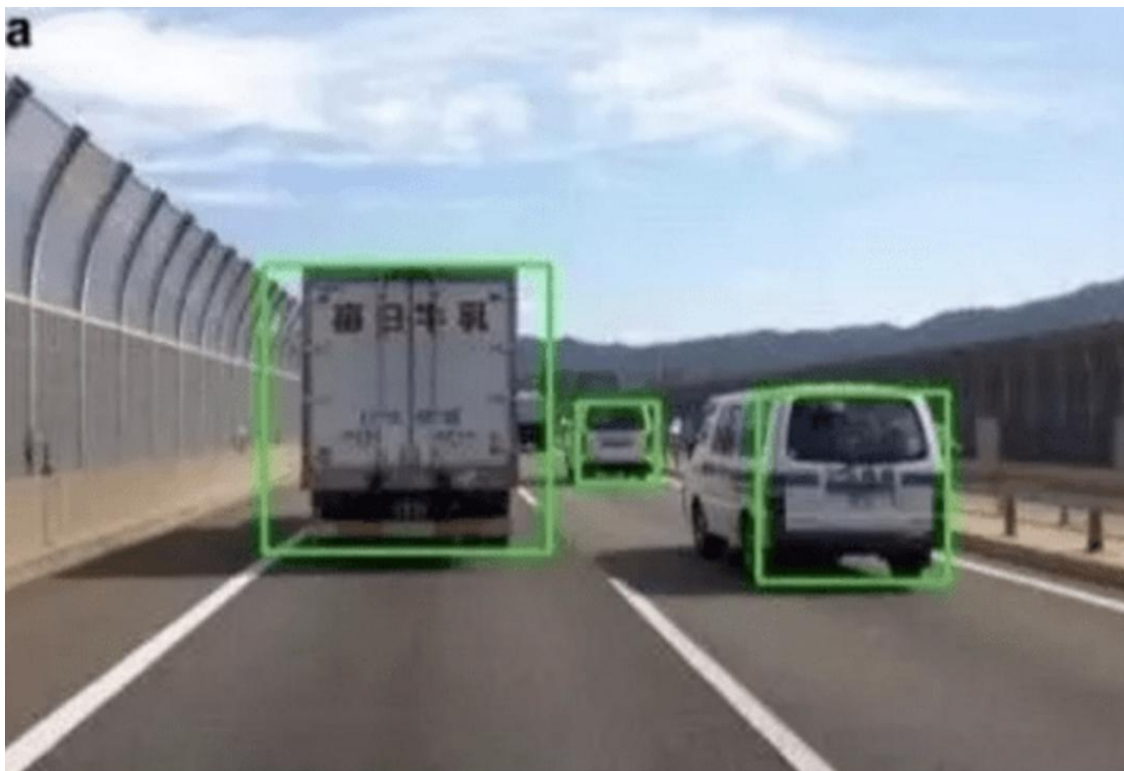
$$Mat1 = R^{-1} K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, Mat2 = R^{-1} T$$

