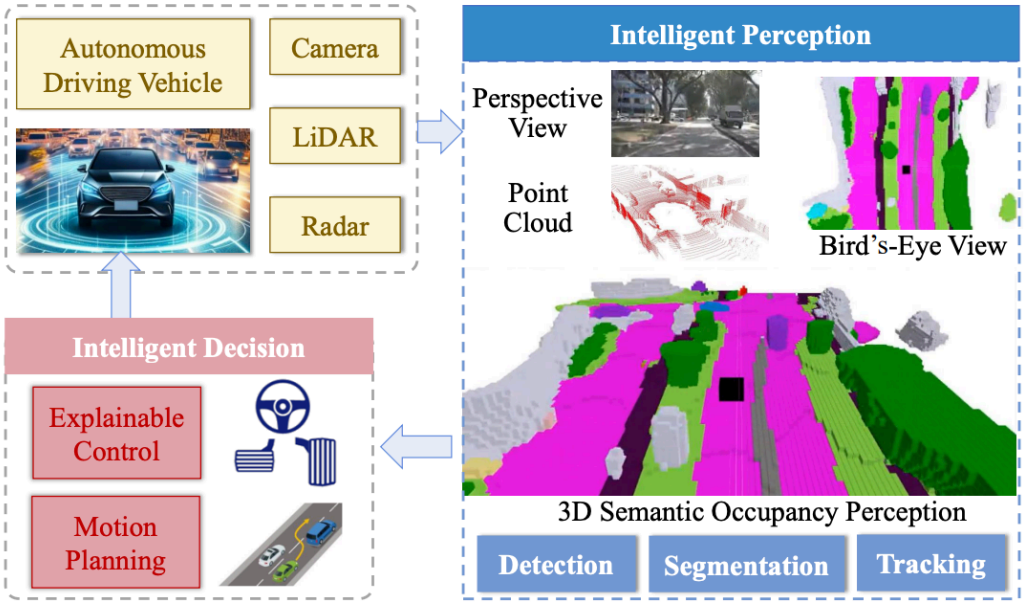
1. 基于Occupancy算法的Autonomous嵌入式系统架构



嵌入式系统用于自动驾驶时，需要具备强大的实时处理能力和对多传感器的高效数据融合能力。主要涉及的架构设计包括： - **处理器架构**：嵌入式系统通常采用SoC（系统级芯片）架构，集成了CPU、GPU和AI加速器。自动驾驶中常用的嵌入式平台包括： - **NVIDIA DRIVE AGX**：带有GPU加速单元，适合处理高性能感知任务。 - **Mobileye EyeQ** 系列：低功耗专用AI处理器，适合嵌入式系统的感知与决策计算。 - **FPGA** 和 **ASIC**：可以根据占据网格算法的特定需求进行自定义硬件加速，降低功耗并提高效率。

- **存储和带宽**：
  - **内存限制**：嵌入式系统的内存通常有限，特别是在自动驾驶的动态环境中，存储大量高分辨率占据网格数据是一个挑战。嵌入式系统需要优化存储使用，例如压缩占据网格或使用稀疏存储格式来减少内存占用。
  - **通信带宽**：自动驾驶车辆中有多个传感器并行运行，每秒生成大量数据。嵌入式系统通过总线（如CAN、PCIe、Ethernet等）传输数据，要求高带宽和低延迟来确保传感器融合的实时性。

2. 实时操作系统（RTOS）

在自动驾驶嵌入式系统中，**实时操作系统（RTOS）**是关键，它确保各任务在规定时间内得到处理，避免延迟对系统的安全性产生影响。使用RTOS的好处包括：

- **任务调度**：自动驾驶嵌入式系统需要处理多种任务（传感器数据采集、占据网格生成、路径规划等），RTOS可以确保这些任务按优先级有序执行，避免系统过载。

- **中断管理**：传感器数据（如激光雷达或摄像头）通常以中断的方式进入系统，RTOS需要快速响应中断信号，确保数据及时处理。
- **硬实时需求**：自动驾驶系统要求对关键事件（如突然出现的障碍物）作出实时响应，RTOS必须保证硬实时性以避免延迟导致的安全问题。

常见的RTOS包括：

- **FreeRTOS**：开源、轻量的RTOS，适用于资源受限的嵌入式系统。
- **QNX**：广泛应用于汽车行业，具备强大的实时性能和可靠性。
- **VxWorks**：高性能实时操作系统，适用于高安全性要求的嵌入式自动驾驶系统。

