



中华人民共和国国家标准

GB/T 44220—2024

虚拟现实设备接口 定位设备

Interface of virtual reality equipment—Positioning equipment

2024-07-24 发布

2025-02-01 实施

国家市场监督管理总局
国家标准化管理委员会

发布

目 次

前言.....Ⅲ

1 范围.....1

2 规范性引用文件.....1

3 术语、定义和缩略语.....1

 3.1 术语和定义.....1

 3.2 缩略语.....1

4 定位设备和接口类型.....2

 4.1 定位设备类型2

 4.2 接口类型.....2

5 接口数据类型.....3

 5.1 接口数据分类.....3

 5.2 数据结构和参数的命名规则.....3

 5.3 配置数据.....3

 5.4 状态数据.....14

 5.5 计算数据.....18

 5.6 多定位设备间的数据交换模式.....25

 5.7 数据刷新率的数据结构.....26

6 接口函数.....26

 6.1 概述.....26

 6.2 参数配置.....26

 6.3 工作状态控制.....31

 6.4 定位设备状态信息获取.....31

参考文献.....33

前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第 1 部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利，本文件的发布机构不承担识别专利的责任。

本文件由全国音频、视频及多媒体系统与设备标准化技术委员会(SAC/TC 242)提出并归口。

本文件起草单位：中国电子技术标准化研究院、海信视像科技有限公司、歌尔股份有限公司、深圳创维智能系统技术研究院、北京凌宇测控科技有限公司、青岛虚拟现实研究院有限公司、工业和信息化部电子第五研究所、江南大学、西安理工大学、南昌虚拟现实检测技术有限公司、深圳市三诺数字科技有限公司、广东虚拟现实科技有限公司。

本文件主要起草人：郝冬宁、王平平、李婧欣、张子浩、郁昌贺、张向军、严小天、张佳宁、赵晓莺、赵洪良、赵浩之、韦胜钰、陈嘉琦、张曼华、王映辉、宁小娟、胡翔、胡开拓、董桂官、许惠斌、贺杰。

虚拟现实设备接口 定位设备

1 范围

本文件规定了虚拟现实设备对定位设备进行参数获取、参数查询、参数配置、工作状态控制和定位设备状态信息传输的接口要求。

本文件适用于虚拟现实设备的设计与生产,以及定位服务接口设计。增强现实设备参考使用。

2 规范性引用文件

本文件没有规范性引用文件。

3 术语、定义和缩略语

3.1 术语和定义

下列术语和定义适用于本文件。

3.1.1

虚拟现实 virtual reality

采用以计算机为核心的现代高科技手段生成的逼真的视觉、听觉、触觉、嗅觉、味觉等多感官一体化的数字化人工环境,用户借助一些输入、输出设备,采用自然的方式与虚拟世界的对象进行交互,相互影响,从而产生亲临真实环境的感觉和体验。

[来源:GB/T 38259—2019,3.1]

3.1.2

定位设备 positioning equipment

跟踪虚拟现实设备使用者的动态位置和方向信息,并将位姿、点云、平面、时间和触觉反馈等信息发送到虚拟现实控制系统中的设备。

注1:设备包括但不限于惯性传感器(如加速度计、陀螺仪)、磁力计、光传感器、声传感器、图像采集设备。

注2:输出信息包括位姿、点云、平面、时间、触觉反馈等。

注3:虚拟现实定位设备按功能分为头部定位、手柄定位、手势识别、肢体定位追踪设备等。

3.1.3

位姿 pose

确定物体在三维空间的三个自由度的姿态和空间位置。

3.1.4

四元数 quaternion

确定物体在三维空间的三个自由度的方向姿态的表示方法。

注:利用该特性能够表达任意三维旋转,一个四元数拥有一个实部和三个虚部。

3.2 缩略语

下列缩略语适用于本文件。

ACC:加速度计(Accelerometer)

GPIO:通用输入/输出(General-Purpose Input/Output)
GYRO:陀螺仪(Gyroscope)
HMD:头戴式显示设备(Head Mounted Display)
ID:识别号(Identification)
IMU:惯性测量单元/惯性传感器(Inertial Measurement Unit)
LED:发光二极管(Light-Emitting Diode)
MAG:磁力计(Magnetometer)
PC:个人计算机(Personal Computer)
SLAM:即时定位与建图(Simultaneous Localization And Mapping)
SPI:串行外设接口(Serial Peripheral Interface)
USB:通用串行总线(Universal Serial Bus)
3DOF:三自由度(3 Degrees of Freedom)
6DOF:六自由度(6 Degrees of Freedom)

4 定位设备和接口类型

4.1 定位设备类型

虚拟现实设备中定位设备的分类,如表 1 所示。

表 1 定位设备类型

设备类型接口字段	取值	描述
VRDEVICE_TYPE	0x00	头戴式设备
VRDEVICE_TYPE	0x01	控制器设备
VRDEVICE_TYPE	0x02	基站
.....	0x03~0xFF	保留

4.2 接口类型

接口类型是指定位设备之间,定位设备与服务器(云端主机、本地 PC、本地游戏主机等)进行信息交互的接口,如表 2 所示。

表 2 接口类型

接口类型接口字段	取值	描述
VRPORT_TYPE	0x00	无线网络接口
VRPORT_TYPE	0x01	蓝牙接口
VRPORT_TYPE	0x02	USB 传输接口
VRPORT_TYPE	0x03	移动行业处理器接口
VRPORT_TYPE_SPI	0x04	串行外设接口
VRPORT_TYPE_GPIO	0x05	通用输入/输出接口
.....	0x06~0xFF	保留

5 接口数据类型

5.1 接口数据分类

虚拟现实设备定位系统接口数据类型可分为配置数据、状态数据和计算数据。

- a) 配置数据包括虚拟现实设备定位系统各部分的运行模式、计算方法以及各种配置参数,是虚拟现实设备定位系统的输入数据,支持查询。
- b) 状态数据包括虚拟现实设备定位系统各部分的工作状态,是虚拟现实定位系统的输出数据,支持查询。
- c) 计算数据包括虚拟现实设备定位系统产生的各种计算数据,包括参与定位计算的原始数据和计算结果数据。其中,原始数据是虚拟现实设备定位系统的输入数据,支持查询;计算结果数据是虚拟现实设备定位系统的输出数据,支持查询。

5.2 数据结构和参数的命名规则

5.2.1 数据结构以及数据类型的英文命名以“vr”开头,连以首字母大写的释义性英文单词,如相机配置的数据结构英文命名为“vrCameraConfig”。

5.2.2 数据结构中参数类型使用约定俗称的类型命名以及本章定义的各种数据类型,如“Vector<vrCameraConfig>”意为相机配置数组,又如 Int32,意为 32 位的整型数据。

5.2.3 数据类型中枚举项的英文命名以“VR”开头,连以全单词大写的释义性英文单词,各单词中间使用“_”相连,例如“VR_CAMERA_TYPE_GENERAL_RGB”表示相机分类中的普通 RGB 相机。

