

高精地图数据引擎 使用说明

Baidu Confidential

修改记录

版本号	修订日期	修订记录	修订人	审核人	状态	链接
V1.0	2020/03/16	引擎使用说明	张永乐	王健	发布	

Baidu Confidential

目录

1 编译说明.....	1
2 数据结构.....	1
2.1 车道边线信息(LaneMarking).....	1
2.2 实体信息 (Object)	2
2.3 车道信息 (Lane)	4
2.3 道路信息 (Link)	6
2.4 车道拓扑信息(LaneTopology).....	7
2.5 道路拓扑信息(LinkTopoLogy).....	7
2.6 ADAS 信息(ADAS_Value)	7
2.7 三维点信息(Point3D).....	8
2.8 WGS84 坐标(WGS84Coord).....	8
2.9 GPS 坐标(GPS_Coord).....	8
2.10 WGS84 包围盒(WGS84GeoBox)	8
2.11 地理围栏信息(Fence).....	8
2.12 高精地图错误码定义	10
3 API 接口定义	10
3.1 初始化 db 文件.....	10
3.2 获取所有道路信息.....	11
3.3 获取所有车道信息.....	11
3.4 获取所有道路拓扑信息	11
3.5 获取所有车道拓扑信息	11
3.6 获取所有 Object 信息.....	12
3.7 获取所有 Fence 信息.....	12
3.8 测试勾子函数.....	12
4 API 接口调用示范	12

1 编译说明

高精数据引擎提供动态库及头文件，支持 Windows 系统。

平台	操作系统	库文件	头文件	编译参数
win_x86_64	Windows	hadapi.lib	had_feature_define.h had_interface.h	-D__hwin__

2 数据结构

2.1 车道边线信息(LaneMarking)

Struct	Member Name	Comment	Data Type	Length	Description
LaneMarking	lane_marking_id	车道线 ID	int		取当前车道
	side	车道线左右标志	int		1:right; 0:left; 2:unkonwn
	lane_marking_type	车道线类型	int		NONE= 0, LONG_DASHED_LINE= 1, DOUBLE_SOLID_LINE= 2, SINGLE_SOLID_LINE= 3, RIGHT_SOLID_LINE_LEFT_DASHED_LINE = 4, LEFT_SOLID_LINE_RIGHT_DASHED_LINE = 5, DOUBLE_DASHED_LINE = 9, UNKNOWN = (-9999)
	color	车道线颜色	int		COLOR_NONE = 0, COLOR_YELLOW = 1, COLOR_WHITE = 2, COLOR_BLUE = 10, COLOR_ORANGE = 11, COLOR_WHITE_YELLOW = 12, COLOR_YELLOW_WHITE = 13

	material	车道线材质	int		UN_KNOWN = 0, METAL = 1, CONCRETE = 2, STONE = 3, WOOD = 4, PLASTIC = 5, TRANSPARENT1 = 6, VIBRATION_MARKINGS = 7, PAINTED_VIBRATION_DI VIDER = 8
	width	车道线的宽 度	double		目前设为 0
	Sequence	车道边线序 号	Int		>=0, -1 无效
	length	车道边线长 度	double		>=0
	geometry	车 前 200M 车道边线几 何信息	Point3D	变长	具体参见 Point3D 定 义。
	order_number	边线打断序 号。指同一 车道同一类 型的序号， 如同道路的 护栏按 1...n 排列， 路沿按 1...n 排列	int		
	offset	上述每一段 首点距离起 点的偏移距 离	Double		

2.2 实体信息（Object）

Struct	Member Name	Comment	Data Type	Length	Description
Object	id	物体 ID	int		

	type	物体类型	int		<p>OBJECT_TYPE_POLE = 1, //路灯等</p> <p>OBJECT_TYPE_LinkMARK = 2, //道路地面标线</p> <p>OBJECT_TYPE_SIGN = 3, //交通标志, 如限速</p> <p>OBJECT_TYPE_BUILDING = 4, //路边建筑, 如电压塔</p> <p>OBJECT_TYPE_GANTRY = 5 //龙门架等</p>
	subtype	物体子类型	int		<p>OST_NO_SUB_TYPE = 0,</p> <p>OST_ROADMARK_ARROW = 1, //导向箭头</p> <p>OST_ROADMARK_SHADEAREA = 2, //导流区</p> <p>OST_ROADMARK_CHARACTER = 3, //文字</p> <p>OST_SIGN_SPEEDLIMIT = 4 //限速</p>
	center_x	物体中心经度坐标	double		>=-180.0, <=180.0
	center_y	物体中心纬度坐标	double		>=-90.0, <=90.0
	center_z	物体中心高度坐标	double		NA
	height	物体高度	float		>=0.0
	width	物体宽度	float		>=0.0
	length	物体长度	float		>=0.0
	heading	物体方向	float		>=-180.0, <=180.0