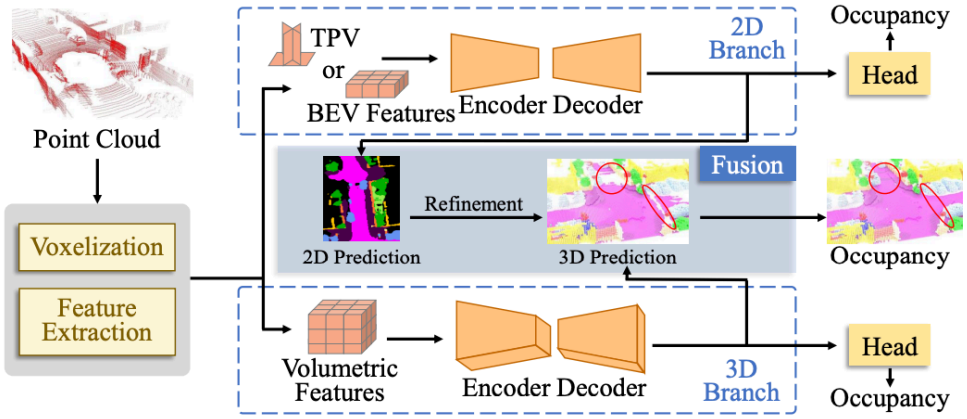
3. 多传感器数据融合

嵌入式系统需要实时处理来自多个传感器（如摄像头、雷达、激光雷达、IMU等）的数据，并通过**数据融合算法**生成一致的环境模型。在基于占据网格的自动驾驶方案中，嵌入式系统的传感器融合任务包括：

- **时间同步**：不同传感器的数据采集频率不同，嵌入式系统需要通过时间戳和插值算法确保多传感器数据的时间同步。
- **空间对齐**：传感器的安装位置和视角不同，系统需要通过外参（extrinsic parameters）将数据对齐到相同的参考坐标系中。嵌入式系统需执行矩阵变换和坐标映射等操作来实现这一点。
- **数据滤波**：嵌入式系统会使用滤波算法（如卡尔曼滤波、扩展卡尔曼滤波（EKF）等）来处理传感器噪声，得到平滑和可靠的感知数据。

嵌入式系统需具备高效的并行处理能力以处理这些传感器融合任务。例如，NVIDIA的嵌入式平台通过GPU加速传感器数据处理，Mobileye则通过专用的硬件加速器来完成融合计算。

4. 占据网格（Occupancy Grid）算法的实现



在嵌入式系统中，**占据网格算法**用于对环境进行离散化建模，并更新各栅格单元的占据状态。该算法的实现步骤包括： - **网格表示**：嵌入式系统中将车辆周围的空间离散化为网格，每个单元格（cell）代表特定的空间位置，并分配一个占据概率值（通常为0到1之间的概率）。 - **贝叶斯更新**：占据网格的状态根据传感器输入数据进行更新，使用贝叶斯滤波方法修正各单元的占据状态。嵌入式系统需要不断进行实时计算，以确保每个网格的占据状态基于传感器的最新数据得到更新。 - **稀疏存储优化**：嵌入式系统的存储资源有限，对于大型占据网格，通常只存储被占据或接近障碍物的部分区域，以降低存储和计算开销。
在实现层面，嵌入式系统可以采用GPU或FPGA来加速占据网格的生成与更新过程。例如，NVIDIA的CUDA并行编程模型允许占据网格的每个单元格并行更新，从而显著加速计算。

5. 路径规划与控制

占据网格提供了对车辆周围环境的精确感知，嵌入式系统在此基础上进行路径规划和车辆控制。常用的路径规划算法包括：

- **A*算法**：一种启发式搜索算法，用于在占据网格上寻找从起始位置到目标位置的最短路径。嵌入式系统中，A*算法在二维网格上运行时，必须考虑实时性和计算开销，使用优化版本（如D*算法）以提高效率。
- **动态窗口法（DWA）**：这是一种在线路径规划算法，结合占据网格信息，基于车辆的运动学模型实时生成局部路径，适合动态障碍物环境。

运动控制方面，嵌入式系统需要根据路径规划生成的轨迹，向车辆执行机构（如转向、加速、制动）发送控制信号。自动驾驶嵌入式系统中，控制信号通过**线控技术（Drive-by-wire）**传输，代替传统的机械连接，直接控制车辆的行驶。

