

# Longhorn

MATLAB & Simulink助力窄道自动驾驶的开发

曾峰, 豪恩

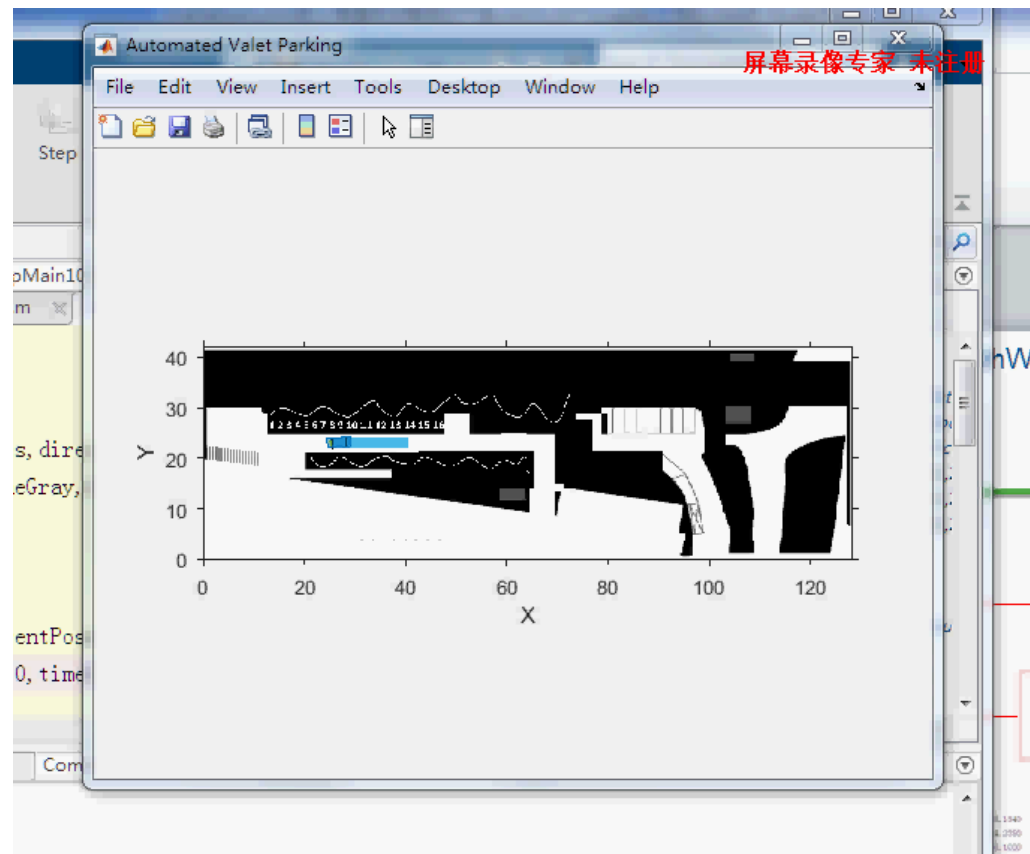


2022 MathWorks  
中国汽车年会

# 概述

本演讲涉及到一种约束场景下低速自动巡航的实现方法，具体内容包括：

- 建立起项目针对性的调试，测试仿真平台
- 改进了RRT + Dubins曲线 + costmap 的路径规划算法
- MATLAB & Simulink助力开发过程的介绍



# 系统架构

- 场景的共性是低速，有路面约束
- 系统是为视觉&超声波实现自动泊车搭建，窄道低速自动巡航只是其附加功能
- 低速下超声波可以有效的避免碰撞；语义分割可以在约束路面构建适合路径规划的粗略地图
- 窄道自动驾驶是技术积累，可以很好的为记忆泊车，远程泊车提供自主巡航解决方案