	b_has_geometry	是否有几何信息	bool		1: 有几何信息; 0: 无几何信息
	geometry	几何信息	Point3D		参见 Point3D
	bounding_box_geometry	保卫盒几何信息	Point3D		参见 Point3D
	laneid	object	std::set<int>		空或者有效的 ID

2.3 车道信息 (Lane)

Struct	Member Name	Comment	Data Type	Length	Description
Lane	laneid	车道 ID	int		
	linkid	道路 ID	int		
	sequence	车道序号	int		以车行方向为参照, 最右侧车道为 1
	Lane_type	车道类型			LAT_NORMAL = 0, LAT_ENTRY = 1, LAT_EXIT = 2, LAT_EMERGENCY = 3, LAT_ON_RAMP = 4, LAT_OFF_RAMP = 5, LAT_CONNECT_RAMP = 6, LAT_ACCELERATE = 7, LAT_DECELERATE = 8, LAT_EMERGENCY_PARKING_STRIP = 9, LAT_DIVERSION = 17
	lane_status	车道状态			LAN_STATUS_NORMAL = 0x000000, //Lane is normal. LAN_STATUS_INBUILDING = 0x000001, //Lane is in building. LAN_STATUS_FORMING = 0x000002, //Lane is new forming.

					<p>LAN_STATUS_ENDING = 0x000004, //Lane is new ending.</p> <p>LAN_STATUS_RAMP_BROKEN = 0x000008, //Lane is not connect normal</p> <p>LAN_STATUS_SPLITTING = 0x000010, //Lane is splited from another lane</p> <p>LAN_STATUS_MERGING = 0x000020 //Lane is to be merged to another lane</p>
	travel_direction 方向				<p>IN_POSITIVE_DIRECTION = 1,</p> <p>IN_NEGATIVE_DIRECTION = 2,</p> <p>IN_BOTH DIRECTIONS = 3</p>
	min_speed_limit	最小限速值	double		KM/H
	max_speed_limit	最大限速值	double		KM/H
	width	车道平均宽度	double		
	length	车道长度	double		
	vehiche_type	车辆类型	Int		
	marking_list	车道边线	LaneMarking		参见 LaneMarking
	geometry	车道中心线	Point3D		参见 Point3D 定义
	successor_lane_ids	后继车道 ID 序列	int	变长	
	precursor_lane_ids	前驱车道 ID 序列	Int	变长	
	obj_ids	本车道 ObjectID	Int	变长	

		序列			
	left_lane_id	左车道 ID	Int		
	right_lane_id	右车道 ID	Int		
	adas_info	本 车 道 ADAS 信息, 基于 1M 间 隔计算出的 ADAS 信息	Link_ADAS		
	min_width	本车道最窄 处宽度	Double		
	max_width	本车道最宽 处宽度	Double		

2.3 道路信息 (Link)

Struct	Member Name	Comment	Data Type	Length	Description
Link	id	道路 ID	int		
	link_class	道路等级	int		LC_EXPRESSWAY = 1, //highway LC_URBAN_EXPRESSWAY = 2, // LC_NORMAL_LINK = 3
	travel_direction	道路方向	int		IN_POSITIVE_DIRECTION = 1, IN_NEGATIVE_DIRECTION = 2, IN_BOTH DIRECTIONS = 3
	Link_type	道路类型			LT_NO_SPECIAL = 0, //normal link LT_RAMP = 1 //ramp
	lane_num	车道数量	int		
	b_tunnel	是否隧道	Bool		
	b_bridge	是否桥梁	Bool		
	b_toll	是否收费道路	Bool		
	b_toll_booth	是否收费站	Bool		

	b_broken_head	是否断头路	bool		
	length	道路长度	Double		
	speed_limit	最大限速值	Int		KM/H
	vehiche_type	车辆类型	int		
	geometry	道路几何信息	Point3D	变长	参见结构体 Point3D
	adas_info	道路 ADAS 信息。基于 1M 间隔，计算出的 ADAS 信息。	Link_ADAS	变长	参见结构体 Link_ADAS
	successor_link_ids	后继道路 ID	Int	变长	
	precursor_link_ids	前驱道路 ID	Int	变长	
	lane_ids	包含车道 ID	int	变长	

