智能驾驶功能软件平台设计规范

第二部分：感知融合功能服务接口

版本：1.0

2020-07-29 发布

目　　次

1 规范应用范围 1
2 规范引用文件 2
3 缩略语 3
4 感知融合功能概述 4
5 感知融合功能服务接口 6
5.1 目标识别和跟踪服务接口 6
5.2 道路结构服务接口 6
5.3 交通标志服务接口 7
5.4 可行驶空间服务接口 7
5.5 自车状态服务接口 8
5.6 泊车位信息服务接口 8
5.7 环境模型服务接口 8
6 数据结构定义 10
6.1 标准元数据头 10
6.2 几何描述 11
6.3 位置变换 12
6.4 位姿 13
6.5 速度 13
6.6 加速度 13
6.7 动静态目标元数据 14
6.8 车道线元数据 16
6.9 停止线元数据 17
6.10 交通灯元数据 17
6.11 交通标志元数据 18
6.12 栅格信息描述 18
6.13 栅格单元信息 19
6.14 泊车位元数据 20
6.15 车辆状态元数据 20
附录 A 22
A.1 数据结构定义描述 22
A.2 目标识别和跟踪服务接口 25
A.3 道路结构服务接口 25
A.4 交通标志接口 26
A.5 可行驶空间接口 26
A.6 车辆状态服务接口 26
A.7 泊车位信息服务接口 26
A.8 环境模型服务接口 26

编制说明

高级别智能驾驶是一项庞大的系统工程，是多个领域前沿技术的融合体，其中涵盖芯片、操作系统、通信和云等 ICT 技术，感知、推理、决策控制等智能算法技术，以及驱动、转向、制动等车辆底层控制技术。目前，高级别智能驾驶还处于成熟前夜，亟需跨产业紧密合作来解决所面临的技术困难，全产业链合力共同促进智能驾驶产业的发展。

根据《车载智能计算基础平台参考架构 1.0》（2019 年）中关于车载计算平台架构的描述，车载计算平台自底向上可以划分为硬件平台、系统软件、功能软件和应用软件等四层。功能软件层包含感知融合、定位、预测和决策规划等核心功能和算法模块，通过北向 API 接口承载智能驾驶场景应用，是智能驾驶系统的核心部分。

本规范旨在通过联合业内优势企业，构建一个标准化的功能软件平台，定义功能软件层的系统架构以及功能模块和算法组件之间的逻辑服务接口，希望针对行业的紧急需求达成共识，从而可以明确产业分工和边界，缩短智能驾驶系统的开发周期并降低系统集成成本。

本规范参加编制单位：

国汽(北京)智能网联汽车研究院有限公司、华为技术有限公司、中国软件评测中心、中国第一汽车集团公司、东风汽车集团有限公司、重庆长安汽车股份有限汽车、上海汽车集团股份有限公司商用车技术中心、广州汽车集团股份有限公司、吉利汽车研究院（宁波）有限公司、北京汽车研究总院有限公司、开沃新能源汽车集团有限公司、中国汽车工程研究院股份有限公司、禾多科技（北京）有限公司、湖南湘江智芯云途科技有限公司、长沙智能驾驶研究院有限公司、畅加风行（苏州）智能科技有限公司

1 规范应用范围

本规范规定了智能驾驶功能软件平台中的感知融合功能服务接口。

本规范适用于设计开发 GB/T《汽车驾驶自动化分级》[1]所定义的 2 级及以上的驾驶自动化系统即智能驾驶系统。

2 规范引用文件

本规范的引用和参考文件见下表：

[1] GB/T《汽车驾驶自动化分级》

[2] ISO/DIS 23150:2020(E) Road vehicles — Data communication between sensors and data fusion unit for automated driving functions — Logical interface

[3] ISO 8855:2011 Road vehicles — Vehicle dynamics and road-holding ability

[4] ADASIS(Advanced Driver Assistance Systems Interface Specification) v3

[5] 智能驾驶功能软件平台设计规范 第一部分：系统架构

[6] 智能驾驶功能软件平台设计规范 第三部分：预测功能服务接口

[7] 智能驾驶功能软件平台设计规范 第四部分：定位功能服务接口

[8] 智能驾驶功能软件平台设计规范 第五部分：决策规划功能服务接口

3 缩略语

文档中可能使用到的缩写说明如下所示：

缩略语	英文全名	中文解释
AD	Autonomous Driving	自动驾驶
ADAS	Advanced driver-assistance systems	高级辅助驾驶系统
ODD	Operational Design Domain	设计运行区域
APP	Application	应用程序
VCS	Vehicle Coordination System	车体坐标系

4 感知融合功能概述

智能驾驶功能软件平台基于不同厂商的技术实现方案进行功能抽象，共分为传感器抽象功能、感知融合功能、预测功能、决策规划功能、定位功能和执行器抽象 6 个功能模块。主机厂基于自身策略，在设计和开发功能软件时可以选择不同的功能模块和算法组件，实现拼插式功能组合，灵活构建智能驾驶系统级解决方案。