$$Z_w = Z_c * Mat1(2,0) - Mat2(2,0)$$

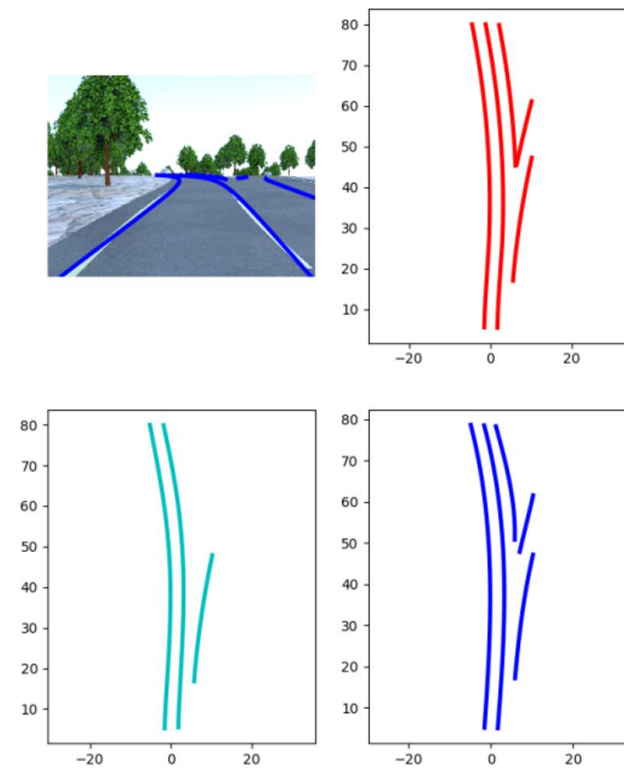
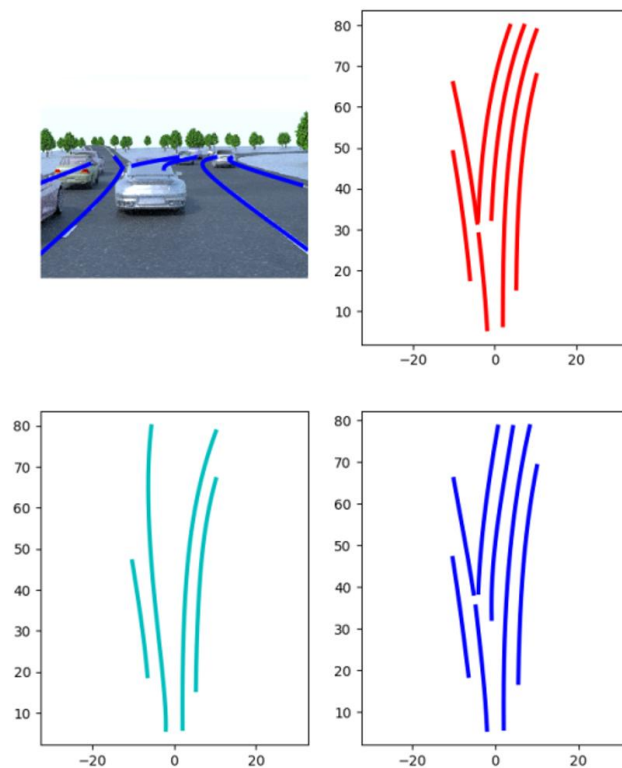
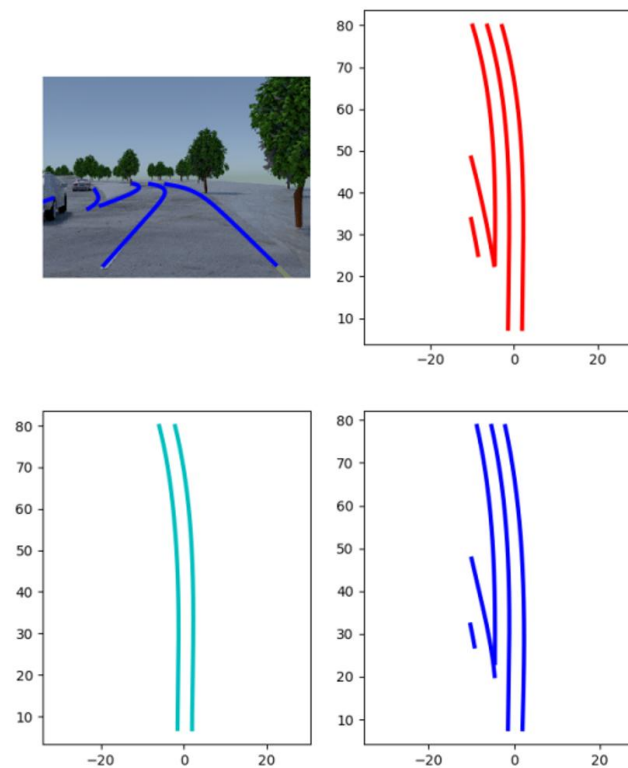
$$Z_c = (Z_w + Mat2(2,0)) / Mat1(2,0)$$