6. 功耗与能效优化

嵌入式系统往往工作在功耗受限的环境中，尤其是在电动汽车或无人驾驶车辆中，系统的能效至关重要。以下是针对功耗的优化措施：

- **低功耗处理器选择**：嵌入式系统常采用低功耗、高性能处理器（如ARM Cortex系列），在满足计算需求的同时降低功耗。
- **动态电源管理**：通过动态调节处理器频率和电压（DVFS），嵌入式系统可以在不同的工作负载下自动调整功耗，保证在低负载时节省能源。
- **硬件加速器**：如FPGA、DSP等专用硬件加速器用于高效处理特定任务（如占据网格计算、图像处理），降低通用处理器的功耗负担。

7. 安全性与容错性设计

自动驾驶嵌入式系统需要具备高可靠性和容错能力，以应对突发故障和极端环境。嵌入式系统中的安全性设计包括：

- **故障检测与隔离**：嵌入式系统可以通过冗余设计（如多传感器冗余、双通道控制系统）来检测系统故障，并在故障发生时切换到安全模式。
- **功能安全（ISO 26262）**：嵌入式系统需要符合ISO 26262等汽车功能安全标准，确保在故障条件下，系统能够保证最小风险状态。
- **软件与硬件容错**：嵌入式系统中的实时操作系统可以采用硬件冗余和软件容错技术，确保在关键任务发生错误时系统能够继续稳定运行。

总结：

Method	IoU	mIoU	car (2.85%)	bicycle (0.01%)	motorcycle (0.01%)	truck (0.16%)	other-veh. (5.75%)	person (0.02%)	road (14.98%)	parking (2.31%)	sidewalk (6.43%)	other-grnd. (0.56%)	building (15.67%)	fence (0.96%)	vegetation (41.99%)	terrain (7.10%)	pole (0.22%)	traf.-sign (0.06%)	other-struct. (4.33%)	other-obj. (0.28%)
LiDAR-Centric Methods																				
SSCNet [25]	**53.58**	**16.95**	**31.95**	0.00	**0.17**	**10.29**	0.00	**0.07**	**65.70**	**17.33**	**41.24**	**3.22**	**44.41**	**6.77**	**43.72**	**28.87**	**0.78**	**0.75**	**8.69**	**0.67**
LMSCNet [28]	47.35	13.65	20.91	0.00	0.00	0.26	0.58	0.00	62.95	13.51	33.51	0.20	43.67	0.33	40.01	26.80	0.00	0.00	3.63	0.00
Vision-Centric Methods																				
GaussianFormer [154]	35.38	12.92	18.93	1.02	4.62	18.07	7.59	3.35	45.47	10.89	25.03	5.32	28.44	5.68	29.54	8.62	2.99	2.32	9.51	5.14
MonoScene [26]	37.87	12.31	19.34	0.43	0.58	8.02	2.03	0.86	48.35	11.38	28.13	3.32	32.89	3.53	26.15	16.75	6.92	5.67	4.20	3.09
OccFiner (Mono.) [155]	38.51	13.29	20.78	1.08	1.03	9.04	3.58	1.46	53.47	12.55	31.27	4.13	33.75	4.62	26.83	18.67	5.04	4.58	4.05	3.32
VoxFormer [33]	38.76	11.91	17.84	1.16	0.89	4.56	2.06	1.63	47.01	9.67	27.21	2.89	31.18	4.97	28.99	14.69	6.51	6.92	3.79	2.43
TPVFormer [32]	40.22	13.64	21.56	1.09	1.37	8.06	2.57	2.38	52.99	11.99	31.07	3.78	34.83	4.80	30.08	17.52	7.46	5.86	5.48	2.70
OccFormer [55]	40.27	13.81	22.58	0.66	0.26	9.89	3.82	2.77	54.30	13.44	31.53	3.55	36.42	4.80	31.00	19.51	7.77	8.51	6.95	4.60
Symphonies [88]	44.12	18.58	**30.02**	1.85	**5.90**	**25.07**	**12.06**	**8.20**	54.94	13.83	32.76	**6.93**	35.11	**8.58**	38.33	11.52	14.01	9.57	**14.44**	**11.28**
CGFormer [156]	**48.07**	**20.05**	29.85	**3.42**	3.96	17.59	6.79	6.63	**63.85**	**17.15**	**40.72**	5.53	**42.73**	8.22	**38.80**	**24.94**	**16.24**	**17.45**	10.18	6.77

基于**占据网格算法**的自动驾驶嵌入式系统实现涉及多
方面的技术，包括高效处理器架构设计、实时操作系统、传感器数据融合、占据网格的生成与更

新、路径规划与控制、功耗管理以及安全性设计。这些技术共同保障了嵌入式自动驾驶系统的实时性、可靠性和功效。