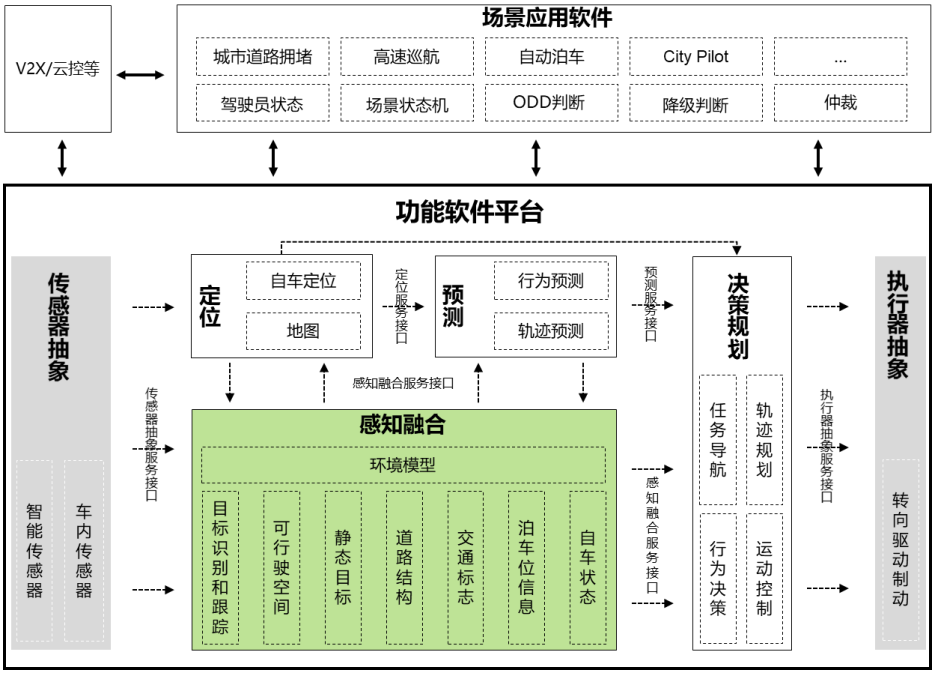

图 1 智能驾驶功能软件平台架构

感知融合功能是指在传感器抽象层的输入数据基础上进行信息融合，结合多种传感器的特性、工况和环境信息，完成对物理世界的数字呈现，作为整体环境的动态输入，是整个智能驾驶功能决策的基础。感知融合功能可以支持不同传感器的组合实现不同的感知任务，传感器组合可以是激光雷达，摄像头和毫米波雷达中的一种或者多种。感知任务包括移动目标检测与跟踪、自由空间、交通灯和交通标志识别、道路结构识别等。