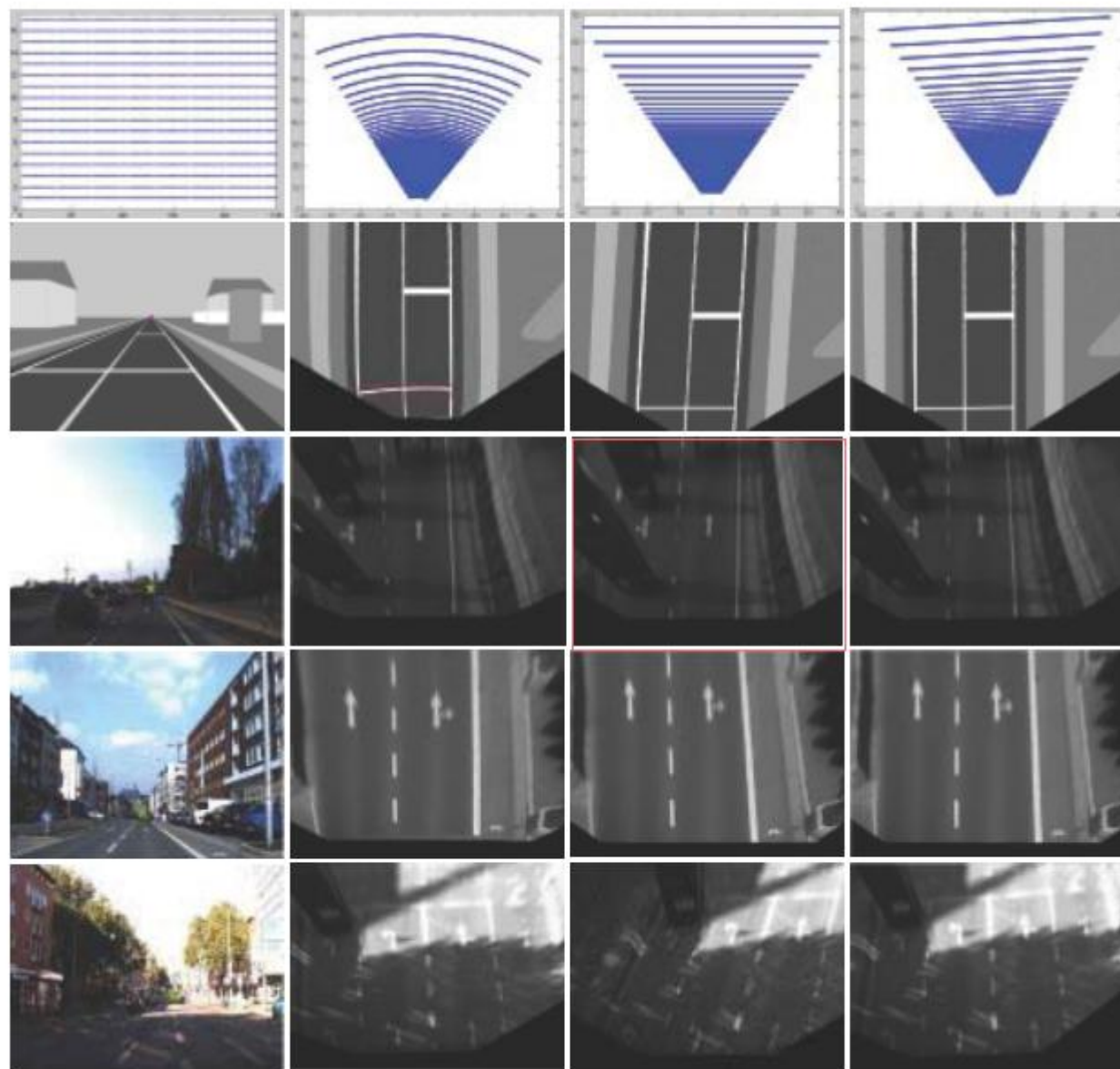
IPM (逆透视变换) -- 地平假说



IPM (线透视变换)



IPM (图像透视变换)