两边有车辆约束



单边/两边路约束



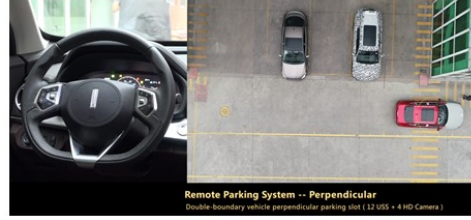
单边/双边车道线



限位桩约束

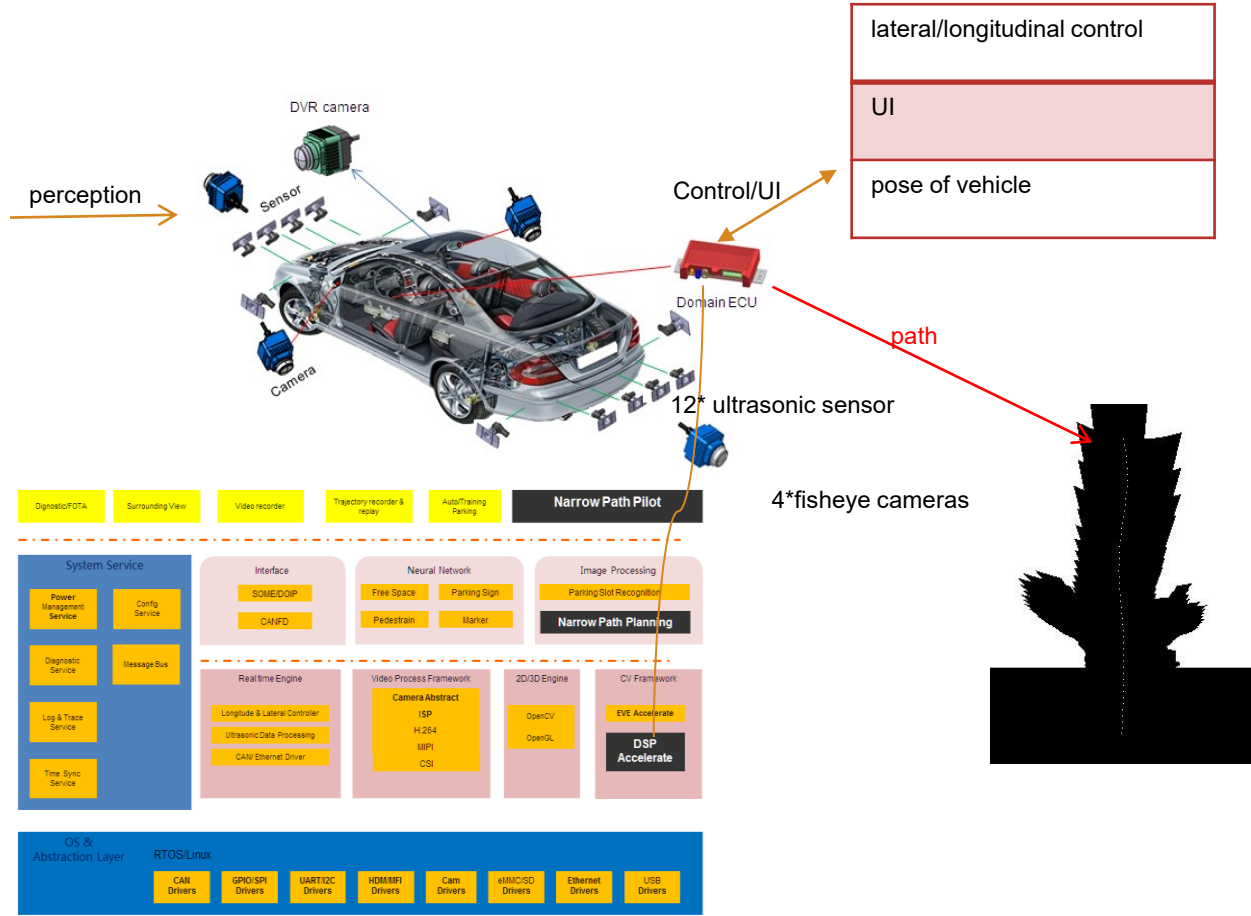


旋转路径的地下停车场



远程泊车的自主巡航过程

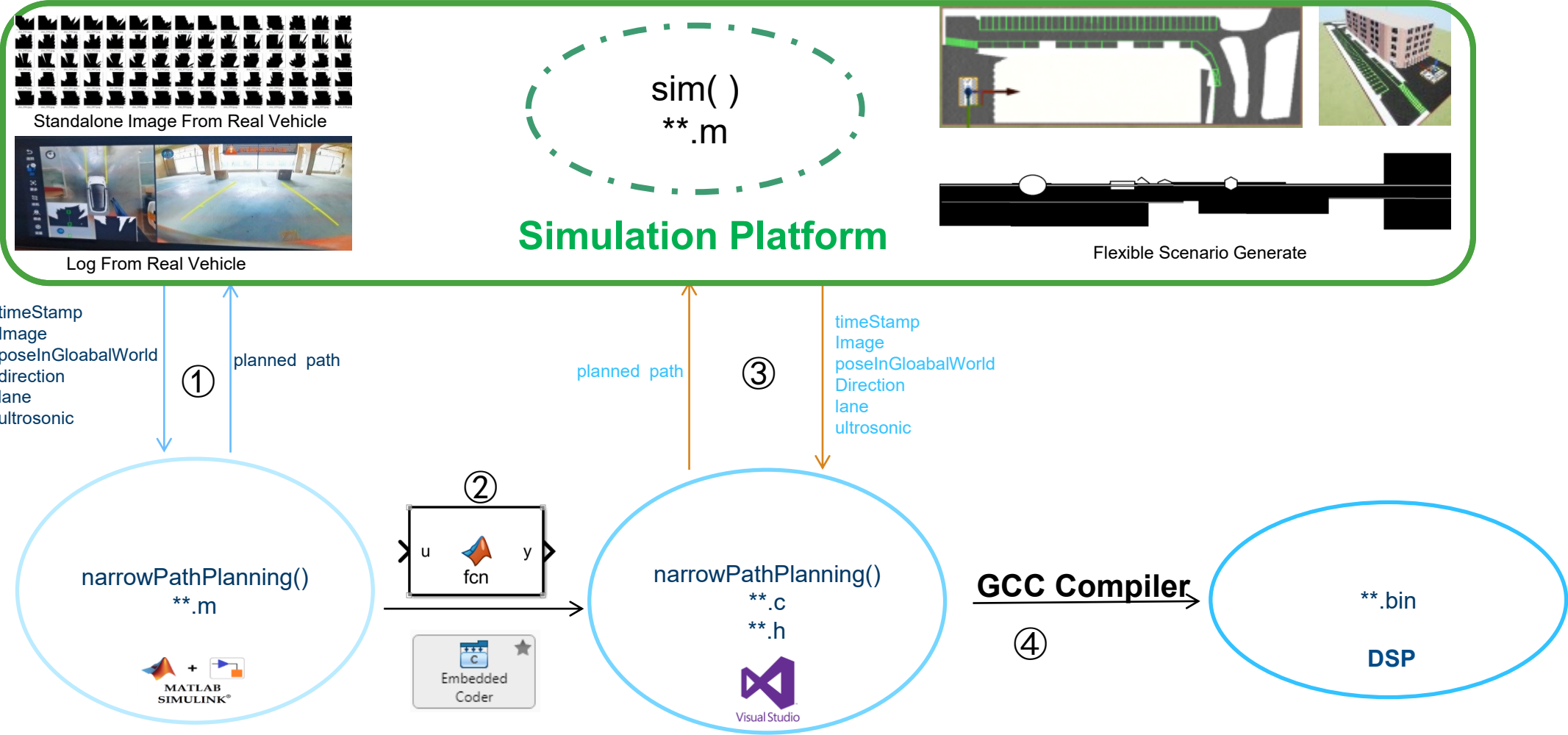