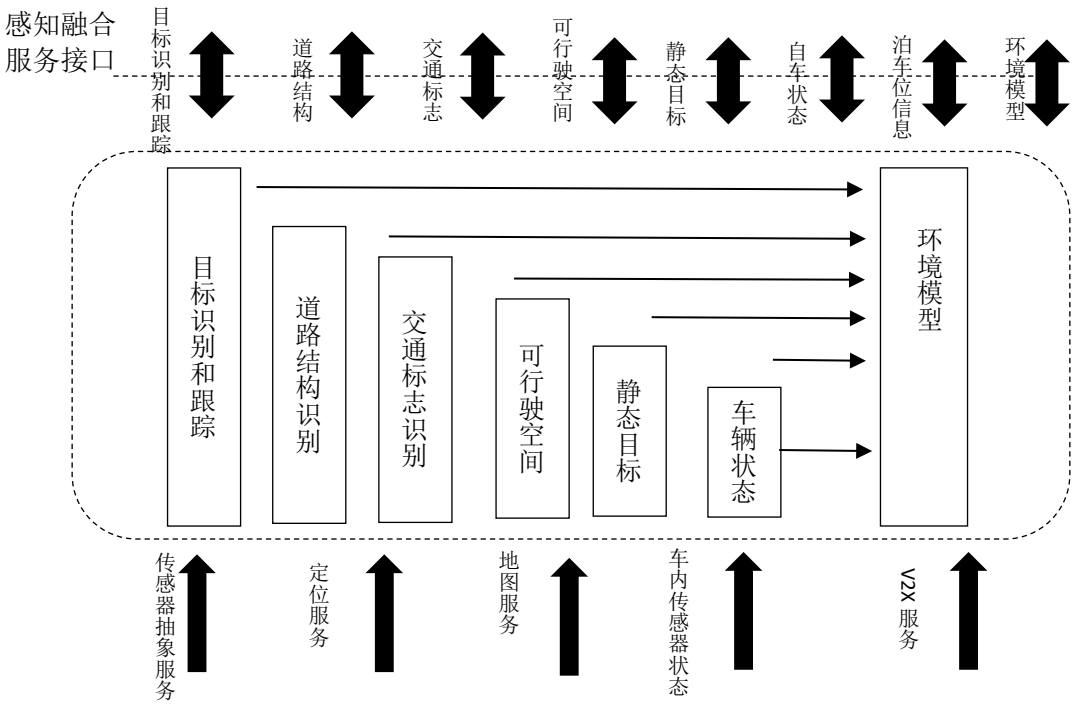

图 2 感知融合功能

按照“应用、服务”分解成单个、自包含的算法组件的原则，感知融合功能可以提供下列算法组件以及对应的服务接口：

- 目标识别和跟踪服务接口
- 道路结构服务接口
- 交通标志服务接口
- 静态目标服务接口
- 可行驶空间服务接口
- 自车状态服务接口
- 泊车位信息服务接口
- 环境模型服务接口

5 感知融合功能服务接口

本规范规定了感知融合功能对外提供的逻辑接口，定义了服务所提供信息的语义和数据类型，逻辑接口主要由数据包头和数据体两部分组成。

- 数据格式 Integer 表示整数类型，Float 表示实数类型；
- 可选必选列表明信号在接口中是否必选，M 代表必选，O 代表可选。

5.1 目标识别和跟踪服务接口

目标识别和跟踪服务接口主要提供动态和静态目标的检测和跟踪服务，包括交通参与者信息、道路标志、静态目标等信息。

表 1 目标识别和跟踪服务接口

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Header	标准元数据头	6.1			M
ListOfObjects	动静态目标数组	Object 数组		[1…MaxNoTrackObjects]	
>Object	目标元数据	6.7			M

取值范围边界	说明
MaxNoTrackObjects	最大跟踪目标数据。

5.2 道路结构服务接口

道路结构信息服务接口主要提供道路标线和停止线的识别信息，信息包括车道线类型，置信度，曲线表示，特征点集合等信息。

表 2 道路结构服务接口

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Header	标准元数据头	6.1			M
ListOfLaneLine	车道线集合数组	LaneLine 数组		[1…MaxNoLanelines]	
>LaneLine	车道线元数据	6.8			M
ListOfStopLine	停止线集合数组	StopLine 数组		[1…MaxNoStopLines]	
>StopLine	停止线元数据	6.9			M

取值范围边界	说明
MaxNoLanelines	最大检测车道线个数
MaxNoStopLines	最大检测停止线个数

5.3 交通标志服务接口

交通标志服务接口主要提供交通灯和交通标志的识别信息，主要是交通灯和交通标志中的指示信息，包括动态和静态语义信息，是当前交通规则的约束信息。接口至少需要包含一个交通灯或者交通标志的信息。

表 3 交通标志服务接口

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Header	标准元数据头	6.1			M
ListOfTrafficLight	交通灯集合数组	TrafficLight 数组		[1…MaxNoTrafficLights]	
>TrafficLight	交通灯元数据	6.10			M
ListOfTrafficSign	交通标志集合数组	TrafficSign 数组		[1…MaxNoTrafficSigns]	
>TrafficSign	交通标志元数据	6.11			M

取值范围边界	说明
MaxNoTrafficLights	最大检测交通信号灯个数
MaxNoTrafficSigns	最大检测交通标志个数

5.4 可行驶空间服务接口

可行驶空间服务接口主要提供自车附近可供自车行驶的自由空间信息，通过在车辆坐标系下的可行驶特征点和栅格信息来表示。

表 4 可行驶空间服务接口

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Header	标准元数据头	6.1			M
MetaInfo	可行驶区域范围和栅格信息	6.12			M
ListOf FreeSpaceCells	栅格集合数组	FreeSpaceCell 数组		[1…MaxNoFreeSpacCells]	
> FreeSpaceCell	栅格单元元数据	6.13			M

取值范围边界	说明
MaxNoFreeSpacCells	区域内最多栅格个数

5.5 自车状态服务接口

自车状态服务采集车辆当前状态，满足后续车辆实现动态智能驾驶业务需求。

表 5 自车状态服务接口

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Header	标准元数据头	6.1			M
vState	车辆当前关键状态数据	6.16 VehicleStatus			M

5.6 泊车位信息服务接口

泊车位信息服务接口，提供停车场区域内车位检测服务，输出泊车位信息集合。

表 7 泊车位信息服务接口

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Header	标准元数据头	6.1			M
ParkingLots	停车位集合	ParkingSpace 数组		[1…MaxNoParkingSapces]	M
>ParkingSpace	停车位数据	6.14			M