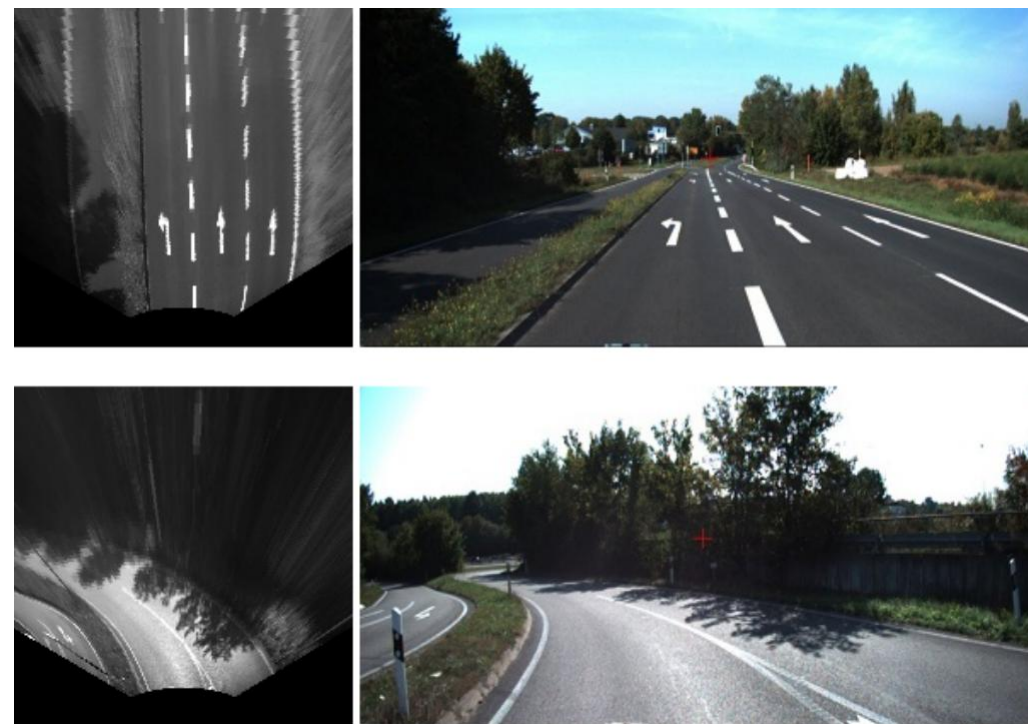


a) Vanishing point

b) Broggi

c) Eric

c) Proposed

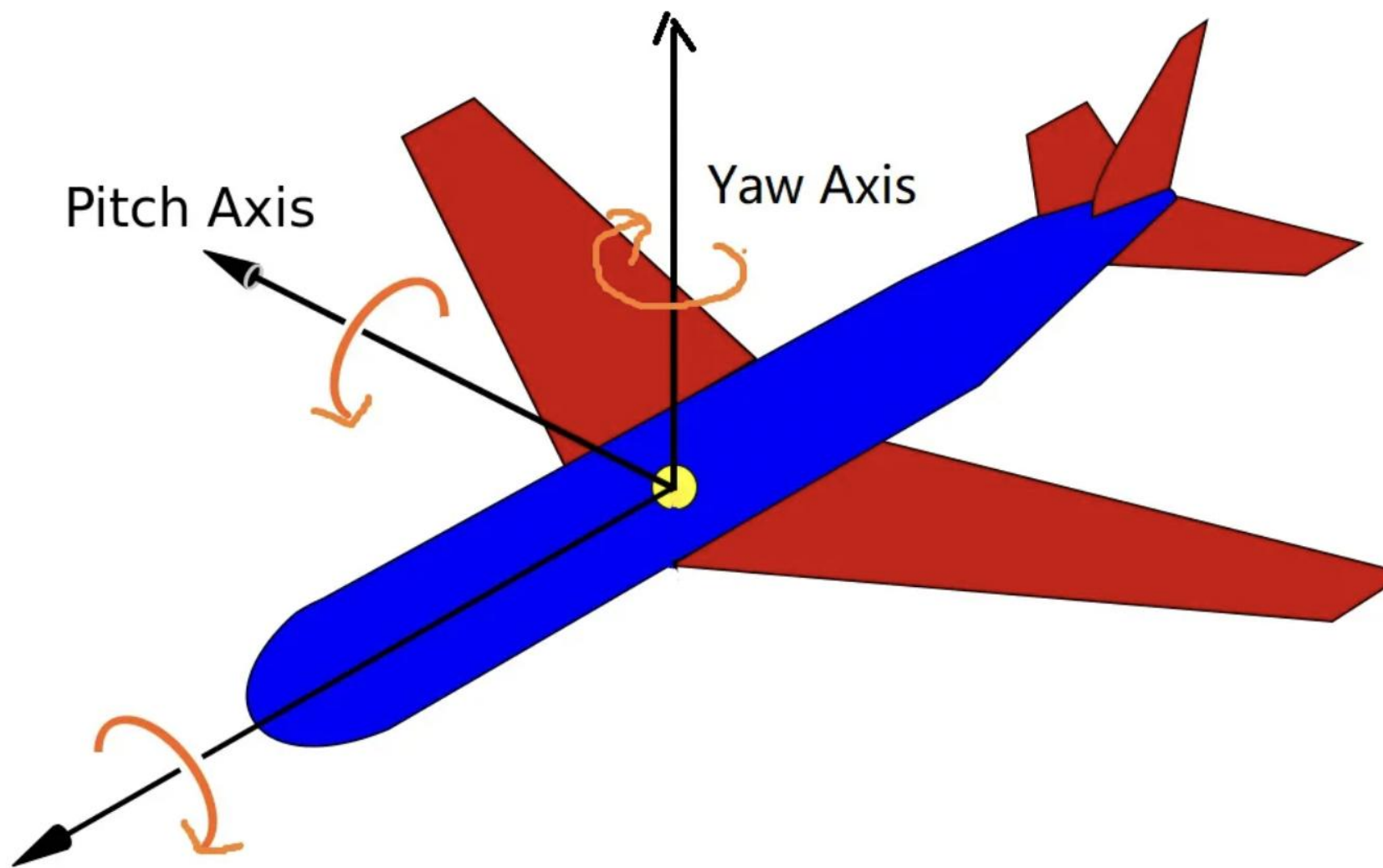


