GUIDANCE SDK-API 接口名称和参数

image\_data

描述: 定义图像的数据结构。每个方向的深度图与双目灰度图中的左图对齐。

typedef struct \_image\_data

{

    unsigned int     frame\_index;                   // frame index

    unsigned int     time\_stamp;                    // time stamp of image captured in ms

    char \*m\_greyscale\_image\_left[CAMERA\_PAIR\_NUM];// greyscale image of left       camera

    char \*m\_greyscale\_image\_right[CAMERA\_PAIR\_NUM];   // greyscale image of right camera

    char   \*m\_depth\_image[CAMERA\_PAIR\_NUM];          // depth image in \*128 meters

    char  \*m\_disparity\_image[CAMERA\_PAIR\_NUM];      // disparity image in \*16 pixels

}image\_data;

解释:

1、m\_greyscale\_image\_left和m\_greyscale\_image\_right都是宽320，高240的8比特灰度图。

2、m\_depth\_image是宽320，高240的16比特深度图，每两个字节描述一个点的深度，低7位为小数位，高9位为整数位。

3、m\_disparity\_image是宽320，高240的16比特视差图，每两个字节描述一个点的深度，低4位为小数位，高12位为整数位。

VI、ultrasonic\_data

描述: 定义超声波的数据结构。ultrasonic是超声波传感器检测到的最近物体的距离，单位是mm。reliability是该距离测量的可信度，1为可信，0为不可信。由于观测数据存在噪声，建议对数据进行滤波后再使用。

typedef struct \_ultrasonic\_data

{

    unsigned int     frame\_index;    // corresponse frame index

    unsigned int     time\_stamp;     // time stamp of corresponse image captured in ms

    short        ultrasonic[CAMERA\_PAIR\_NUM];            // distance in mm. -1 means invalid measurement.

    unsigned short   reliability[CAMERA\_PAIR\_NUM];   // reliability of the distance data

}ultrasonic\_data;

VII、velocity

描述: 定义体坐标系下的速度。单位是mm/s。

typedef struct \_velocity

{

    unsigned int     frame\_index;        // corresponse frame index

    unsigned int     time\_stamp;         // time stamp of corresponse image captured in ms

    short            vx;                 // velocity of x in mm/s

    short            vy;                 // velocity of y in mm/s

    short            vz;                 // velocity of z in mm/s

}velocity;

VIII、obstacle\_distance

描述: 定义由视觉和超声波融合得到的障碍物距离。单位是cm。

typedef struct \_obstacle\_distance

{

    unsigned int     frame\_index;    // corresponse frame index

    unsigned int     time\_stamp;     // time stamp of corresponse image captured in ms

    unsigned short   distance[CAMERA\_PAIR\_NUM];  // distance of obstacle in cm

}obstacle\_distance;

IX、imu

描述: 定义IMU数据结构。加速度单位为m/s^2

typedef struct \_imu

{

    unsigned int     frame\_index;             //相应的帧索引

    unsigned int     time\_stamp;           //以ms为单位捕获的对应图像的时间戳

    float            acc\_x;                   // acceleration of x in unit of m/s^2

    float            acc\_y;                   // acceleration of y in unit of m/s^2

    float            acc\_z;                   // acceleration of z in unit of m/s^2

    float            q[4];                    // quaternion: [w,x,y,z]

}imu;

X、stereo\_cali

描述: 摄像头的标定参数。如果某个方向的传感器不在线，则所有值为0.

typedef struct \_stereo\_cali

{

    float cu;                // 焦点中心的x位置，以像素为单位

    float cv;                // 焦点中心的y位置，以像素为单位

    float focal;             // 焦距以像素为单位

    float baseline;        // 以米为单位的立体相机基线

    \_stereo\_cali() { }

    \_stereo\_cali(float \_cu, float \_cv, float \_focal, float \_baseline)

    {

        cu = \_cu, cv = \_cv;

        focal = \_focal, baseline = \_baseline;

    }

}stereo\_cali;

XI、exposure\_param

描述: 摄像头的曝光参数。当m\_expo\_time = m\_expected\_brightness=0时，变成默认的自动曝光控制

typedef struct \_exposure\_param

{

float      m\_step;        // 自动曝光控制（AEC）的调整步骤。 默认值为10。

    float   m\_exposure\_time;// 持续曝光时间以微秒为单位。 范围是0.1~20。 默认值为7.25。

unsigned int  m\_expected\_brightness;   // AEC的恒定预期亮度。 范围是50~200。 默认值为85。

unsigned int  m\_is\_auto\_exposure;     // 1：自动曝光; 0：持续曝光

    int           m\_camera\_pair\_index; // index of Guidance Sensor

    \_exposure\_param(){

        m\_step = 10;

        m\_exposure\_time = 7.68;

        m\_expected\_brightness = 85;

        m\_is\_auto\_exposure = 1;

        m\_camera\_pair\_index = 1;

    }

}exposure\_param;

XII、motion

描述: 定义全局位置数据结构。位置单位为m，速度单位为m/s.

typedef struct \_motion

{

    unsigned int     frame\_index;

    unsigned int     time\_stamp;

    int                corresponding\_imu\_index;

    float            q0;

    float            q1;

    float            q2;

    float            q3;

    int              attitude\_status;  // 0:invalid; 1:valid

    float            position\_in\_global\_x;  // position in global frame: x

    float            position\_in\_global\_y;  // position in global frame: y

    float            position\_in\_global\_z;  // position in global frame: z

    int              position\_status; // lower 3 bits are confidence. 0:invalid; 1:valid

    float            velocity\_in\_global\_x;  // velocity in global frame: x

    float            velocity\_in\_global\_y;  // velocity in global frame: y

    float            velocity\_in\_global\_z;  // velocity in global frame: z

    int              velocity\_status; // lower 3 bits are confidence. 0:invalid; 1:valid

    float            reserve\_float[8];

    int              reserve\_int[4];

    float            uncertainty\_location[3];// uncertainty of position

    float            uncertainty\_velocity[3];// uncertainty of velocity

} motion;