取值范围边界	说明
MaxNoParkingSapces	最大可检测车位个数

5.7 环境模型服务接口

环境、世界模型服务提供当前车辆所处环境信息以及车辆自身状态信息，这些信息是前述服务的一个集合，并在时间上进行对齐，反应在一个时间轴剖面上的车辆内外环境信息。

表 8 环境模型接口

信号名称	信号描述	数据类型	单位	取值范围	可选必选
ListOfObjects	动静态目标数组	Object 数组		[1…MaxNoTrackObjects]	
>Object	目标元数据	6.7			M
ListOfLaneLine	车道线集合数组	LaneLine 数组		[1…MaxNoLanelines]	O
>LaneLine	车道线元数据	6.8			O

ListOfStopLine	停止线集合数组	StopLine 数组		[0…MaxNoStopLines]	O
>StopLine	停止线元数据	6.9			O
ListOfTrafficLight	交通灯集合数组	TrafficLight 数组		[0…MaxNoTrafficLights]	O
>TrafficLight	交通灯元数据	6.10			O
ListOfTrafficSign	交通标志集合数组	TrafficSign 数组		[0…MaxNoTrafficSigns]	O
>TrafficSign	交通标志元数据	6.11			O
MetaInfo	可行驶区域范围和栅格信息	6.12			O
ListOf FreeSpaceCells	栅格集合数组	FreeSpaceCell 数组		[0…MaxNoFreeSpacCells]	O
> freeSpaceCell	栅格单元元数据	6.13			O
vState	车辆当前关键状态数据	6.16 VehicleStatus			M
ParkingLots	停车位集合	ParkingSpace 数组		[0…MaxNoParkingSapces]	O
>ParkingSpace	停车位数据	6.14			O

取值范围边界	说明
MaxNoParkingSapces	最大可检测车位个数
MaxNoFreeSpacCells	区域内最多栅格个数
MaxNoTrafficLights	最大检测交通信号灯个数
MaxNoTrafficSigns	最大检测交通标志个数
MaxNoLanelines	最大检测车道线个数
MaxNoStopLines	最大检测停止线个数
MaxNoTrackObjects	最大跟踪目标数据。

6 数据结构定义

6.1 标准元数据头

标准元数据包头是感知融合功能服务接口的通用包头，主要包含算法组件信息、版本信息，序列号、时间戳、采用坐标系信息、模块状态信息等。

表 9 数据包头

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Header	标准元数据头	结构体			
>ModuleID	算法组件名称	Integer	1		M
>Version ID	模块版本号	结构体			M
>>Major	主版本号	Integer	1		M
>>Minor	辅版本号	Integer	1		M
>>Patch	补丁号	Integer	1		M
>SequenceNum	序列号	Integer	1		M
>TimeStamp	时间戳	结构体			M
>>TimeStampS	时间戳整秒，格林威治时间 1970 年 01 月 01 日 00 时 00 分 00 秒起至当前时刻的时间	Integer	s		M
>>TimeStampNs	时间戳纳秒，格林威治时间 1970 年 01 月 01 日 00 时 00 分 00 秒起至当前时刻的时间	Integer	ns		M
>Frame	坐标系	枚举	1	0:NA; 1:VCS; 2:WGS84; 3:UTM	M

>Status	模块状态	枚举		0:GOOD; 1:MED; 2:FAILURE	M

6. 2 几何描述

6.2.1 二维坐标

二维坐标信息表示二维平面的点或者向量信息。

表 10 二维坐标

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Point2D	二维坐标	结构体			
>x	X 轴坐标或向量	Float			M
>y	Y 轴坐标或向量	Float			M

6.2.2 三维坐标

三维坐标信息表示三维平面的点或者向量信息。

表 11 三维坐标

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Point3D	三维坐标	结构体			
>x	X 轴坐标或向量	Float			M
>y	Y 轴坐标或向量	Float			M
>z	Z 轴坐标或向量	Float			M

6.2.3 二维多边形

二维多边形提供了多边形的二维特征点数组信息。

表 12 二维多边形

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Polygon2D	二维多边形,由一系列 2D 点构成的 Array，顺序相连；	Point2D 数组		[1..MaxNoPolygon2D Point]	
>Point2D	二维坐标	6.2.1			M

6.2.4 三维多边形

三维多边形提供了多边形的三维特征点数组信息。

表 12 三维多边形

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Polygon3D	二维多边形,由一系列 3D 点构成的 Array，顺序相连；	Point3D 数组		[1..MaxNoPolygon3D Point]	
>Point3D	三维坐标	6.2.2			M

取值范围边界	说明
MaxNoPolygon3DPoint	最大的二维多边形点个数

6.2.5 边框

边框信息主要提供物体轮廓的长宽高信息。

表 13 边框信息

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Dimension	边框	结构体			
>height	高度	Float	m	[0..300]	M
>width	宽度	Float	m	[0..300]	M
>length	长度	Float	m	[0..300]	M