scenario



ECU software architecture

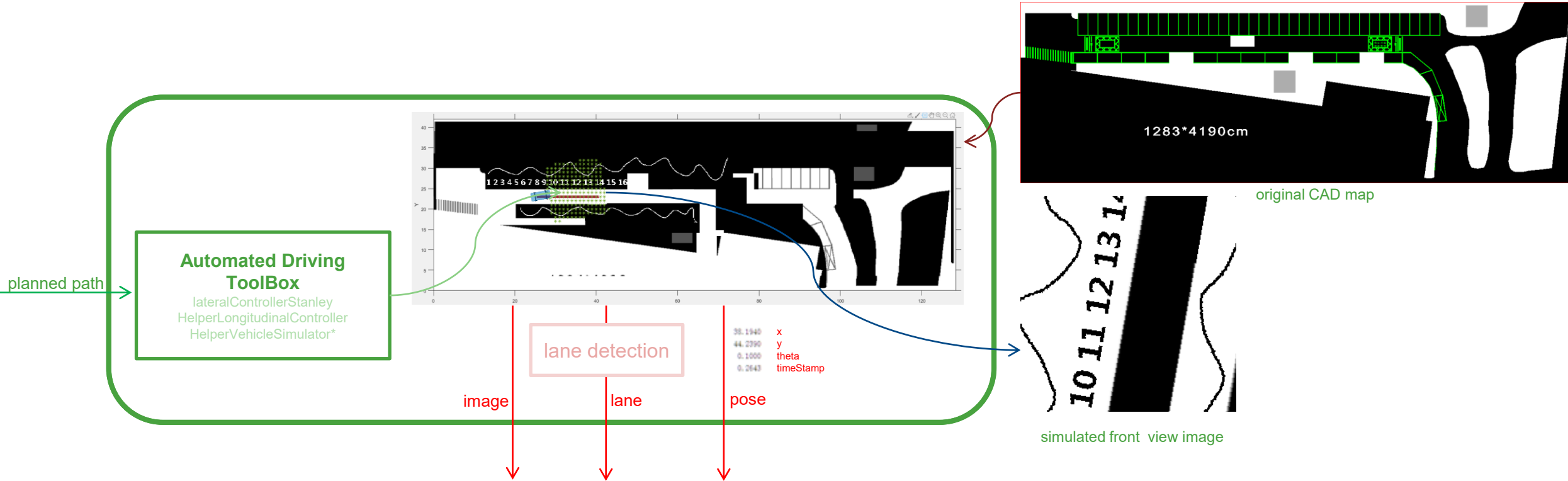
# 开发工具链

- 目标代码运行在DSP上，基于RTOS，决定了从C语言生成机器代码是第一选择
- 代码开发及更新流程按照图示的①②③④顺序执行
- 绿色线宽表示的是仿真平台，椭圆蓝色线是核心代码及中间产物