5.3 配置数据

5.3.1 概述

配置数据结构如表 3 所示。

表 3 配置数据结构

配置数据结构名称	描述
vrSystemConfig	系统配置,见表 5
vrBaseConfig	基站配置,见表 14
vrCameraConfig	相机配置,见表 16
vrSensorConfig	传感器配置,见表 23

各配置数据类型对应的配置模式参数如表 4 所示,可供用户查询设置。

表 4 配置模式参数

配置模式参数	描述
vrSystemCostMode	系统功耗模式,见表 7
vrSystemDormancyMode	系统休眠模式,见表 8
vrSystemShutDownMode	系统关机模式,见表 9
vrHMDPositioningMode	HMD 定位模式,见表 10
vrHMDRelocalizationMode	HMD 重定位模式,见表 11
vrControllerPositioningMode	控制器定位模式,见表 12
vrControllerInteractiveMode	控制器交互模式,见表 13

5.3.2 系统配置

系统配置数据结构如表 5 所示。

表 5 系统配置数据结构

数据类型	参数类型	描述
vrSystemConfig	vrSystemInfo	系统基本信息,见表 6
	vrSystemCostMode	系统功耗模式,见表 7
	vrSystemDormancyMode	系统休眠模式,见表 8
	vrSystemShutDownMode	系统关机模式,见表 9
	vrHMDPositioningMode	HMD 定位模式,见表 10
	vrHMDRelocalizationMode	HMD 重定位模式,见表 11
	vrControllerPositioningMode	控制器定位模式,见表 12
	vrControllerInteractiveMode	控制器交互模式,见 5 表 13
	vrHMDCloudSyncStatus	HMD 与云服务器的同步状态,见表 34
	int32_t dormancy_time	系统休眠计时参数
	vrBaseConfig	基站配置,见表 14

5.3.3 系统基本信息

系统基本信息数据结构如表 6 所示。

表 6 系统基本信息数据结构

数据类型	参数类型	描述
vrSystemInfo	string device_name	设备名称
	string device_type	设备类型
	string device_software_version	软件版本
	string device_hardware_version	硬件版本
	string device_firmware_version	固件版本
	string device_serial_number	设备序列号

5.3.4 系统功耗模式

系统功耗模式如表 7 所示。

表 7 系统功耗模式

数据类型	枚举值	描述
vrSystemCostMode	VR_SYSTEM_COST_MODE_LOW=0x00	低功耗模式
	VR_SYSTEM_COST_MODE_NORMAL=0x01	正常功耗
	VR_SYSTEM_COST_MODE_HIGH_PERFORMANCE=0x02	高性能功耗
	0x03~0xFF	保留

5.3.5 系统休眠模式

系统休眠模式如表 8 所示。

表 8 系统休眠模式

数据类型	枚举值	描述
vrSystemDormancyMode	VR_SYSTEM_DORMANCY_MODE_ALWAYS_ON=0x00	不休眠
	VR_SYSTEM_DORMANCY_MODE_AUTO =0x01	自动休眠
	VR_SYSTEM_DORMANCY_MODE_AUTO_NO_MOTION=0x02	无动作时 自动休眠
	VR_SYSTEM_DORMANCY_MODE_AUTO_NO_KEY_INPUT=0x03	无输入时 自动休眠
	0x04~0xFF	保留

5.3.6 系统关机模式

系统关机模式如表 9 所示。

表 9 系统关机模式

数据类型	枚举值	描述
vrSystemShutDownMode	VR_SYSTEM_SHUTDOWN _MODE_ALWAYS_ON=0x00	不关机
	VR_SYSTEM_SHUTDOWN_MODE_AUTO =0x01	自动关机
	VR_SYSTEM_SHUTDOWN_MODE_AUTO_NO_MOTION=0x02	无动作时自 动关机
	VR_SYSTEM_SHUTDOWN_MODE_AUTO_NO_KEY_INPUT= 0x03	无输入时自 动关机
	0x04~0xFF	保留

5.3.7 HMD 定位模式

HMD 定位模式如表 10 所示。

表 10 HMD 定位模式

数据类型	枚举值	描述
vrSystemShutDownMode	VR_HMD_POSITIONING_MODE_INSIDE_OUT_LOCALIZATION=0x00	由内向外定位
	VR_HMD_POSITIONING_MODE_INSIDE_OUT_MAPPING=0x01	由内向外建图
	VR_HMD_POSITIONING_MODE_INSIDE_OUT_SLAM=0x02	由内向外 SLAM
	VR_HMD_POSITIONING_MODE_OUTSIDE_IN_DISCO=0x03	由外向内声光定位
	0x04~0xFF	保留

5.3.8 HMD 重定位模式

HMD 重定位模式如表 11 所示。

表 11 HMD 重定位模式

数据类型	枚举值	描述
vrHMDRelocalizationMode	VR_HMD_RELOCALIZATION_MODE_LOCAL=0x00	局部重定位
	VR_HMD_RELOCALIZATION_MODE_GLOBAL=0x01	全局重定位
	0x02~0xFF	保留

5.3.9 控制器定位模式

控制器定位模式如表 12 所示。

表 12 控制器定位模式

数据类型	枚举值	描述
vrControllerPositioningMode	VR_CONTROLLER_POSITIONING_MODE_3DOF_IMU_ONLY=0x00	IMU 姿态模式
	VR_CONTROLLER_POSITIONING_MODE_6DOF_ELECTROMAGNETIC=0x01	电磁位姿模式
	VR_CONTROLLER_POSITIONING_MODE_6DOF_ULTRASONIC=0x02	超声位姿模式
	VR_CONTROLLER_POSITIONING_MODE_6DOF_COMPUTER_VISION=0x03	计算机视觉位姿模式
	VR_CONTROLLER_POSITIONING_MODE_6DOF_OUTSIDE_IN_DISCO=0x04	由外向内声光位姿模式
	0x05~0xFF	保留

5.3.10 控制器交互模式

控制器交互模式如表 13 所示。

表 13 控制器交互模式

数据类型	枚举值	描述
vrControllerInteractiveMode	VR_CONTROLLER_INTERACTIVE_MODE_NONE=0x00	无手柄交互
	VR_CONTROLLER_INTERACTIVE_MODE_ONE=0x01	单手柄交互
	VR_CONTROLLER_INTERACTIVE_MODE_MULTIPLE=0x02	多手柄交互
	0x03~0xFF	保留

5.3.11 基站配置

基站配置数据结构如表 14 所示。

表 14 基站配置数据结构

数据类型	参数类型	描述
vrBaseConfig	int32_t base_id	基站 ID
	vrBaseType	基站类型,见表 15
	float vrBaseHeight	基站高度参数

5.3.12 基站类型

基站类型如表 15 所示。

表 15 基站类型

数据类型	枚举值	描述
vrBaseType	VR_BASE_TYPE_DISCO=0x00	声光定位
	VR_BASE_TYPE_ULTRASONIC=0x01	超声
	VR_BASE_TYPE_LASE=0x02	镭射
	0x03~0xFF	保留