2.4 车道拓扑信息(LaneTopology)

Struct	Member Name	Comment	Data Type	Length	Description
LaneTopology	laneid_from	前驱车道 ID	int		
	linkid_from	前驱道路 ID	int		
	laneid_to	后继车道 ID	int		
	linkid_to	后继道路 ID	int		

2.5 道路拓扑信息(LinkTopoLogy)

Struct	Member Name	Comment	Data Type	Length	Description
LinkTopology	linkid_from	前驱道路 ID	int		
	linkid_to	后继道路 ID	int		
	junction_point	连接点	Point3D		参见 Point3D 结构体定义

2.6 ADAS 信息(ADAS_Value)

Struct	Member Name	Comment	Data Type	Length	Description
ADAS_Value	curvature	曲率	double		≥ 0.0
	heading	航向	double		≥ 0.0 , ≤ 360.0
	slope	坡度	double		≥ -90.0 , ≤ 90.0
	offset	偏移距离	Int		相对当前车道、道

					路起点的偏移距离 (M)
--	--	--	--	--	--------------

2.7 三维点信息(Point3D)

Struct	Member Name	Comment	Data Type	Length	Description
Point3D	x		double		经度, ≥ -180.0 , ≤ 180.0
	y		double		纬度, ≥ -90.0 , ≤ 90.0
	z		double		高度, NA

2.8 WGS84 坐标(WGS84Coord)

Struct	Member Name	Comment	Data Type	Length	Description
WGS84Coord	x		double		经度, ≥ -180.0 , ≤ 180.0
	y		double		纬度, ≥ -90.0 , ≤ 90.0
	z		double		高度, NA

2.9 GPS 坐标(GPS_Coord)

Struct	Member Name	Comment	Data Type	Length	Description
GPS_Coord	lon		double		经度, ≥ -180.0 , ≤ 180.0
	lat		double		纬度, ≥ -90.0 , ≤ 90.0
	height		double		高度, NA
	heading		double		≥ -180.0 , ≤ 180.0

2.10 WGS84 包围盒(WGS84GeoBox)

Struct	Member Name	Comment	Data Type	Length	Description
WGS84GeoBox	min_coord		WGS84Coord		包围盒最小坐标, 即左下坐标
	max_coord		WGS84Coord		包围盒最大坐标, 即右上坐标

2.11 地理围栏信息(Fence)

Struct	Member Name	Comment	Data Type	Length	Description
Fence	linkid	道路 ID	unsigned int		

	laneid	车道 ID	unsigned int		
	start_offset	围栏起点偏移	double		单位 M
	end_offset	围栏终点偏移	double		单位 M
	fence_type	详见右侧描述			FT_INVALID = 0, FT_UNCONFIRMED = 1, FT_UNPLANNED_AREA = 2, FT_TUNNEL = 3, FT_TOLLGATE = 4, FT_RAMP_JCT = 5, FT_CONSTRUCTION = 6, FT_TRAFFIC_LIGHT = 7, FT_NO_GUARDRAIL = 8, FT_HIGHWAY_END = 9, FT_ALL_LANES_CHANGE = 10, FT_LINK_CURVATURE = 11, FT_LINK_SLOPE = 12, FT_TRANSITION_AREA = 13, FT_LANE_TO_RAMP = 14, FT_LANE_FORMING_ENDING = 15, FT_LANE_WIDTH = 16, FT_LANE_CURVATURE = 17, FT_LANE_MARKING_BROKEN = 18, FT_LANE_EMERGENCY = 19, FT_RESERVATION = 99
	is_link		bool		是否道路标志, true 为道路; false 非道路
	is_in_fence		Bool		是否在地理围栏内。true 为围栏内; false 围栏外
	id		unsigned int		地理围栏 ID

unsigned int unsigned int linkid;

```
unsigned int laneid;
double start_offset;
double end_offset;
int fence_type;
bool is_link;
bool is_in_fence;
unsigned int id;;
unsigned int laneid;
double start_offset;
double end_offset;
int fence_type;
bool is_link;
bool is_in_fence;
unsigned int id;
```