# 仿真平台 – 虚拟场景

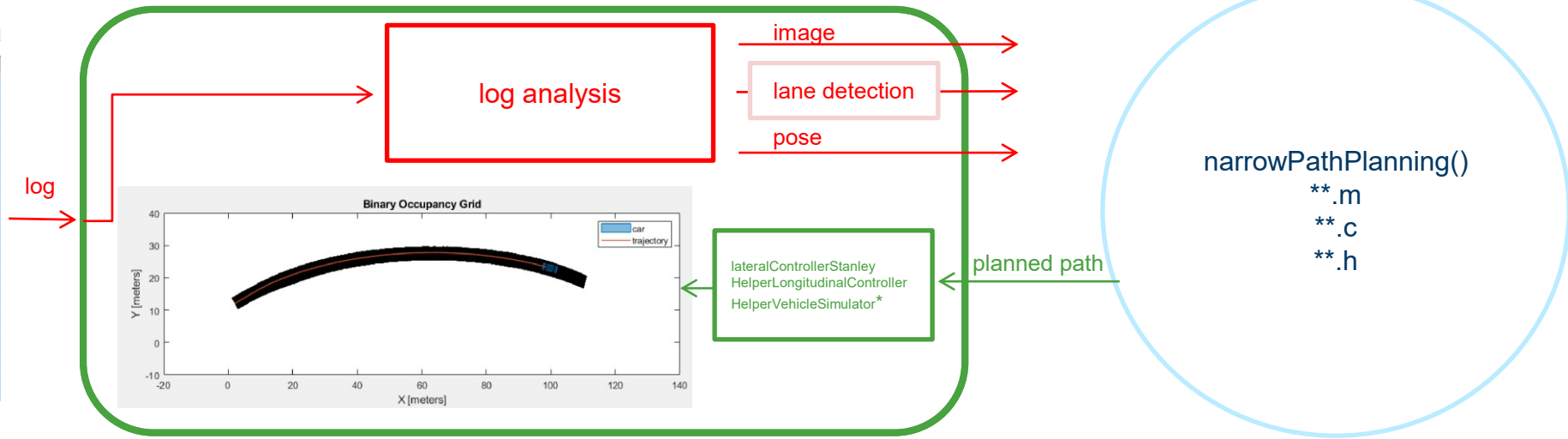
- 模拟车辆在定制地图中探索行进，传输全景系统获得的图像，车辆的姿态信息，处理完的车道线信息以及时间戳信息
- 可接受待测系统回传的路径信息，使用原生的方法产生控制信号并图示化车辆实际的位置姿态
- 地图是基于图像建立，很方便修改,可以实际测绘后使用CAD工具和图像处理工具生成几何位置准确的2D地图
- 平台可以直接支持同平台下\*.m文件的仿真，也支持基于VS的C/C++代码的仿真
- 在MATLAB原生仿真系统上做了针对性的改进，把时间速度驱动改成了距离驱动；沿用了原生的画图系统，车辆动力学模型