1. 严重依赖于内外参数的准确性
2. 始终假设地面是一个平面。
3. 始终假设地面和车底端平行



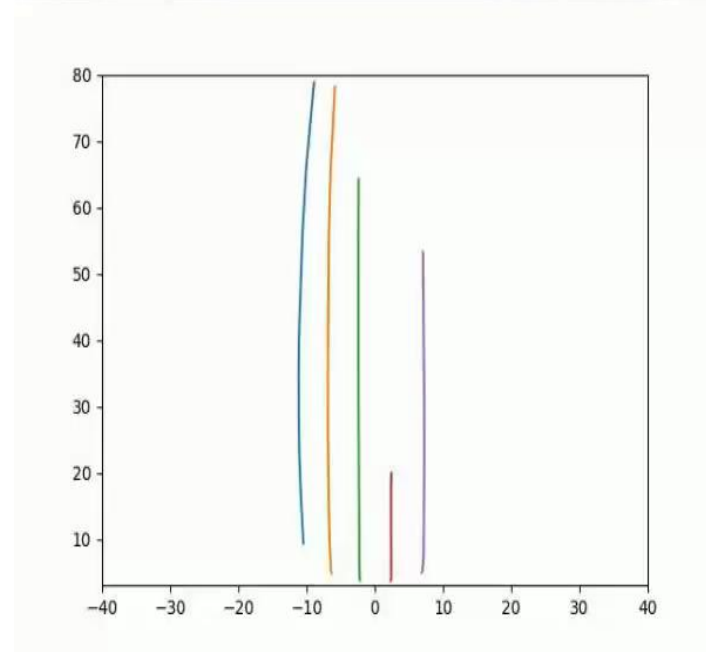
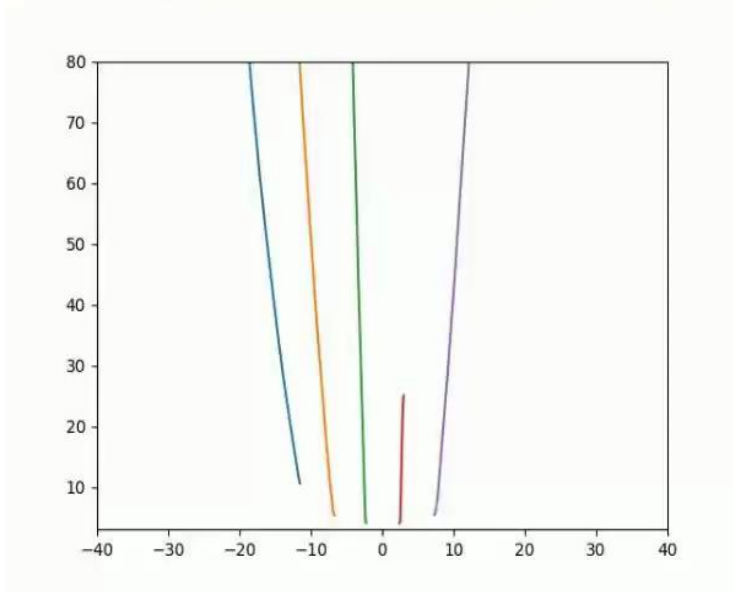
IPM 问题和补偿