6.2.6 四元数

四元数信息用来表示三维空间中的旋转信息。

表 14 四元数

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Quaternion	四元数	结构体			
>qx	X 分量	Float			M
>qy	Y 分量	Float			M
>qz	Z 分量	Float			M
>qw	W 分量	Float			M

6. 3 位置变换

位置变换信息主要提供了目标位置的平动和转动信息。

表 15 位置变换信息

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Transformation	位置变化	结构体			
>Translation	平动	Point3D 6.2.2	{m,m,m}		M
>Rotation	转动	Quaternion 6.2.6			M

6.4 位姿

位姿信息主要提供目标位姿信息，包括位置、姿态和协方差。

表 16 位姿

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Pose	位姿	结构体			
>Position	位置	Point3D 6.2.2	{m,m,m}		M
>Orientation	姿态	Quaternion 6.4			M
>Covariance	协方差	Float 数组			O

6.5 速度

速度信息提供了目标的线速度和角速度信息以及协方差信息。

表 17 速度信息

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Velocity	速度	结构体			
>linear	线速度	Point3D 6.2.2	{m/s,m/s,m/s}		M
>Angular	角速度	Point3D 6.2.2	{rad/s,rad/s,rad/s}		M
>Covariance	协方差	Float 数组			O

6.6 加速度

加速度信息提供了目标的线加速度和角加速度信息以及协方差信息。

表 18 角速度信息

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Acceleration	加速度	结构体			
>linear	线加速度	Point3D 6.2.2	{m/s^2, m/s^2, m/s^2}		M
>Angular	角加速度	Point3D 6.2.2	{rad/s^2, rad/s^2, rad/s^2}		M
>Covariance	协方差	Float 数组			O

6.7 动静态目标元数据

动静态目标元数据提供了目标识别和跟踪信息，包括目标标识，类型，存在概率，跟踪时间，几何尺寸，速度，加速度，车道关系和灯光信息等。

表 19 目标元数据

信号名称	信号描述	数据类型	单位	取值范围	可选必选
Object	目标元数据	结构体			
>objectId	目标 ID	Integer	1		M
>objectType	目标类型	枚举	1	1:Pedestrian; 2:Cyclist; 3:Car; 4:Truck; 5:Tram; 6:Motorcycle; 7:Bicycle; 8:Tricycle; 9:TrafficLight; 10:TrafficSign; 11:RoadSign; 12:TrafficCone; 13:TrafficStick; 14:BillBoard; 15:ParkingSlot;	M

				16:FireHydrant; 17:Unknow_static; 18:Unknow_movable;	
>classConfidence	类型置信度	Float	%	[0..100]	M
>isMovable	静态、动态目标区分	Bool	1	0：静态目标 1：潜在动态目标	M
>existProbability	存在概率	Float	%	[0..1]	O
>timeLastSeen	目标上次更新时间	Float	s		O
>timeCreation	目标创建时刻	Float	s		O
>objectAge	目标被跟踪的时间	Float	s		O
>trackQuality	跟踪质量	Float	%	[0..1]	O
>referencePoint	目标包围框中心位置，相对自车坐标系；长方体质心；	Pose 6.4			M
>objectDimension	目标几何尺寸	Dimension 6.2.5			M
>ObjectVelocity	目标速度相对自车坐标系；	Velocity 6.5			C 动态目标
>ObjectAccel	目标加速度相对自车坐标系；	Acceleration 6.6			C 动态目标
>laneAssignment	车道分配，对动态目标所属车道进行估计	Integer	1	[-31..31] 本车行驶车道为 0，左侧车	O

				道为负，右侧车道为正	
>laneOffsetLeft	距左侧车道偏移量	Float	m	[0..10]	O
>laneOffsetRight	距右侧车道偏移量	Float	m	[0..10]	O
>LightStatus	转向灯、刹车灯状态、紧急指示灯	枚举	1	0:无； 1:左转； 2:右转； 3:双闪； 4:刹车	O

6.8 车道线元数据

车道线元数据主要提供车道线的类型、置信度、双曲线表示信息等。

表 20 车道线元数据

信号名称	信号描述	数据类型	单位	取值范围	可选必选
LaneLine	车道线元数据	结构体			
>Type	车道线类型：双黄线、单黄线、实线、虚线等	枚举		0:WHITE_DASHED; 1:WHITE_SOLID; 2:YELLOW_DASHED; 3:YELLOW_SOLID; 4: UNKNOWN;	M
>confidence	车道线置信度	Float		[0..1]	M
>linePoints	车道线点列表	Point3D 数组		[1..MaxNoLinePoints]	O
>>Point3D	车道线点	6.2.2			M