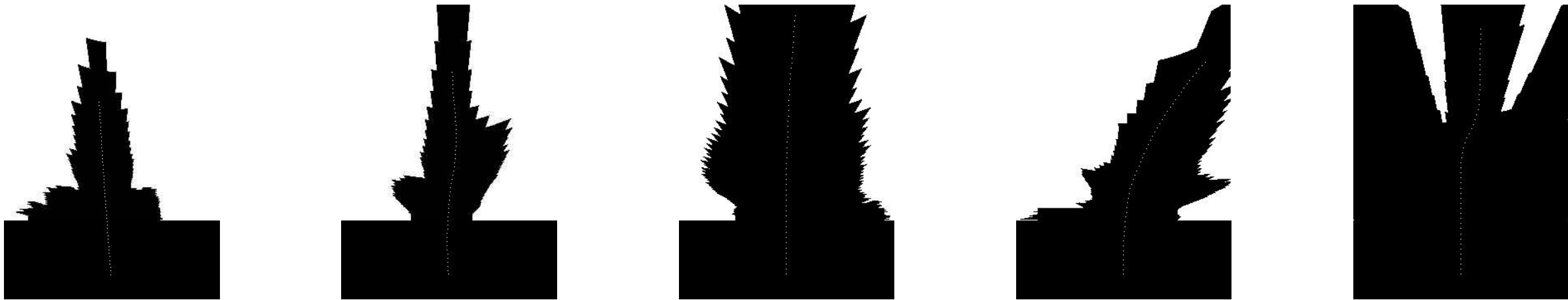
# 仿真平台 - 实车数据

Fields	time	egoDirection	egoVelocity	deltaXY	poseEgoInGlobalWorld	laneIn	UltrasonicData	imgIn
0	3052	1	0 [0,0]	[65.2527,21.5113,359.5807]	[1.180,99...]	1x12 double	260x260 logical	
1	3053	1	0 [0,0]	[65.2727,21.5111,359.5839]	[1.180,99...]	1x12 double	260x260 logical	
2	3054	1	0 [0,0]	[65.2927,21.5110,359.5871]	[1.180,99...]	1x12 double	260x260 logical	
3	3055	1	0 [0,0]	[65.3127,21.5108,359.5910]	[1.180,99...]	1x12 double	260x260 logical	
4	3056	1	0 [0,0]	[65.3327,21.5107,359.5949]	[1.180,99...]	1x12 double	260x260 logical	
5	3057	1	0 [0,0]	[65.3527,21.5105,359.5987]	[1.180,99...]	1x12 double	260x260 logical	
6	3058	1	0 [0,0]	[65.3727,21.5104,359.6024]	[1.180,99...]	1x12 double	260x260 logical	
7	3059	1	0 [0,0]	[65.3927,21.5103,359.6061]	[1.180,99...]	1x12 double	260x260 logical	
8	3060	1	0 [0,0]	[65.4127,21.5101,359.6097]	[1.180,99...]	1x12 double	260x260 logical	
9	3061	1	0 [0,0]	[65.4327,21.5100,359.6133]	[1.180,99...]	1x12 double	260x260 logical	
10	3062	1	0 [0,0]	[65.4527,21.5098,359.6168]	[1.180,99...]	1x12 double	260x260 logical	
11	3063	1	0 [0,0]	[65.4727,21.5097,359.6202]	[1.180,99...]	1x12 double	260x260 logical	
12	3064	1	0 [0,0]	[65.4927,21.5096,359.6236]	[1.180,99...]	1x12 double	260x260 logical	
13	3065	1	0 [0,0]	[65.5127,21.5095,359.6270]	[1.180,99...]	1x12 double	260x260 logical	
14	3066	1	0 [0,0]	[65.5327,21.5093,359.6303]	[1.180,99...]	1x12 double	260x260 logical	
15	3067	1	0 [0,0]	[65.5527,21.5092,359.6335]	[1.180,99...]	1x12 double	260x260 logical	
16	3068	1	0 [0,0]	[65.5727,21.5091,359.6367]	[1.180,99...]	1x12 double	260x260 logical	
17	3069	1	0 [0,0]	[65.5927,21.5089,359.6401]	[1.180,99...]	1x12 double	260x260 logical	
18	3070	1	0 [0,0]	[65.6127,21.5088,359.6435]	[1.180,99...]	1x12 double	260x260 logical	
19	3071	1	0 [0,0]	[65.6327,21.5087,359.6468]	[1.180,99...]	1x12 double	260x260 logical	
20	3072	1	0 [0,0]	[65.6527,21.5086,359.6501]	[1.180,99...]	1x12 double	260x260 logical	
21	3073	1	0 [0,0]	[65.6727,21.5084,359.6533]	[1.180,99...]	1x12 double	260x260 logical	
22	3074	1	0 [0,0]	[65.6927,21.5083,359.6565]	[1.180,99...]	1x12 double	260x260 logical	
23	3075	1	0 [0,0]	[65.7127,21.5082,359.6596]	[1.180,99...]	1x12 double	260x260 logical	
24	3076	1	0 [0,0]	[65.7327,21.5081,359.6627]	[1.180,99...]	1x12 double	260x260 logical	
25	3077	1	0 [0,0]	[65.7527,21.5080,359.6657]	[1.180,99...]	1x12 double	260x260 logical	

log from real vehicle in specific scenario



针对特殊场景（失败场景），采集包含图像，姿态的完整log，在仿真平台上回放整个过程，可视化的找出问题所在

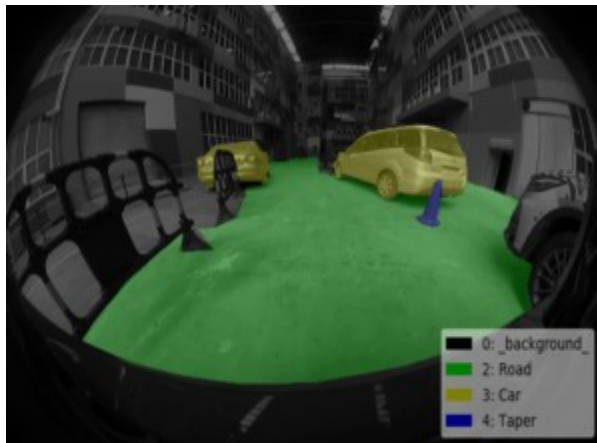
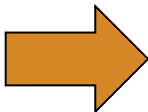