5.3.13 相机配置

相机配置数据结构如表 16 所示。

表 16 相机配置数据结构

数据类型	参数类型	描述
vrCameraConfig	int32_t camera_id	相机 ID 号
	vrCamera Type	相机类型,见表 17
	float focal_length_x	x 方向焦距
	float focal_length_y	y 方向焦距
	float center_length_x	x 方向光轴-中心偏移
	float center_length_y	y 方向光轴-中心偏移
	float radial_distortion_k1	径向畸变参数
	float radial_distortion_k2	
	float radial_distortion_k3	
	float tangential_distortion_p1	切向畸变参数
	float tangential_distortion_p2	
	vrCameraExposureType	相机曝光类型
	vrCameraFocusType	相机焦距类型
	vrCameraShutterType	相机快门类型
	vrImageFormatType	图像数据格式类型
	vrImageResolutionType	图像分辨率类型
	vrRange with RangeType-FPS	相机帧率范围
	int32_t current_FPS	相机当前帧率

5.3.14 相机类型

相机类型如表 17 所示。

表 17 相机类型

数据类型	枚举值	描述
vrCameraType	VR_CAMERA_TYPE_GENERAL_RGB=0x00	普通 RGB 相机
	VR_CAMERA_TYPE_FISHEYE=0x01	鱼眼相机
	VR_CAMERA_DATA_TYPE_DEPTH=0x02	深度相机
	0x03~0xFF	保留

5.3.15 图像数据格式

图像数据格式类型如表 18 所示。

表 18 图像数据格式类型

数据类型	枚举值	描述
vrImageFormatType	VR_IMAGE_FORMAT_TYPE_RGBA_8888=0x00	RGBA_8888 格式
	VR_IMAGE_FORMAT_TYPE_RGBX_8888=0x01	RGBX_8888 格式
	VR_IMAGE_FORMAT_TYPE_RGB_888=0x02	RGB_888 格式
	VR_IMAGE_FORMAT_TYPE_RGB_565=0x03	RGB_565 格式
	VR_IMAGE_FORMAT_TYPE_RGBA_FP16=0x04	RGBA_FP16 格式
	VR_IMAGE_FORMAT_TYPE_YUV_420_888=0x05	YUV_420_888 格式
	VR_IMAGE_FORMAT_TYPE_JPEG=0x06	JPEG 格式
	VR_IMAGE_FORMAT_TYPE_DEPTH16=0x07	16 位的深度图格式
	VR_IMAGE_FORMAT_TYPE_POINT_CLOUD=0x08	点云格式
	0x09~0xFF	保留

5.3.16 图像分辨率

图像分辨率如表 19 所示。

表 19 图像分辨率

数据类型	枚举值	描述
vrImageResolutionType	VR_IMAGE_RESOLUTION_TYPE_320_240=0x00	320×240
	VR_IMAGE_RESOLUTION_TYPE_400_400=0x01	400×400
	VR_IMAGE_RESOLUTION_TYPE_480_360=0x02	480×360
	VR_IMAGE_RESOLUTION_TYPE_640_360=0x03	640×360
	VR_IMAGE_RESOLUTION_TYPE_640_480=0x04	640×480
	VR_IMAGE_RESOLUTION_TYPE_720_540=0x05	720×540
	VR_IMAGE_RESOLUTION_TYPE_720_720=0x06	720×720
	VR_IMAGE_RESOLUTION_TYPE_960_540=0x07	960×540
	VR_IMAGE_RESOLUTION_TYPE_960_720=0x08	960×720
	VR_IMAGE_RESOLUTION_TYPE_1280_720=0x09	1 280×720
	VR_IMAGE_RESOLUTION_TYPE_1280_960=0x0A	1 280×960
	VR_IMAGE_RESOLUTION_TYPE_1440_720=0x0B	1 440×720
	VR_IMAGE_RESOLUTION_TYPE_1440_1080=0x0C	1 440×1 080
	VR_IMAGE_RESOLUTION_TYPE_1920_960=0x0D	1 920×960
	VR_IMAGE_RESOLUTION_TYPE_1920_1080=0x0E	1 920×1 080
	VR_IMAGE_RESOLUTION_TYPE_2160_1080=0x0F	2 160×1 080
	VR_IMAGE_RESOLUTION_TYPE_3840_2160=0x10	3 840×2 160
	0x11~0xFF	保留

5.3.17 相机曝光模式

相机曝光模式如表 20 所示。

表 20 相机曝光模式

数据类型	枚举值	描述
vrCameraExposureType	VR_CAMERA_EXPOSURE_TYPE_AUTO=0x00	自动曝光
	VR_CAMERA_EXPOSURE_TYPE_AUTO_OFF=0x01	非自动曝光
	0x02~0xFF	保留

5.3.18 相机聚焦模式

相机聚焦模式如表 21 所示。

表 21 相机聚焦模式

数据类型	枚举值	描述
vrCameraFocusType	VR_CAMERA_FOCUS_MODE_AUTO=0x00	自动聚焦
	VR_CAMERA_FOCUS_MODE_AUTO_OFF=0x01	非自动聚焦
	VR_CAMERA_FOCUS_MODE_CONTINUE_VIDEO=0x02	连续录像式聚焦
	VR_CAMERA_FOCUS_MODE_CONTINUE_PICTURE=0x03	连续照片式聚焦
	0x04~0xFF	保留

5.3.19 相机快门模式

相机快门模式如表 22 所示。

表 22 相机聚焦模式

数据类型	枚举值	描述
vrCameraShutterType	VR_CAMERA_SHUTTER_TYPE_GLOBAL=0x00	全局快门
	VR_CAMERA_SHUTTER_TYPE_ROLLING=0x01	卷帘快门
	0x04~0xFF	保留

5.3.20 传感器配置

每个传感器拥有一个单独的与其对应的传感器配置数据实例,如表 23 所示。

表 23 传感器配置数据结构

数据类型	参数类型	描述
vrSensorConfig	int32_t sensor_id	传感器 ID
	int32_t belong_device_id	所属设备 ID
	vrSensorType	传感器类型,见表 24
	int32_t current_update_frequency	传感器数据更新率
	Vector<vrRange>	量程数据数组,见表 25
	Vector<vrCorrection>	标定数据数组,见表 27

5.3.21 传感器类型

传感器类型如表 24 所示。

表 24 传感器类型

数据类型	枚举值	描述
vrSensorType	VR_SENSOR_TYPE_IMU_ACC=0x00	加速度计
	VR_SENSOR_TYPE_IMU_GYRO=0x01	陀螺仪
	VR_SENSOR_TYPE_MAGNETIC=0x02	磁力计
	VR_SENSOR_TYPE_GPS=0x03	GPS
	VR_SENSOR_TYPE_AIR_PRESSURE=0x04	气压计
	VR_SENSOR_TYPE_BATTERY=0x05	电池电量传感器
	VR_SENSOR_TYPE_ACOUSTIC=0x06	声学传感器
	0x07~0xFF	保留

5.3.22 量程配置

量程配置数据结构如表 25 所示。

表 25 量程配置数据结构

数据类型	参数类型	描述
vrRange	vrRangeType	量程类型,见表 26
	int32_t belong_sensor_id	所属传感器 ID
	double min	最小值
	double max	最大值