1. 修正相机内参
2. 修正相机外参
3. 拟合曲面



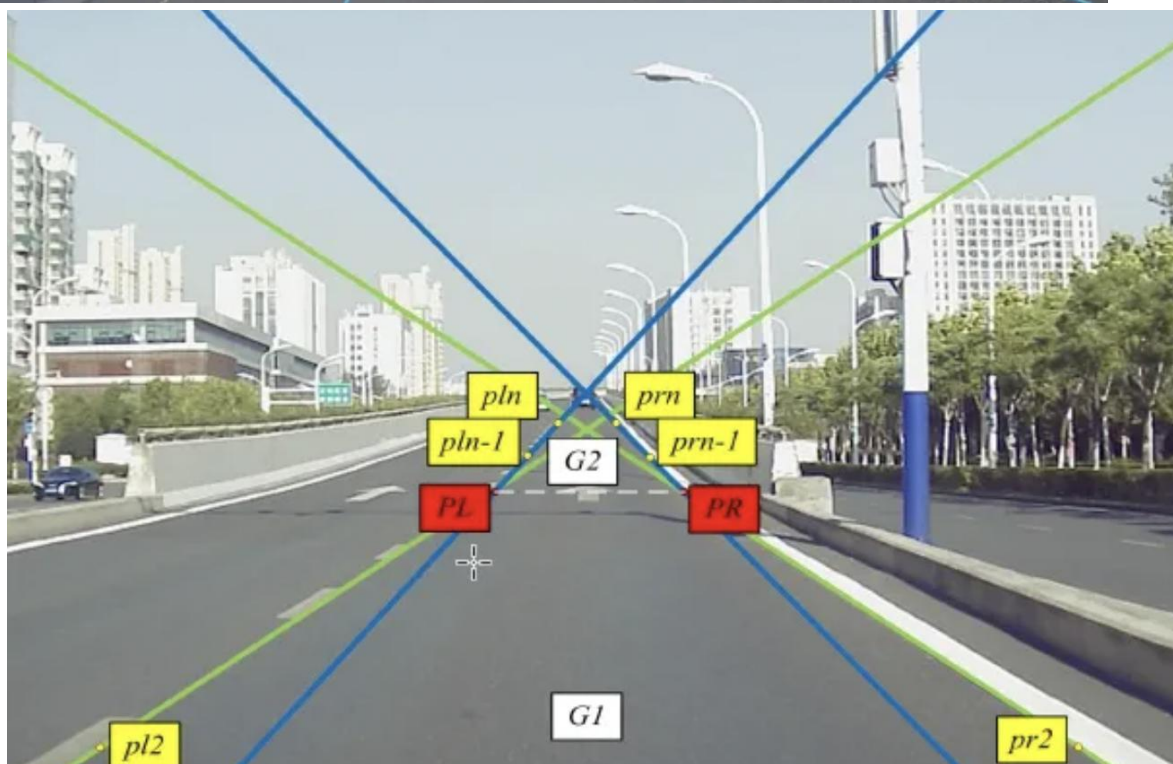
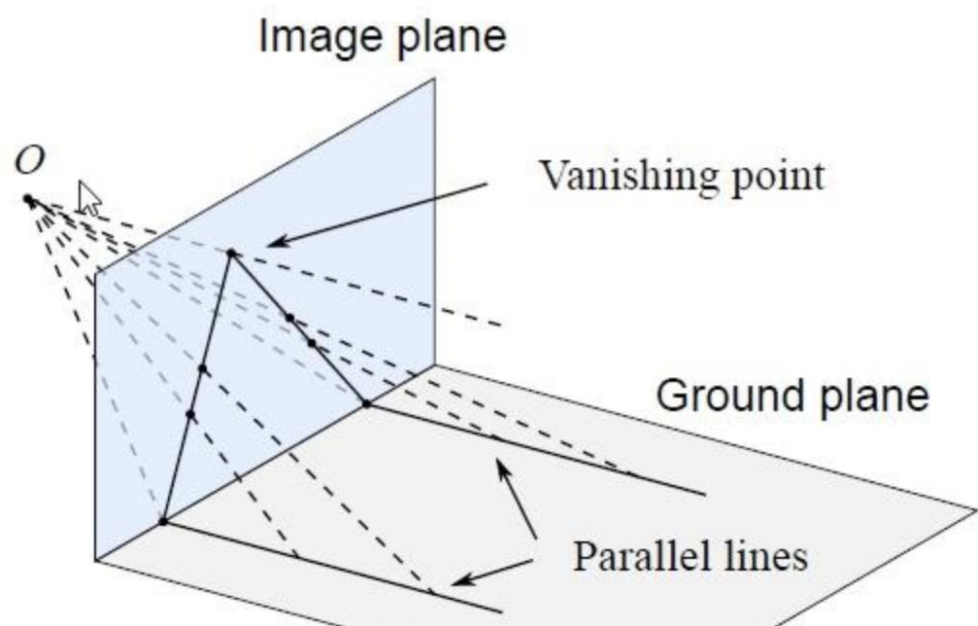
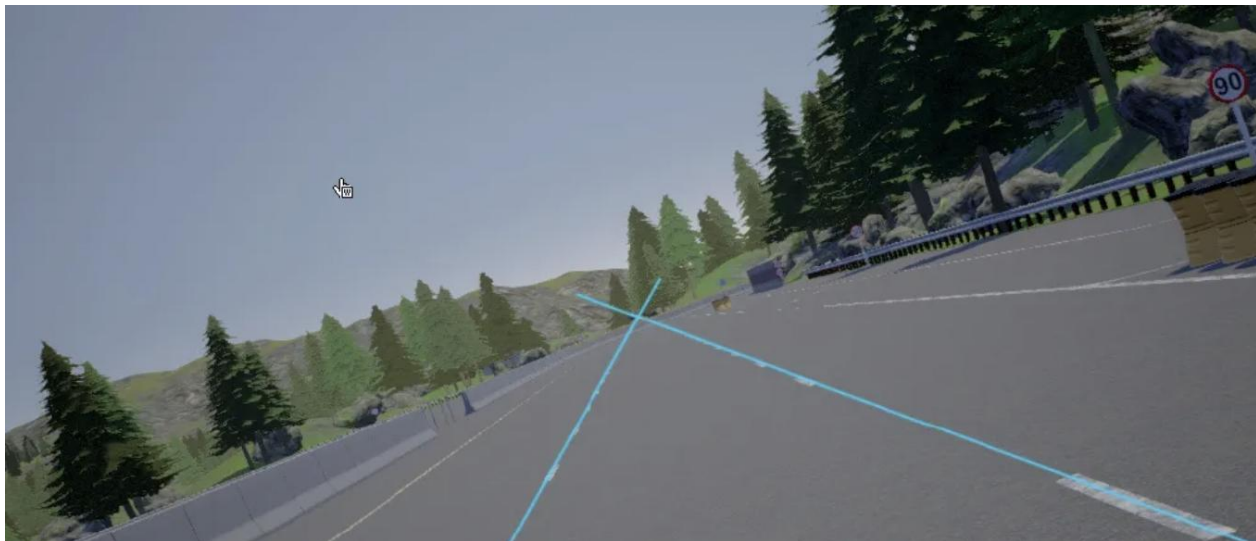
IPM 问题和补偿