实车测试时候采集大量的图片，作为代码设计变更的测试输入

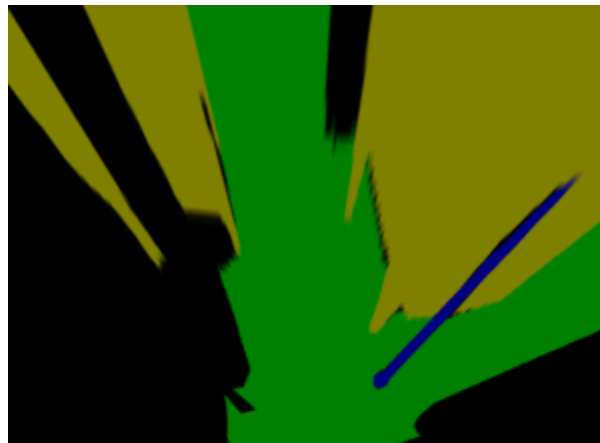
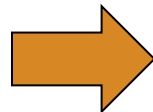
# 核心功能开发-图像处理流程



Original fisheye

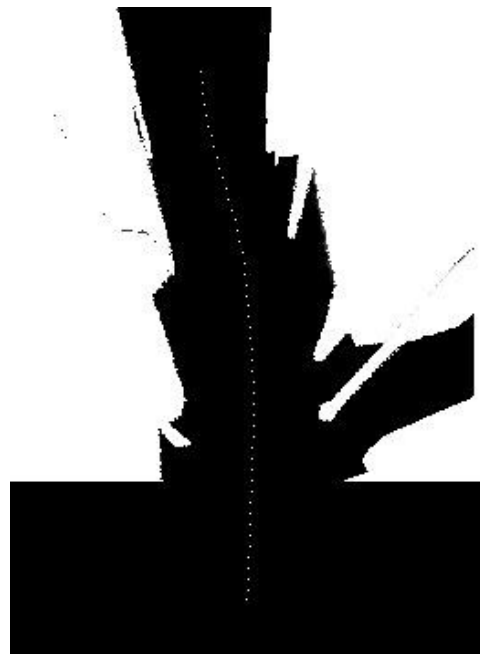


After semantic segmentation



Top view

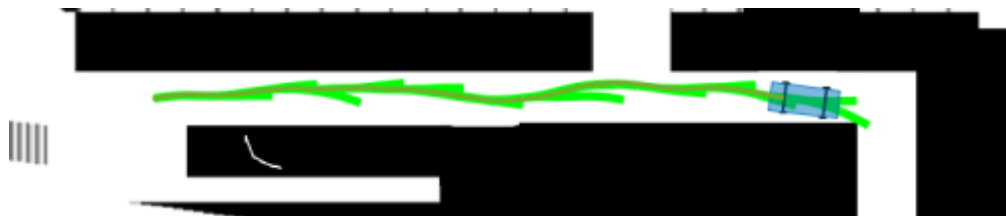
- 基于环视鱼眼摄像头采集图像，在鱼眼图中做基于深度学习的语义分割
- 世界坐标系的映射是通过标定实现的，达到厘米级精度，语义分割及下采样后的精度在分米级
- 关键步骤为在costmap中找到规划出合理的轨迹