5.3.23 量程类型

量程类型如表 26 所示。

表 26 量程类型

数据类型	枚举值	描述
vrRangeType	VR_RANGE_TYPE_IMU_ACC_X=0x00	加速度计 X 轴量程
	VR_RANGE_TYPE_IMU_ACC_Y=0x01	加速度计 Y 轴量程
	VR_RANGE_TYPE_IMU_ACC_Z=0x02	加速度计 Z 轴量程
	VR_RANGE_TYPE_IMU_GYRO_X=0x03	陀螺仪 X 轴量程
	VR_RANGE_TYPE_IMU_GYRO_Y=0x04	陀螺仪 Y 轴量程
	VR_RANGE_TYPE_IMU_GYRO_Z=0x05	陀螺仪 Z 轴量程
	VR_RANGE_TYPE_MAG_X=0x06	磁力计 X 轴量程
	VR_RANGE_TYPE_MAG_Y=0x07	磁力计 Y 轴量程
	VR_RANGE_TYPE_MAG_Z=0x08	磁力计 Z 轴量程
	VR_RANGE_TYPE_FPS=0x09	FPS 范围
	VR_RANGE_TYPE_UPDATE_FREQUENCY=0x0A	数据更新率范围
	0x0B~0xFF	保留

5.3.24 标定配置

标定配置如表 27 所示。

表 27 标定配置数据结构

数据类型	参数类型	描述
vrCorrection	vrCorrectionType	标定参数类型,见表 28
	int32_t belong_sensor_id	所属传感器 ID
	mat	矩阵

5.3.25 标定参数类型

标定参数类型如表 28 所示。

表 28 标定参数类型

数据类型	枚举值	描述
vrCorrectionType	VR_CORRECTION_TYPE_IMU_ACC_BIAS=0x00	IMU 加速度计的零偏值
	VR_CORRECTION_TYPE_IMU_ACC_SCALE=0x01	IMU 加速度计的尺度因数
	VR_CORRECTION_TYPE_IMU_ACC_MISALIGNMENT=0x02	IMU 加速度计的非正交性矩阵
	VR_CORRECTION_TYPE_IMU_GYRO_BIAS=0x03	IMU 陀螺仪的零偏值
	VR_CORRECTION_TYPE_IMU_GYRO_SCALE=0x04	IMU 陀螺仪的尺度因数

表 28 标定参数类型（续）

数据类型	枚举值	描述
vrCorrectionType	VR_CORRECTION_TYPE_IMU_GYRO_MISALIGNMENT=0x05	IMU陀螺仪的非正交性矩阵
	VR_CORRECTION_TYPE_MAG_BIAS=0x06	磁力计的零偏值
	VR_CORRECTION_TYPE_MAG_SCALE=0x07	磁力计的尺度因数
	VR_CORRECTION_TYPE_MAG_MISALIGNMENT=0x08	磁力计的非正交性矩阵
	0x09~0xFF	保留

5.3.26 LED 类型

LED 类型如表 29 所示。

表 29 LED 类型

数据类型	枚举值	描述
vrLEDType	VR_LED_TYPE_SWITCH=0x00	开关显示用LED
	VR_LED_TYPE_STATUS=0x01	状态显示用LED
	VR_LED_TYPE_LOCALIZATION=0x02	定位用LED
	VR_LED_TYPE_DISPLAY=0x03	特效显示用LED
	0x04~0xFF	保留

5.3.27 声学单元类型

声学单元类型如表 30 所示。

表 30 声学单元类型

数据类型	枚举值	描述
vrSonicUnitType	VR_SONIC_UNIT_TYPE_LOCALIZATION=0x00	定位用声学单元
	VR_SONIC_UNIT_TYPE_DISPLAY=0x01	表示用声学单元
	0x02~0xFF	保留

5.3.28 控制器类型

控制器类型如表 31 所示。

表 31 控制器类型

数据类型	枚举值	描述
vrControllerType	VR_CONTROLLER_TYPE_LEFT_HAND_HOLD=0x00	左手手持式控制器
	VR_CONTROLLER_TYPE_RIGHT_HAND_HOLD=0x01	右手手持式控制器
	VR_CONTROLLER_TYPE_TIED=0x02	捆绑式控制器
	0x03~0xFF	保留

5.4 状态数据

5.4.1 概述

定位设备工作状态分为 HMD 定位系统工作状态、HMD 与云服务器的同步状态、控制器定位系统工作状态等 7 种类型。具体状态见表 32 所示。

表 32 工作状态类型

状态数据类型名称	描述
vrHMDPositioningSystemWorkingStatus	HMD 定位系统工作状态,见表 33
vrHMDCloudSyncStatus	HMD 与云服务器的同步状态,见表 34
vrControllerPositioningSystemWorkingStatus	控制器定位系统工作状态,见表 35
vrSensorStatus	传感器工作状态,见表 36
vrLEDStatus	LED 工作状态,见表 37
vrSonicUnitStatus	声学单元工作状态,见表 38
vrBaseStatus	基站工作状态,见表 39
vrDeviceErrorStatus	设备故障状态,见表 40
vrDeviceResetStatus	设备复位状态,见表 41

5.4.2 HMD 定位系统工作状态

HMD 定位系统工作状态如表 33 所示。

表 33 HMD 定位系统工作状态

数据类型	枚举值	描述
vrHMDPositioningSystemWorkingStatus	VR_HMD_POSITIONING_SYSTEM_WORKING_STATU_START=0x00	开机状态
	VR_HMD_POSITIONING_SYSTEM_WORKING_STATU_STANDBY=0x01	待机状态
	VR_HMD_POSITIONING_SYSTEM_WORKING_STATU_INITIALIZATION=0x02	初始化状态

表 33 HMD 定位系统工作状态（续）

数据类型	枚举值	描述
vrHMDPositioningSystemWorking Status	VR_HMD_POSITIONING_SYSTEM_WORKING_STATU_TRACKING=0x03	定位跟踪状态
	VR_HMD_POSITIONING_SYSTEM_WORKING_STATU_TRACKING_PAUSED=0x04	暂停定位跟踪状态
	VR_HMD_POSITIONING_SYSTEM_WORKING_STATU_TRACKING_RESUME=0x05	重新定位跟踪状态
	VR_HMD_POSITIONING_SYSTEM_WORKING_STATU_RELOCALIZATION=0x06	重定位跟踪装态
	VR_HMD_POSITIONING_SYSTEM_WORKING_STATU_TRACKING_STOPPED=0x07	停止跟踪状态
	VR_HMD_POSITIONING_SYSTEM_WORKING_STATU_TRACKING_FALED=0x08	跟踪失败状态
	VR_HMD_POSITIONING_SYSTEM_WORKING_STATU_DORMANCY=0x09	休眠状态
	VR_HMD_POSITIONING_SYSTEM_WORKING_STATU_REBOOT=0x0A	重启状态
	VR_HMD_POSITIONING_SYSTEM_WORKING_STATU_SHUT_DOWN=0x0B	关机状态
	0x0C~0xFF	保留