1. 平行假说



IPM 问题和补偿

1. 基于vp点修正

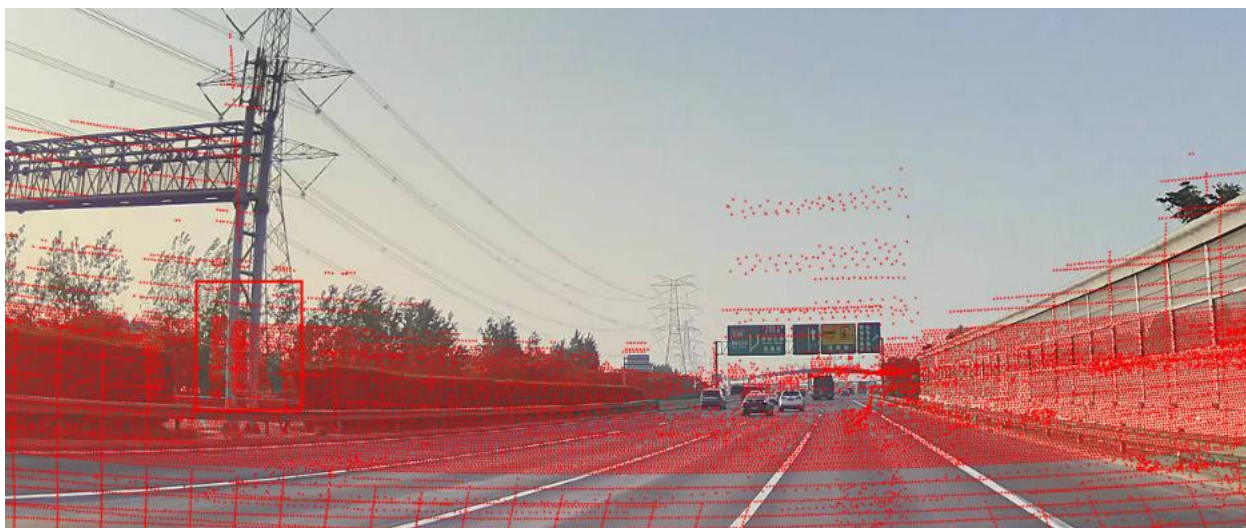
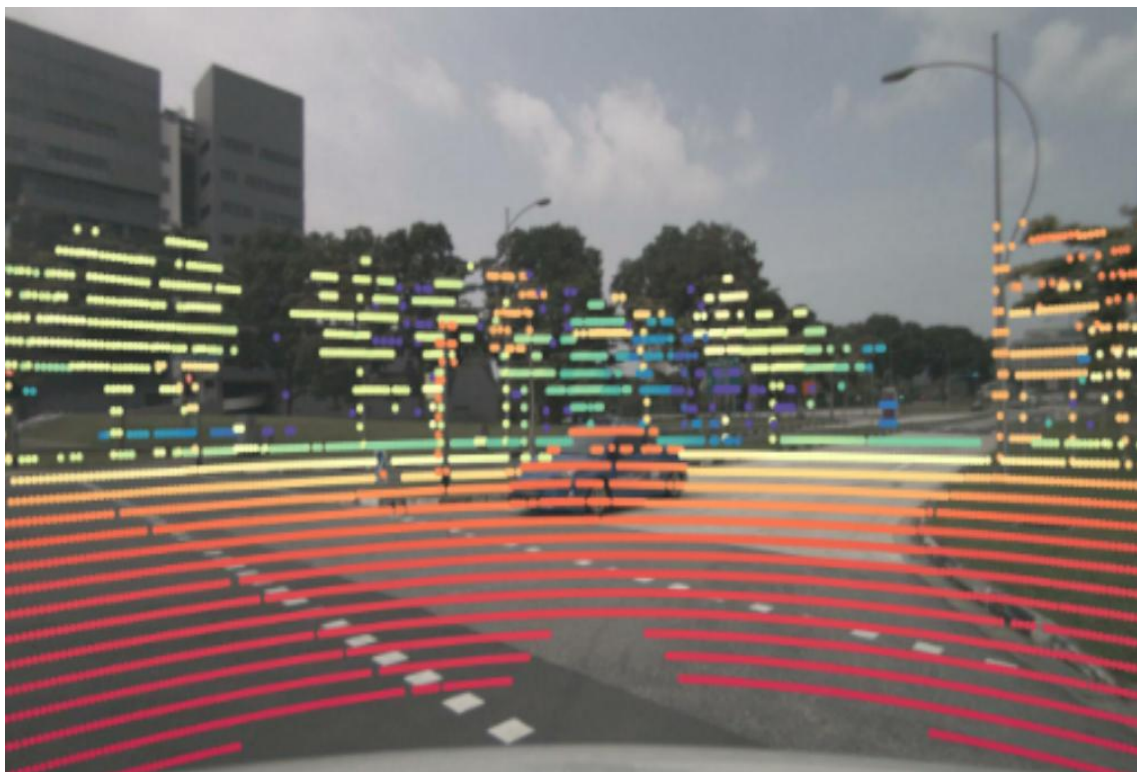