取值范围边界	说明
MaxNoLinePoints	最大的车道线点个数

6.9 停止线元数据

停止线元数据提供了停止线类型、置信度，起点和终点的信息。

表 21 停止线元数据

信号名称	信号描述	数据类型	单位	取值范围	可选必选
StopLine	停止线元数据	结构体			
>Type	停止线检测类型	枚举		0: NORMAL; 1: UNKNOWN;	M
>Confidence	类型置信度	Float	%	[0..100]	M
>LeftStart	停止线左起点	Point3D 6.2.2			M
>RightEnd	停止线右终点	Point3D 6.2.2			M

6.10 交通灯元数据

交通灯元数据提供了交通灯的类型、颜色、位置、剩余时间信息。

表 22 交通灯元数据

信号名称	信号描述	数据类型	单位	取值范围	可选必选
TrafficLight	交通灯元数据	结构体			
>Id	红绿灯标号	Integer	1		M
>Type	交通灯类型	枚举		0: STRAIGHT; 1: LEFT; 2: RIGHT; 3: U_TURN;	M
>Colour	交通灯颜色	枚举		0:RED; 1:YELLOW; 2:GREEN; 3:BLACK; 4:UNKNOWN;	M
>Position	红绿灯位置，地图坐标系，灯框	Point3D 6.2.2			M

	外轮廓左上角				
>Box	灯框的外轮廓	Polygon2D 6.2.3			M
>LaneAttached	对应车道	Integer	1	[-31..31]{本车行驶车道为 0，左侧车道为负，右侧车道为正}	M
>RemainingTime	剩余时间	Float	s		O

6.11 交通标志元数据

交通标志元数据信息包括了交通标志的类型、位置和对应车道。

表 23 交通标志元数据

信号名称	信号描述	数据类型	单位	取值范围	可选必选
TrafficSign	交通标志元数据	结构体			
>Id	交通标志标号	Integer	1		M
>Sematic	交通标志类型	枚举		0: SpeedLimit; 1: HzardWarning; …	M
>Value	值	Integer			O
>Unit	单位	枚举		0: km/h; 1: ton; …	O
>Position	位置，地图坐标系	Point3D 6.2.2			M
>Box	交通标志的外包盒	Polygon2D 6.2.3			M
>LaneAttached	对应车道	Integer	1	[-31..31]{本车行驶车道为 0，左侧车道为负，右侧车道为正}	O

6.12 栅格信息描述

车辆坐标系下，构建栅格描述自由空间：格栅的范围是汽车左30米，右30米，前250米，后80米。

表 24 栅格信息

信号名称	信号描述	数据类型	单位	取值范围	可选必选
FreeSpaceMeta	栅格划分信息描述	结构体			
>XStart	X 方向起点	Float	m		M
>XEnd	X 方向终点	Float	m		M
>XReslution	X 方向栅格解析度	Float	m/grid		M
>XGridNum	X 方向有多少个栅格	Integer	m/grid		M
>YStart	Y 方向起点	Float	m		M
>YEnd	Y 方向终点	Float	m		M
>YReslution	Y 方向栅格解析度	Float			M
>XGridNum	Y 方向有多少个栅格	Integer			M

6.13 栅格单元信息

栅格单元信息表明栅格单元的占用状态和占用概率。

表 25 栅格单元信息

信号名称	信号描述	数据类型	单位	取值范围	可选必选
FreeSpaceCell	栅格单元信息	结构体			
>Status	栅格占用状态	枚举		0:UNKNOW; 1:FREE; 2:OCC_STATIC; 3:OCC_MOVABLE; 分别代表未知状态、可行驶、固定障碍物占用、移动障碍物占用	M
>Probability	占用概率	Float	%	[0..100]	M

6.14 泊车位元数据

泊车位元数据表明泊车位的类型，位置，置信度等特征信息。

表 26 泊车位元数据信息

信号名称	信号描述	数据类型	单位	取值范围	可选必选
ParkingSpace		结构体			
>Id	停车位 ID，同一停车区域描述唯一；	Integer	1		M
>Ptype	停车位类型	枚举		0:PARKING_TYPE_UNKNOWN; 1:PARKING_TYPE_VERTICAL; 2:PARKING_TYPE_PARALLEL; 3:PARKING_TYPE_INC;	M
>TypeConfidence	类型置信度	Float64	%	[0..100]	M
>ParkingSpotPoints	泊车位构成点列，车辆坐标系下	Point3D 数组 6.2.2	{m,m,m}		M
>Width	停车位宽度	Float64	m	[0..100]	O
>Length	停车位长度	Float64	m	[0..100]	O
>Valid	是否为有效车位	Bool		0：无效 1：有效	M