5.4.3 HMD 与云服务器的同步状态

HMD 与云服务器的同步状态如表 34 所示。

表 34 HMD 与云服务器的同步状态

数据类型	枚举值	描述
vrHMDCloudSync Status	VR_HMD_CLOUD_SYNC_STATUS_SYNCED=0x01	HMD 与云服务器已同步
	VR_HMD_CLOUD_SYNC_STATUS_ASYNC=0x01	HMD 与云服务器未同步
	0x02~0xFF	保留

5.4.4 控制器定位系统工作状态

控制器定位系统工作状态如表 35 所示。

表 35 控制器定位系统工作状态

数据类型	枚举值	描述
vrControllerPositioning SystemWorkingStatus	VR_CONTROLLER_POSITIONING_SYSTEM_WORKING_STATUS_NO_LINK=0x00	控制器无连接
	VR_CONTROLLER_POSITIONING_SYSTEM_WORKING_STATUS_LINKING=0x01	控制器正在连接
	VR_CONTROLLER_POSITIONING_SYSTEM_WORKING_STATUS_DEVICE_CORRECTING=0x02	控制器已连接,设备正在标定中
	VR_CONTROLLER_POSITIONING_SYSTEM_WORKING_SYNCED_TRACKING =0x03	控制器已连接,坐标系与HMD已同步,并且持续跟踪
	VR_CONTROLLER_POSITIONING_SYSTEM_WORKING_STATUS_SYNCED_LOSE_TRACKING=0x04	控制器已连接,坐标系与HMD已同步,跟踪丢失
	VR_CONTROLLER_POSITIONING_SYSTEM_WORKING_STATUS_ASYNC=0x05	控制器已连接,坐标系未与HMD同步
	0x06~0xFF	保留

5.4.5 传感器工作状态

传感器工作状态如表 36 所示。

表 36 传感器工作状态

数据类型	枚举值	描述
vrSensorStatus	VR_SENSOR_STATUS_ON=0x00	上电
	VR_SENSOR_STATUS_OFF=0x01	掉电
	VR_SENSOR_STATUS_ERROR=0x02	故障
	0x03~0xFF	保留

5.4.6 LED 工作状态

LED 工作状态如表 37 所示。

表 37 LED 工作状态

数据类型	枚举值	描述
vrLEDStatus	VR_LED_STATUS_ON_STADY=0x00	常亮
	VR_LED_STATUS_ON_FLASH=0x01	闪烁
	VR_LED_STATUS_OFF=0x02	不亮
	0x03~0xFF	保留

5.4.7 声学单元工作状态

声学单元工作状态如表 38 所示。

表 38 声学单元工作状态

数据类型	枚举值	描述
vrSonicUnitStatus	VR_SONIC_UNIT_STATUS_ON=0x00	开
	VR_SONIC_UNIT_STATUS_OFF=0x01	关
	0x02~0xFF	保留

5.4.8 基站工作状态

基站工作状态如表 39 所示。

表 39 基站工作状态

数据类型	枚举值	描述
vrBaseStatus	VR_BASE_STATUS_USE_ATTITUDE_ON=0x00	姿态模式启用
	VR_BASE_STATUS_USE_ATTITUDE_OFF=0x01	姿态模式关闭
	0x02~0xFF	保留

5.4.9 设备故障状态

设备故障状态是指设备无法报告正确定位数据的原因,见表 40。

表 40 设备故障状态

数据类型	枚举值	描述
vrDeviceError Status	VR_DEVICE_ERROR_TYPE_HARDWARE=0x00	硬件错误,是指设备硬件异常导致无法正常工作,这类错误可能是由于某些模块没有正确打开、供电不足或者无法正确建立通信等硬件原因导致的
	VR_DEVICE_ERROR_TYPE_FIRMWARE=0x01	固件信息错误,是指定位设备的软件有错误,这类错误通常是因为固件版本不正确或者固件更新不完整等原因导致的
	VR_DEVICE_ERROR_TYPE_CONFIG=0x02	配置错误,是指定位设备工作正常,但用户对定位设备的配置有错或者不合理,导致设备无法正常工作
	VR_DEVICE_ERROR_TYPE_DISABLE=0x03	不在可用区域,是指定位设备整体工作正常,但被定位物体已经移出可用区域,或者被遮挡,导致无法更新定位数据
	0x04~0xFF	保留

5.4.10 设备复位状态

设备复位状态指设备遇到一些未知错误导致无法继续正常工作时进行的复位重启状态,见表 41。

表 41 设备复位状态

数据类型	枚举值	描述
vrDeviceResetStatus	VR_DEVICE_RESET_HMD_RESET=0x00	HMD 设备复位
	VR_DEVICE_RESET_CONTROLLER_RESET=0x01	控制器设备复位
	VR_DEVICE_SENSOR_RESET=0x02	传感器复位
	VR_DEVICE_LED_RESET=0x03	LED 复位
	VR_DEVICE_SONICUNIT_RESET=0x04	声学单元复位
	VR_DEVICE_BASE_RESET=0x05	基站复位
	0x06~0xFF	保留

5.5 计算数据

5.5.1 概述

计算数据类主要包含可定位设备、图像、位姿、运动状态、虚拟平面等数据类型,如表 42 所示。

表 42 计算数据结构

计算数据结构名称	描述
vrPositionableDevice	可定位设备,见表 43
vrFrame	数据帧,见表 45
vrImage	图像数据,见表 46
vr2DPoint	二维点坐标数据,见表 47
vr3DPoint	三维点坐标数据,见表 48
vrQuaternion	四元数数据,见表 49
vrPose	位姿数据,见表 50
vrMat	矩阵数据,见表 51
vrImageFeatureData	图像集合特征数据,见表 52
vrPlane	虚拟平面数据,见表 53
vrIMUData	六自由度的 IMU 数据,见表 54
vrMagdata	三自由度磁力计测量数据,见表 55
vrMotionData	六自由度的运动状态数据,见表 56
vrSensor	传感器数据,见表 57
vrLED	LED 设备数据,见表 58
vrLEDColor	LED 颜色数据,见表 59
vrSonicUnit	声学单元数据,见表 60
vrController	控制器设备数据,见表 61
vrBase	基站数据,见表 62

5.5.2 可定位设备

可定位设备数据如表 43 所示。

表 43 可定位设备数据结构

数据类型	参数类型	描述
vrPositionableDevice	int32_t device_id	设备 ID
	vrPositionableDeviceType	可定位设备类型,见表 44
	int64_t time_stamp	所含数据时间戳
	vrPose	所在时间戳时刻的位姿数据,见表 50
	vrMotionData	所在时间戳时刻的运动状态数据,见表 56

5.5.3 定位设备类型

定位设备类型如表 44 所示。

表 44 可定位设备类型

数据类型	枚举值	描述
vrPositionableDeviceType	VR_POSITIONABLE_DEVICE_TYPE_HMD=0x00	HMD
	VR_POSITIONABLE_DEVICE_TYPE_CONTROLLER=0x01	控制器
	VR_POSITIONABLE_DEVICE_TYPE_BASE=0x02	基站
	0x03~0xFF	保留