Path planning



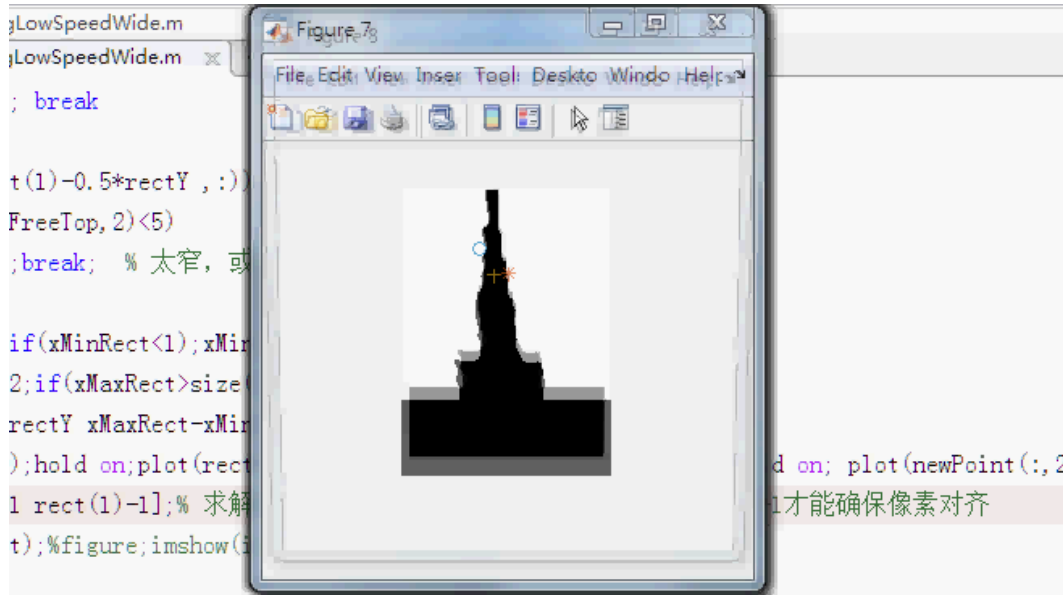
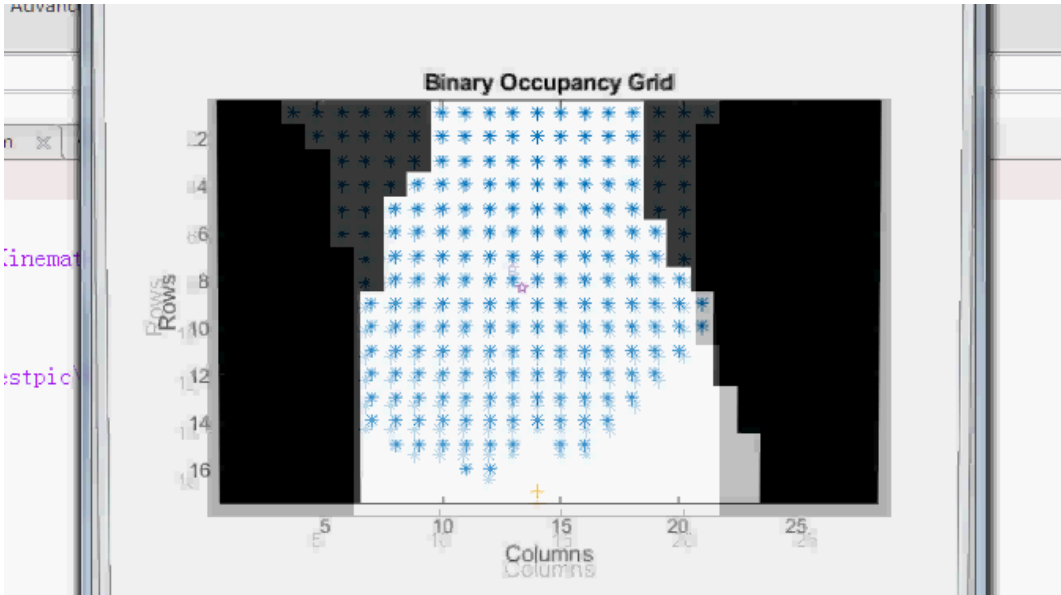
# 核心功能开发- 优化升级



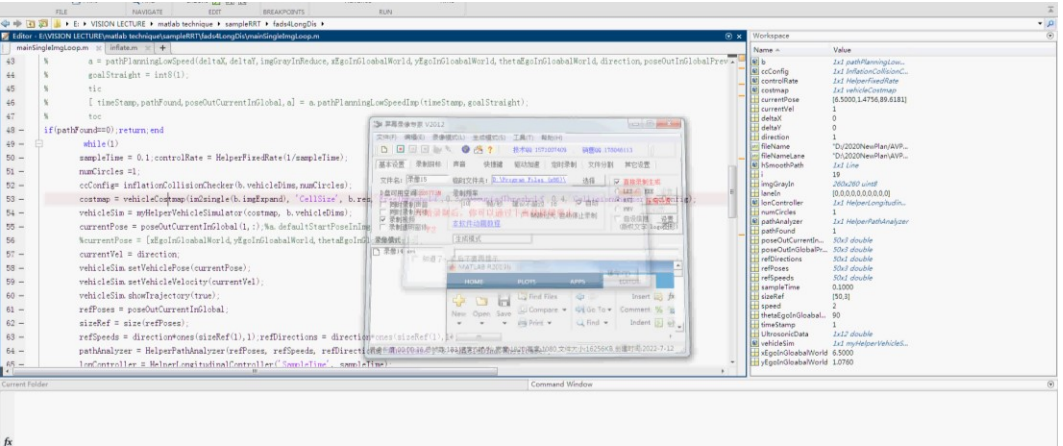
	第一代	第二代
算力消耗	算力消耗大，源于无序增长，源于全局costmap，以及Dubins曲线的计算，生成C代码包含大量的动态内存分配导致在嵌入式运行效率低，	做了三点改进来节约算力。第一，控制增长方向；第二，只做局部costmap；第三，简化行驶曲线
行驶轨迹	“蛇行”轨迹，和自然驾驶相差太远，实车体验效果不好。源于costmap相关参数为编译时输入，不方便迭代，导致规划的路径和最优解相差太远	优化costmap方法，可以在代码中迭代膨胀大小，找到局部最优节点，平滑化处理完的曲线更接近最优解
适用性	数学逻辑完备，即从理论上说，规划的路线是数学上可行驶路线，对避障，转弯有更好的性能	更适用于解决实际的道路问题。避障，转弯复杂场景下需要融合除可行驶区域之外的其他特征