6.15 车辆状态元数据

车辆状态元数据表明车辆的状态信息，包括转向信息、速度信息、加速度信息、控制模式、发动机状态等。

表 27 车辆状态元数据

信号名称	信号描述	数据类型	单位	取值范围	可选必选
VehicleStatus	标准元数据头	结构体			

SteeringAngle	转向角	Float	角度	-1080° ~1080°	M
steeringVelocity	转向速率	Float	° /s	车型决定， -1080° /s~1080° /s	M
Acceleration	加速度信息	6.6	{m/s^2,m/s^2， m/s^2，}		M
Velocity	速度信息	6.5	{m/s,m/s,m/s}		M
ControlMode	当前控制模式	枚举	1	MANUAL=0; AUTO=1; LAT_ONLY=2; LON_ONLY=3;	M
GearMode	当前档位	枚举	1	GEAR_NEUTRAL = 0; GEAR_PARKING = 1; GEAR_DRIVE = 2; GEAR_REVERSE = 3; GEAR_LOW = 4; GEAR_INVALID = 5; GEAR_NONE = 6;	M
eMode	发动机状态	枚举	1	ON=0; Off=1； FAILURE =2；	M
WheelSpeed	左前、右前、左后、右后轮速	Float 数组			M

附 录 A
（资料性附录）
感知融合服务接口 protobuf 描述

A.1 数据结构定义描述

```
// common.proto
message Header{
    required uint32 ModuleID=1;
    message VersionID{
        required uint32 major=1;
        required uint32 minor=2;
        required uint32 patch=3;
    }
    required VersionID vid=2;
    required uint32 sequenceNum=3;
    message timeStamp{
        required uint64 timeStampS=1;
        required uint64 timeStampNs=2;
    }
    required timeStamp TimeStamp=4;
    enum FRAMETYPE{
        NA=0;
        VCS=1;
        WGS84=2;
        UTM=3;
    }
    required FRAMETYPE Frame=5;
    emum STATUS{
        GOOD=0;
        MED=1;
        FAILURE=2;
    }
    required STATUS Status =6;
}

message Point2D{
    required double x=1;
    required double y=2;
    }

message Point3D{
    required double x=1;
    required double y=2;
    required double z=3;
    }

message Polygon2D{
    repeated point2D p2d=1;
    }

message Polygon3D{
    repeated point3D p3d=1;
    }

message Dimension{
    required double height=1;
    required double width=2;
    required double length=3;
    }

message Quaternion {
    required double qx = 1 ;
    required double qy = 2 ;
    required double qz = 3 ;
    required double qw = 4 ;
    }

message Transformation{
    required Point3D Translation=1;
```

```
    required Quaternion Rotation=2;
    }

message Pose{
    required Point3D Position=1;
    required Quaternion Orientation=2;
    required double Covariance=3;
    }

message Velocity{
    required Point3D Linear=1;
    required Point3D Angular =2;
    repeated double Covariance=3;
    }

message Acceleration{
     required Point3D Linear=1;
     required Point3D Angular =2;
     repeated double Covariance=3;
     }

message Object{
     required uint32 objectId=1;
     enum OBJECTYPE{
        Pedestrian=1;
        Cyclist=2;
        Car=3;
        Truck=4;
        Tram=5;
        Motorcycle=6;
        Bicycle=7;
        Tricycle=8;
        TrafficLight=9;
        TrafficSign=10;
        RoadSign=11;
        TrafficCone=12;
        TrafficStick=13;
        BillBoard=14;
        ParkingSlot=15;
        FireHydrant=16;
        Unknow_static=17;
        Unknow_movable=18;
             }
     required OBJECTYPE objectType=2;
     required double classConfidence=3;
     required bool isMovable=4;
     optional double existProbability=5;
     optional double timeLastSeen=6;
     optional double timeCreation=7;
     optional double objectAge=8;
     optional double trackQuality=9;
     required pose referencePoint=10;
     required Dimension objectDimension=11;
     required Velocity ObjectVelocity=12;
     required Acceleration ObjectAccel=13;
     optional int32 laneAssignment=14;
     optional double laneOffsetLeft=15;
     optional double laneOffsetRight=16;
     enum LIGHTSTATUS{
        OFF=0;
        LEFT=1;
        RIGHT=2;
        BLINK=3;
        BRAKE=4;
        }
    Optional LIGHTSTATUS LightStatus =17;
}

message LaneLine{
    enum LANELINETYPE{
        WHITE_DASHED = 0;
```

```
        WHITE_SOLID = 1;
        YELLOW_DASHED = 2;
        YELLOW_SOLID = 3;
        UNKNOWN=4;
      }
    required LaneLineType Type=1;
    required double confidence=2;
    repeated Point3D linePoints=3;
}

message StopLine{
    enum STOPLINETYPE {
        NORMAL=0;
        UNKNOWN=1;
         }
    required STOPLINETYPE Type=1;
    required double confidence=2;
    required Point3D LeftStart=3;
    required Point3D RightEnd=4;
    }

message TrafficLight{
     required uint32 id=1;
    enum TLType {
        STRAIGHT = 0;
        LEFT = 1;
        RIGHT = 2;
        U_TURN = 3;
        }
     required TLtype type=2;
     enum TLColour{
        RED = 0;
        YELLOW = 1;
        GREEN = 2;
        BLACK = 3;
        UNKNOWN = 4;
         }
    required TLColour Colour =3;
    required Point3D Position=4;
    required Polygon2D Box=5;
    required int32 LaneAttached=6;
    optional double RemainingTime=7;
    }

message TrafficSign{
    required uint32 id=1;
    enum TSTYPE{
        SpeedLimit=0;
        HzardWarning=1;
         }
     required TSTYPE Semantic=2;
     optional uint32 value=3;
     enum UNITTYPE{
        KMPH=0;
        TON=1;
         }
    optional UNITTYPE Unit=4;
    required Point3D position=5;
    required Polygon2D Box=6;
    optional int32 LaneAttached=7;
}
message FreeSpaceMeta{
    required double xStart=1;
    required dboule xEnd=2;
    required dboule xReslution =3;
    required uint32 xGridNum =4;
    required double yStart=5;
    required dboule yEnd=6;
    required dboule yReslution =7;
    required uint32 yGridNum =8;
}
```

```
message FreeSpaceCell{
     enum CELLSTATUS {
        UNKNOW=0;
        FREE=1;
        OCC_STATIC=2;
        OCC_MOVABLE=3;
         }
    required CELLSTATUS Status=1;
    required double Probability=2;
}

message ParkingSpace{
    required uint32 id=1;
    enum PARKINGTYPE{
        PARKING_TYPE_UNKNOWN = 0;
        PARKING_TYPE_VERTICAL = 1;
        PARKING_TYPE_PARALLEL = 2;
        PARKING_TYPE_INC      = 3;
        }
    required PARKINGTYPE Ptype=2;
    required double typeConfidence=3;
    repeated Point3D ParkingSpotPoints=4;
    optional double width=5;
    optional double length=6;
    required bool valid=7;
    }

message VehicleStatus{
     required double SteeringAngle=1;
     required double steeringVelocity=2;
     required Acceleration acc=3;
     required Velocity vel=4;
     enum DrivingMode{
            MANUAL=0;
            AUTO=1;
            LAT_ONLY=2;
            LON_ONLY=3;
     }
     required DrivingMode ControlMode=5;
     enum GearPosition{
            GEAR_NEUTRAL = 0;
            GEAR_PARKING = 1;
            GEAR_DRIVE = 2;
            GEAR_REVERSE = 3;
            GEAR_LOW = 4;
            GEAR_INVALID = 5;
            GEAR_NONE = 6;
     }
     required GearPosition GearMode=6;
     enum EngineMode{
            ON=0;
            OFF=1;
            FAILURE=2;
     }
     required EngineMode eMode =7;
     repeated double WheelSpeed=8;
}
```