5.5.4 数据帧结构

数据帧至少包含同一时刻内虚拟现实设备定位系统搜集到的图像,以及经过对其进行计算得出的位姿、运动状态数据、图像集合特征数据,其数据结构如表 45 所示。

表 45 数据帧数据结构

数据类型	参数类型	描述
vrFrame	int32_t frame_id	数据帧 ID
	int64_t timestamp	时间戳
	Vector<vrImage>	图像数据,见表 46
	Vector<vrPose>	可定位设备的位姿数据,见表 50
	Vector<float quality>	可定位设备的位姿估计质量评价结果数组
	Vector<vrMotionData>	可定位设备的运动状态数据数组,见表 56
	Vector<vrImageFeatureData>	与图像数组对应的图像集合特征数据,见表 52
	Vector<vrPlane>	定位设备探测到的虚拟平面数据数组,见表 53

5.5.5 图像数据

图像数据为某一时刻下,由相机拍摄得到的图像数据,如表 46 所示。

表 46 图像数据结构

数据类型	参数类型	描述
vrImage	int32 image_id	图像 ID
	int64_t time_stamp	图像时间戳
	int32_t belong_frame_id	图像所属帧 ID
	int32_t belong_camera_id	图像所属相机 ID
	vrImageFormatType	图像数据格式,见表 18
	vrImageResolutionType	图像分辨率,见表 19
	vrMat	图像矩阵数据,见表 51

5.5.6 二维点坐标

二维点坐标的数据结构见表 47。

表 47 二维点坐标数据结构

数据结构名称	参数名称	数据类型	描述
vr2DPoint	x	double	X轴坐标
	y	double	Y轴坐标

5.5.7 三维点坐标

三维点坐标数据的数据结构如表 48 所示。

表 48 三维点坐标数据结构

数据结构名称	参数名称	数据类型	描述
vr3DPoint	x	double	X轴坐标
	y	double	Y轴坐标
	z	double	Z轴坐标

5.5.8 四元数数据

四元数数据的数据结构如表 49 所示。

表 49 四元数数据结构

数据结构名称	参数名称	数据类型	描述
vrQuaternion	w	double	$\cos(\theta/2)$
	x	double	$-\sin(\theta/2) \times vx$
	y	double	$-\sin(\theta/2) \times vy$
	z	double	$-\sin(\theta/2) \times vz$

5.5.9 位姿数据

位姿数据的数据结构如表 50 所示。

表 50 位姿数据结构

数据类型	参数类型	描述
vrPose	int64_t time_stamp	位姿数据时间戳
	vrQuaternion	姿态数据, 旋转四元数
	vr3DPoint	位置数据, 三维位置点

5.5.10 矩阵数据

矩阵数据的数据结构如表 51 所示。

表 51 矩阵数据结构

数据类型	参数类型	描述
vrMat	int32_t COLS	列数
	int32_t ROWS	行数
	int32_t Channel	通道数
	Int32_t bits	位数
	Vector<Int8_t>	二进制数组用于存储数据

5.5.11 图像集合特征数据

图像集合特征数据的数据结构如表 52 所示。

表 52 图像集合特征数据结构

数据类型	参数类型	描述
vrImageFeatureData	int32_t point_cloud_id	点云 ID
	int32_t belong_frame_id	所属图像 ID
	int32_t belong_camera_id	所属相机 ID
	int64_t time_stamp	数据时间戳
	vector<vr2DPoint>	参与定位跟踪的图像二维特征点数组

表 52 图像集合特征数据结构（续）

数据类型	参数类型	描述
vrImageFeatureData	vector <int32_t point_id>	参与定位跟踪的图像二维特征点 ID 数组
	vector<vr3DPoint>	参与定位跟踪的图像二维特征点映射在三维空间的空间点数组
	vector<int32_t 3DPoint_id>	参与定位跟踪的图像二维特征点映射在三维空间的空间点 ID 数组

5.5.12 虚拟平面数据

虚拟平面数据定义了虚拟现实设备定位系统探测到的平面,其数据结构如表 53 所示。

表 53 虚拟平面数据结构

数据类型	参数类型	描述
vrPlane	int32_t plane_id	平面 ID
	int32_t belong_frame_id	所属数据帧 ID
	int64_t time_stamp	数据时间戳
	vrPosition plane_normal	平面法向量
	vrPose	平面中心点位姿
	vector<vr3DPoint>(for vertex)	顶点数组
	float extent_xyz	包围盒扩展边界大小

5.5.13 IMU 数据结构

某一时刻下,IMU 测量得到的数据为加速度计和陀螺仪测量的组合,其数据结构如表 54 所示。

表 54 IMU 数据结构

数据类型	参数类型	描述
vrIMUdata	int64_t time_stamp	IMU 数据时间戳
	double acc_x	三轴加速度计数据
	double acc_y	
	double acc_z	
	double gyro_x	三陀螺仪数据
	double gyro_y	
	double gyro_z	

5.5.14 磁力计测量数据结构

某一时刻下,磁力计测量得到的数据,是指定位物体在参考坐标系下某一确定位置周围的三自由

度磁场强度,其数据结构如表 55 所示。

表 55 磁力计测量数据结构

数据类型	参数类型	描述
vrMagdata	int64_t time_stamp	IMU 数据时间戳
	double mag_x	三轴磁力计测量得到的数据
	double mag_y	
	double mag_z	

5.5.15 运动状态数据

运动状态数据是指目标在空间左手坐标系下三自由度的运动速度、角速度、加速度,其数据结构如表 56 所示。

表 56 运动状态数据结构

数据类型	参数类型	描述
vrMotionData	double velocity_x	在 X 轴的速度(单位:m/s)
	double velocity_y	在 Y 轴的速度(单位:m/s)
	double velocity_z	在 Z 轴的速度(单位:m/s)
	double angular_velocity_x	在 X 轴的角速度(单位:rad/s)
	double angular_velocity_y	在 Y 轴的角速度(单位:rad/s)
	double angular_velocity_z	在 Z 轴的角速度(单位:rad/s)
	double accerleration_x	在 X 轴的加速度(单位:m/s ²)
	double accerleration_y	在 Y 轴的加速度(单位:m/s ²)
	double accerleration_z	在 Z 轴的加速度(单位:m/s ²)

5.5.16 传感器数据结构

传感器数据结构如表 57 所示。

表 57 传感器数据结构

数据类型	参数类型	描述
vrSensor	Int32_t	传感器 ID
	vrSensorConfig	传感器配置,见表 23
	Vector<Uint8_t>	传感器数据二进制数组
	int32_t data_length	传感器数据二进制数组长度

5.5.17 LED 设备数据结构

LED 设备的数据结构如表 58 所示。

表 58 LED 设备数据结构

数据类型	参数类型	描述
vrLED	int32_t led_id	LED ID
	enum vrLEDType	LED 类型, 见表 29
vrLED	int32_t belong_device_id	LED 所属设备 ID
	enum vrLEDStatus	LED 工作状态, 见表 37
	float led_brightness	LED 亮度
	vrLEDColor	LED 颜色
	float flash_period	LED 频闪周期