IPM 问题和补偿

基于地面和lidar进行融合

1. 基于前融合（BEV）

2. 基于后融合，根据lidar 点算出点的车道线物理世界的位置。返乡算出地面元素的高度。



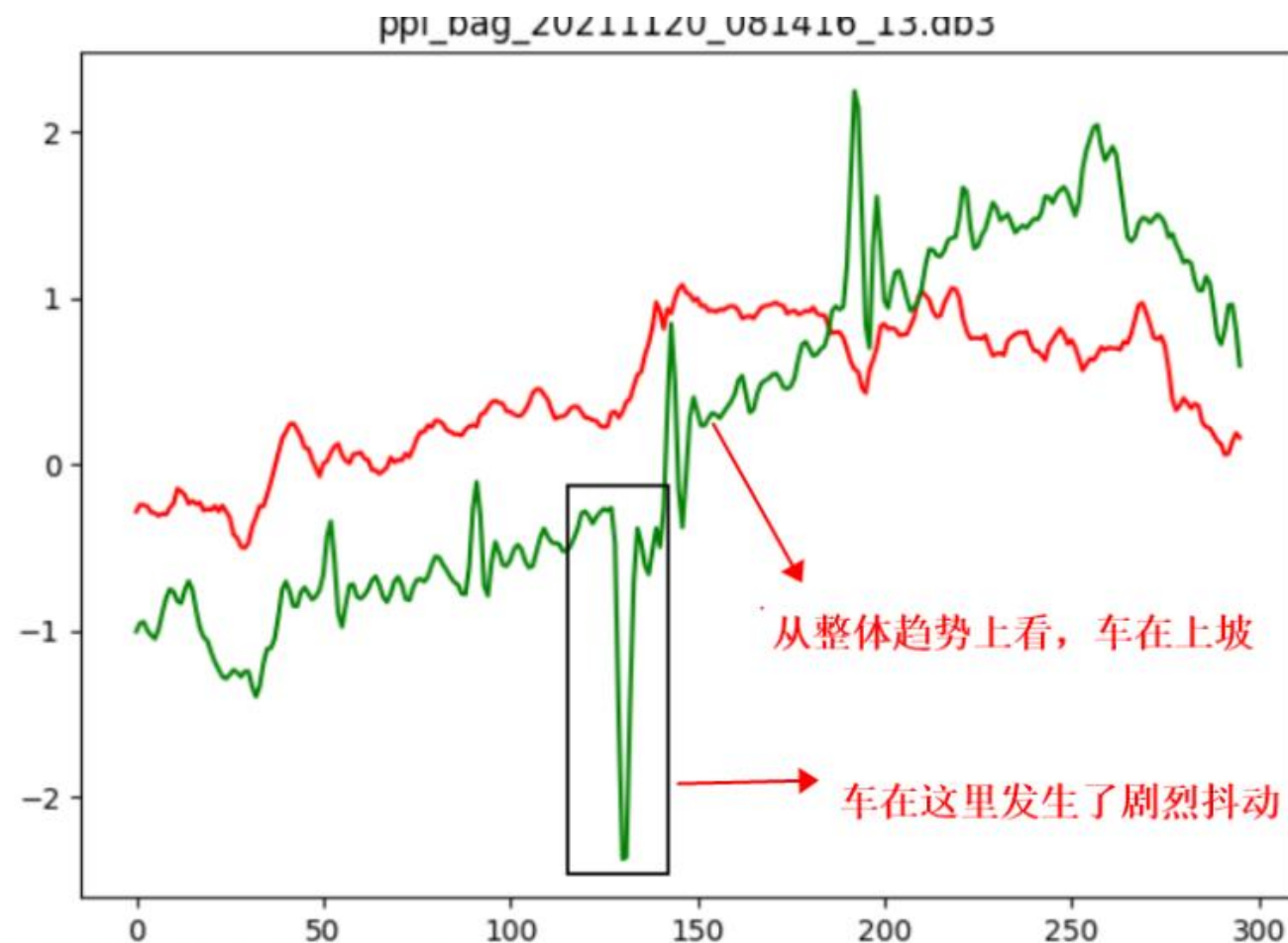
驶之心



IPM 问题和补偿

基于imu的补充

1. 车道线突然之间忽然发生了内八和外八，有坑是pitch 角度发生了抖动，这里用pitch角度进行补偿。



如图：这段路在某个时刻车发生了严重抖动。



公众号



自动驾驶与计算机视觉
日常干货分享

知识星球



加入自动驾驶之心知识星球，
获取更多硬核干货

自动驾驶与AI全栈技术学习+领域大咖交流+职位内推！