A.2 目标识别和跟踪服务接口

```
// ObjectDetectionAndTrackingService.proto
import common.proto
message ObjectDetectionAndTrackingService{
    required Header header=1;
    repeated Object ListOfObjects=1;
    }
```

A.3 道路结构服务接口

```
// RoadStructureService.proto
import common.proto
message RoadStructureService{
    required Header header=1;
    repeated LaneLine ListOfLaneLine=2;
    repeated StopLine ListOfStopLine =3;
    }
```

A.4 交通标志接口

```
// TrafficSignLightService.proto
message TrafficSignLightService{
    required Header header=1;
    repeated TrafficLight ListOfTrafficLight=2;
    repeated TrafficSign ListOfTrafficSign=3;
        }
```

A.5 可行驶空间接口

```
// DrivableAreaService.proto
import common.proto
message DrivableAreaService{
    required Header header=1;
    required FreeSpaceMeta MetaInfo=2;
    repeated FreeSpaceCell ListOfFreeSpaceCells=3;
        }
```

A.6 车辆状态服务接口

```
//VehicleStatusService.proto
import common.proto
message VehicleStatusService{
    required Header header=1;
    required VehicleStatus vState=2;
    }
```

A.7 泊车位信息服务接口

```
// ParkinglotDetectionService.proto
import common.proto
message ParkinglotDetectionService {
    required Header header=1;
    repeated ParkingSpace ParkingLots =2;
        }
```

A.7 环境模型服务接口

```
import common.proto
message ParkinglotDetectionService {
    required Header header=1;
    repeated Object ListOfObjects=2;
    repeated LaneLine ListOfLaneLine=3;
    repeated StopLine ListOfStopLine =4;
    repeated TrafficLight ListOfTrafficLight=5;
    repeated TrafficSign ListOfTrafficSign=6;
    optional FreeSpaceMeta MetaInfo=7;
    repeated FreeSpaceCell ListOfFreeSpaceCells=8;
    required VehicleStatus vState=9;
    repeated ParkingSpace ParkingLots =10;
        }
```