# MATLAB & Simulink工具优势 – 可视化



强大，成熟的绘图工具让猜想的验证变得简单且有效，直观的看到抽象算法处理完的实际效果。相比C/C++环境下的开发，MATLAB & Simulink是更有效的加速器。



# MATLAB & Simulink工具优势 – 算法库及C代码生成

```
J = imerode(I,SE)
J = imerode(I,nhood)
J = imerode(___,packopt,m)
J = imerode(___,shape)
```

```
BW2 = bwselect(BW,c,r,n)
BW2 = bwselect(BW,n)
[BW2,idx] = bwselect(___)
BW2 = bwselect(x,y,BW,xi,yi,n)
[x,y,BW2,idx,xi,yi] = bwselect(___)
```

```
costmap = vehicleCostmap(C)
costmap = vehicleCostmap(mapWidth,mapLength)
costmap = vehicleCostmap(mapWidth,mapLength,costVal)
costmap = vehicleCostmap(occMap)
costmap = vehicleCostmap(___, 'MapLocation',mapLocation)
costmap = vehicleCostmap(___,Name,Value)
```

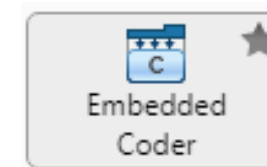
```
planner = pathPlannerRRT(costmap)
planner = pathPlannerRRT(costmap,Name,Value)
```

```
[poses,directions] = smoothPathSpline(refPoses,refDirections,numSmoothPoses)
[poses,directions] = smoothPathSpline(refPoses,refDirections,numSmoothPoses,minSeparation)
[___,cumLengths,curvatures] = smoothPathSpline(___)
```

```
steerCmd = lateralControllerStanley(refPose,currPose,currVelocity)
steerCmd = lateralControllerStanley(refPose,currPose,currVelocity,Name,Value)
```

```
lines = houghlines(BW,theta,rho,peaks)
lines = houghlines(___,Name,Value,...)
```

- 你或许可以找到一个支持上述算法库的平台，但你很难找到一个既能支持上述算法库又基于C代码的平台，而这就是我面临的项目需求。
- MATLAB & Simulink帮助我完成简洁的交付，1个\*.c和1个\*.h文件，可以轻松集成到任何系统中。

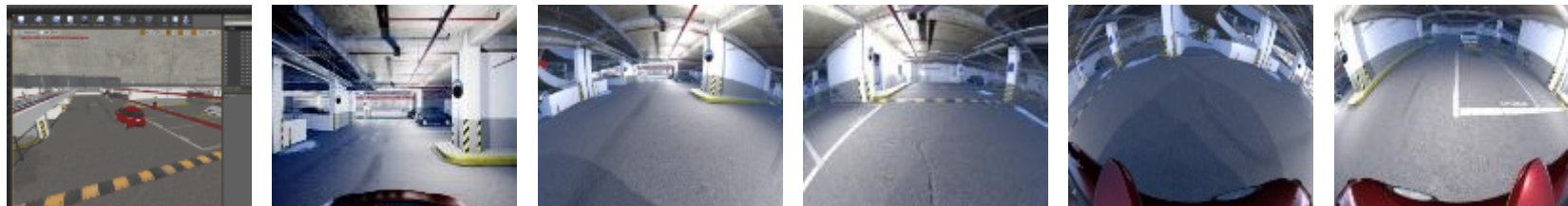


codegen('myCode', 'C')  
 myCode.c (453 KB)    myCode.h (18 KB)

# 下一代功能升级

## 仿真平台升级:

基于Simulink+unreal生成车辆姿态信息和5路摄像头信息以及真值表，支持跨层停车场场景的记忆泊车&代客泊车的SIL仿真



## 硬件配置升级:

1. 摄像头升级，FOV:120，FHD，微畸变。高分辨率可以看的更精确，看到更远的范围；微畸变可以提升语义分割建图位置精度
2. SoC算力升级，使用更高深度学习算力的平台，提升可行驶区域侦测的准确度；结合车位，柱子，车辆及其姿态信息，生成混合的高精度地图

## 拓宽使用场景:

1. 远程泊车&记忆泊车过程中，实现地下停车场的可行驶区域建图及定位
2. 地下停车场的旋转路径引导

# 2022 MathWorks 中国汽车年会

Thank you