2.12 高精地图错误码定义

```
HD_OK = 0,
HD_DB_OPEN_FAILED = 1,
HD_DB_NOT_COMPATIBLE = 2,
HD_INVALID_POS = 3,

//licence related
HD_LICENCE_NOT_INITIALIZE = 100,
HD_ACTIVATE_FAILED = 101,
HD_LICENCE_NOT_FOUND = 102,
HD_LICENCE_EXPIRED = 103,
HD_LICENCE_IO_ERROR = 104,

//net related
HD_NET_CONNECT_FAILED = 200,
HD_NET_TIMEOUT = 201,

HD_UNKOWN_ERROR = 9999
```

3 API 接口定义

本节函数定义详见代码注释。

3.1 初始化 db 文件

```
int initialize(const std::string &dbpath, std::set<int>& region_code, std::string &error)
```

输入 dbpath: 地图数据路径

输入 region_code 城市行政区划编码列表

输出 error 返回错误信息

返回: 错误状态。0 为成功; 其他值为错误, 详见“高精地图错误编码定义”章节。

注: 该函数负责加载指定 region_code 的地图信息到缓存中, 下面的接口分别获取不同的地图要素信息。

3.2 获取所有道路信息

```
bool get_all_link_info(  
    std::vector<had::Link>& link_map)
```

输入: 无

输出 link_map: 道路信息

返回: 至少找到一条 link 信息, 则返回 true; 否则, 返回失败

3.3 获取所有车道信息

```
bool get_all_lane_info(  
    std::vector<had::Lane>& lane_map)
```

输入: 无

输出 lane_map: 车道信息

返回: 至少找到一条 lane 信息, 则返回 true; 否则, 返回失败

3.4 获取所有道路拓扑信息

```
bool get_all_link_topology(  
    std::vector<had::LinkTopology>& link_topology_vec)
```

输入: 无

输出 link_topology_vec: 道路拓扑信息

返回: 至少找到一条 LinkTopology 信息, 则返回 true; 否则, 返回失败

3.5 获取所有车道拓扑信息

```
bool get_all_lane_topology(  
    std::vector<had::LaneTopology>& lane_topology_vec)
```

输入: 无

输出 lane_topology_vec: 道路拓扑信息

返回:至少找到一条 LaneTopology 信息, 则返回 true; 否则, 返回失败

3.6 获取所有 Object 信息

```
bool get_all_object_info(  
    std::vector<had::Object>& object_map)
```

输入: 无

输出 object_map: Object 信息

返回:至少找到一个 Object 信息, 则返回 true; 否则, 返回失败

3.7 获取所有 Fence 信息

```
bool get_all_fence_info(  
    std::vector<had::Fence>& fence_vec)
```

输入: 无

输出 fence_vec: Fence 信息

返回:至少找到一个 Fence 信息, 则返回 true; 否则, 返回失败

3.8 测试勾子函数

```
void engine_hook(std::vector<int> & input_para,  
    std::vector<had::GPS_Coord>& input_coord)
```

输入 input_para: 测试用 ID 序列

输入/输出 input_coord: 测试坐标序列, 或返回坐标序列

保留接口, 暂不开放使用。

4 API 接口调用示范

```
#include "had_interface.h"
#include "had_feature_define.h"
.....
//城市行政区划编码，指地图数据目录中该城市数据所在子目录名（数字）
int region = 110000;
//地图数据所在根目录
std::string str_datapath = "./beijing_bxf";
std::string err_info;

had::IHadDB* _pdb = get_haddb();
_pdb->initialize(str_datapath, region, err_info);

std::vector<had::Link> link_vec;
std::vector<had::Fence> fence_vec;

_pdb->get_all_link_info(link_vec);
std::cout << "link size =" << link_vec.size() << std::endl;
_pdb->get_all_fence_info(fence_vec);
std::cout << "fence size =" << fence_vec.size() << std::endl;
```