5.5.18 LED 颜色数据结构

LED 颜色的数据结构如表 59 所示。

表 59 LED 颜色数据结构

数据类型	参数类型	描述
vrLEDColor	int32_t red	红色值
	int32_t blue	蓝色值

5.5.19 声学单元数据结构

声学单元的数据结构如表 60 所示。

表 60 声学单元数据结构

数据类型	参数类型	描述
vrSonicUnit	int32_t id	声学单元 ID
	int32_t belong_device_id	声学单元所属设备 ID
	vrSonicUnitType	声学单元类型, 见表 30
	vrSonicUnitStatus	声学单元工作状态, 见表 38
	float volumn	音量参数

5.5.20 控制器设备数据结构

控制器设备的数据结构, 如表 61 所示。

表 61 控制器设备数据结构

数据类型	参数类型	描述
vrLED	int32_t controller_id	控制器 ID
	vrControllerType	控制器类型,见表 31
	int64_t timestamp	控制器设备时间戳
	vrPose	位姿数据,见表 50
	float pose_quality	可定位设备的位姿估计质量评价结果参数
	vrMotionData	运动状态数据,见表 56
	vrControllerPositionSystemWorkingStatus	控制器定位系统工作状态,见表 35
	float vibration	震动参数
	int32_t controller_id	控制器 ID

5.5.21 基站数据结构

基站数据结构见表 62。

表 62 基站数据结构

数据类型	参数类型	描述
vrBase	int32_t id	基站 ID
	vrBaseConfig	基站配置,见表 14
	vrBaseStatus	基站工作状态,见表 39

5.6 多定位设备间的数据交换模式

当定位系统中具有多个定位设备时,各设备间数据交换模式见表 63。

表 63 多定位设备间数据交换模式

数据类型	枚举值	描述
vrCommunicationMode	VR_CALLBACK_MODE=0x00	回调模式,是指设备有数据更新时,主动调用主机设备预留的接口来更新数据的方式
	VR_QUERY_MODE=0x01	询问模式,是指主机设备在需要数据更新时,调用设备接口查询最新定位数据的方式
	VR_BROADCAST_MODE=0x02	广播模式,是指当前设备向周围或者网络上其他设备发送广播信号,来告诉其他设备该主机一些特定信息,通常为待连接状态信息、位置信息、设备名称等
	VR_SUBSCRIPTION_MODE=0x03	订阅模式是指设备通过网络向用户提供网上直接订阅、直接信息浏览的电子商务模式

表 63 多定位设备间数据交换模式（续）

数据类型	枚举值	描述
vrCommunicationMode	VR_PUSH_MODE=0x04	推送模式是指信息源主动将信息推送给用户的一种模式
	VR_PULL_MODE=0x05	拉取模式是指用户主动拉取自己想要的信息的一种模式
	0x06~0xFF	保留

5.7 数据刷新率的数据结构

数据刷新率是指设备平均每秒更新定位数据的次数,有的定位设备可以根据不同需求,被配置工作在不同的刷新率下。数据刷新率的数据结构见表 64。

表 64 数据刷新率数据结构

数据类型	枚举值	描述
vrFrameRate	VR_ORIENTATION_FRAME_RATE=0x00	姿态数据的刷新率,是指设备姿态、角速度、角加速度数据的刷新频率
	VR_POSITION_FRAME_RATE=0x01	位置数据的刷新率,是指设备位置、速度和加速度数据的刷新频率
	VR_DISPLAY_FRAME_RATE=0x02	VR 屏幕的刷新率,是指头显设备屏幕显示图像的刷新频率
	VR_INTERACTION_FRAME_RATE=0x03	交互数据的刷新率,是指摇杆、按键、扳机等常用人机交互数据的刷新频率
	VR_CAMERA_FRAME_RATE=0x04	图像采集的刷新率,是指设备上所携带的摄像头等外设的图像采集频率

6 接口函数

6.1 概述

本章列出的接口函数为虚拟现实设备定位系统软件 API 接口,接口函数样式使用英文代码样式,给出函数名、输入输出值以及函数描述,中间使用的函数变量名使用约定俗称的普通变量类型以及第 5 章给出的数据结构体类型及数据枚举类型。接口函数的返回值为 bool 型变量,表示调用是否成功,成功返回值为 true,失败返回值为 false。其他数据的返回方式是将传入的指针数据进行变更,数据类型与传入的数据类型一致。

6.2 参数配置

6.2.1 系统配置参数

系统配置参数获取接口函数见表 65。

表 65 系统配置参数获取接口

接口描述	接口名称
获取配置实例	bool vrConfigGetConfig(vrConfig *config)
获取系统功耗模式	bool vrConfigGetSystemCostMode(const vrConfig *config, vrSystemCostMode *mode)
获取系统睡眠模式	bool vrConfigGetSystemDormancyMode(const vrConfig *config, vrSystemDormancyMode *mode)
获取系统关机模式	bool vrConfigGetSystemShutDownMode(const vrConfig *config, vrSystemShutDownMode *mode)
获取HMD定位工作模式	bool vrConfigGetHMDPositioningMode(const vrConfig *config, vrHMDPositioningMode *mode)
获取HMD重定位工作模式	bool vrConfigGetHMDRelocalizationMode(const vrConfig *config, vrHMDRelocalizationMode *mode)
获取控制器定位工作模式	bool vrConfigGetControllerPositioningMode(const vrConfig *config, vrControllerPositioningMode *mode)
获取控制器交互模式	bool vrConfigGetControllerInteractiveMode(const vrConfig *config, vrControllerInteractiveMode *mode)
获取HMD定位工作状态	bool vrConfigGetHMDPositionSystemWorkingStatus(const vrConfig *config, vrHMDPositionSystemWorkingStatus *status)
获取HMD云同步状态	bool vrConfigGetHMDCloudSyncStatus(const vrConfig *config, vrHMDCloudSyncStatus *status)
获取控制器定位工作状态	bool vrConfigGetControllerPositionSystemWorkingStatus(const vrConfig *config, vrControllerPositionSystemWorkingStatus *status)
获取设备的基本信息	bool vrConfigGetSystemInfo(const vrConfig *config, vrSystemInfo *system_info)
获取休眠倒计时设定	bool vrConfigGetDormancyTime(const vrConfig *config, int32_t *system_info)

6.2.2 相机参数

相机参数获取接口函数见表 66。

表 66 相机参数获取接口

接口描述	接口名称
获取相机实例列表	bool vrConfigGetCameraList(const vrConfig *config, vrCameraConfig **out_camera_list, int32_t *out_size)
获取相机的类型	bool vrConfigGetCameraType(const vrCameraConfig *camera_config, vrCameraType *out_camera_type)
获取相机帧率变化范围	bool vrConfigGetCameraFpsRange(const vrCameraConfig *camera_config, vrRange *fps_range)
获取相机当前的帧率	bool vrConfigGetCameraFps(const vrCameraConfig *camera_config, int32_t *out_fps)

表 66 相机参数获取接口（续）

接口描述	接口名称
获取相机图片数据的格式	bool vrConfigGetCameraImageFormat(const vrCameraConfig *camera_config, vrImageFormat *out_image_format)
获取相机图片的分辨率	bool vrConfigGetCameraImageDimensions(const vrCameraConfig *camera_config, vrCameraResolution *resolution)
获取相机内参-焦距与像素尺寸的乘积	bool vrConfigGetCameraIntrinsicsFocalLength(const vrCameraConfig *config, float *out_fx, float *out_fy)
获取相机内参-从成像平面到像素平面的平移	bool vrConfigGetCameraIntrinsicsPrincipalPoint(const vrCameraConfig *config, float *out_cx, float *out_cy)
获取相机内参-相机畸变类型	bool vrConfigGetCameraType(const vrCameraConfig *config, *cameraType)
获取相机内参-畸变参数	bool vrConfigGetCameraIntrinsicsTangentialDistortion(const vrCameraConfig *config, float *p1, float *p2)
获取相机曝光模式	bool vrConfigGetCameraExposureMode(const vrCameraConfig *camera_config, vrExposureMode *out_exposure_mode)
获取相机聚焦模式	bool vrConfigGetCameraFocusMode(const vrCameraConfig *camera_config, vrFocusMode *out_focus_mode)
获取相机外参-相机之间的位姿差	bool vrConfigGetCameraExtrinsicTransform(const vrCameraConfig *camera1, const vrCameraConfig *camera2, vrPosition *out_translation, vrQuaternion *out_quaternion)
获取相机实例列表	bool vrConfigGetCameraList(const vrConfig *config, vrCameraConfig **out_camera_list, int32_t *out_size)

6.2.3 传感器参数

传感器参数获取接口函数见表 67。

表 67 传感器参数获取接口

接口描述	接口名称
获取传感器的类型	bool vrConfigGetSensorType(const vrSensorConfig *config, vrSensorType *out_sensor_type)
获取传感器当前数据更新频率	bool vrConfigGetSensorUpdateFrequency(const vrSensorConfig *sensor_config, int32_t *out_freq)
获取传感器外参-与相机的位姿差	bool vrConfigGetSensorExtrinsicTransform(const vrSensorConfig *sensor_config, const int32_t camera_id, vrPosition *out_translation, vrQuaternion *out_quaternion)
获取目标传感器量程	bool vrConfigGetSensorMeasurementRange(const vrSensorConfig *sensor_config, vrRangeList *range_list, int32_t *out_size)
获取序号对应的量程实例	bool vrRangeListGetRange(const vrRangeList *range_list, const int32_t index, vrRnage *range)
获取量程类型	bool vrRangeGetType(const vrRange *range, vrRangeType *type)

表 67 传感器参数获取接口（续）

接口描述	接口名称
获取量程参数	bool vrRangeGetParams(const vrRange *range, int32_t *min, int32_t *max)
获取目标传感器标定数据列表	bool vrConfigGetSensorCorrectionList(const vrSensorConfig *sensor_config, vrCorrectionList *correction_list, int32_t *out_size)
获取 Index 对应的标定数据实例	bool vrCorrectionListGetCorrection(const vrCorrectionList *correction_list, const int32_t index, vrCorrection *correction)
获取标定数据类型	bool vrCorrectionGetType(const vrCorrection *correction, vrCorrectionType *correction_type)
获取标定参数	bool vrCorrectionGetParams(const vCorrection *correction, int32_t *param_x, int32_t *param_y, int32_t *param_z)

6.2.4 基站参数

基站参数获取接口函数见表 68。

表 68 基站参数获取接口

接口描述	接口名称
获取基站实例列表	bool vrBaseGetBase(vrBase *base)
获取基站配置	bool vrConfigGetBaseConfig(const vrConfig *config, vrBaseConfig *base_motor_config)
获取基站类型	bool vrConfigGetBaseType(const vrBaseConfig *config, vrBaseType *type)
获取基站状态	bool vrConfigGetBaseStatus(const vrBaseConfig *config, vrBaseStatus *status)

6.2.5 LED 参数

LED 参数获取接口函数见表 69。

表 69 LED 参数获取接口

接口描述	接口名称
获取控制器 LED 列表	bool vrDisplayGetLEDList(vrLEDList *led_list, int32_t *out_size)
获取 LED ID 列表	bool vrLEDLlistGetIdList(const vrLEDList *led_list, int32_t **id_list, int32_t *out_size)
获取 ID 对应的 LED 实例	bool vrLEDListGetLED(const vrLEDList *led_list, const int32_t led_id, vrLED *led)
获取目标 LED 类型	bool vrLEDGetType(const vrLED *led, vrLEDType *led_type)
获取目标 LED 状态	bool vrLEDGetStatus(const vrLED *led, vrLEDStatus *led_status)

6.2.6 声学单元参数

声学单元参数获取接口函数见表 70。

表 70 声学单元参数获取接口

接口描述	接口名称
获取控制器 SonicUnit 列表	bool vrDisplayGetSonicUnitList(vrSonicUnitList *sonicUnit_list, int32_t *out_size)
获取 SonicUnit ID 列表	bool vrSonicUnitLlistGetIdList(const vrSonicUnitList *sonicUnit_list, int32_t **id_list, int32_t *out_size)
获取 ID 对应的 SonicUnit 实例	bool vrSonicUnitListGetSonicUnit(const vrSonicUnitList *sonicUnit_list, const int32_t sonicUnit_id, vrSonicUnit *sonicUnit)
获取目标 SonicUnit 类型	bool vrSonicUnitGetType(const vrSonicUnit *sonicUnit, vrSonicUnitType *sonicUnit_type)
获取目标 SonicUnit 状态	bool vrSonicUnitGetStatus(const vrSonicUnit *sonicUnit, vrSonicUnitStatus*sonicUnit_status)

6.2.7 参数配置接口

参数配置接口函数见表 71。

表 71 参数配置接口

接口描述	接口名称
获取配置实例	bool vrConfigGetConfig(vrConfig *config)
获取系统功耗模式	bool vrConfigGetSystemCostMode(const vrConfig *config, vrSystemCostMode *mode)
获取系统睡眠模式	bool vrConfigGetSystemDormancyMode(const vrConfig *config, vrSystemDormancyMode *mode)
获取系统关机模式	bool vrConfigGetSystemShutDownMode(const vrConfig *config, vrSystemShutDownMode *mode)
获取 HMD 定位工作模式	bool vrConfigGetHMDPositioningMode(const vrConfig *config, vrHMDPositioningMode *mode)
获取 HMD 重定位工作模式	bool vrConfigGetHMDRelocalizationMode(const vrConfig *config, vrHMDRelocalizationMode *mode)
获取控制器定位工作模式	bool vrConfigGetControllerPositioningMode(const vrConfig *config, vrControllerPositioningMode *mode)
获取控制器交互模式	bool vrConfigGetControllerInteractiveMode(const vrConfig *config, vrControllerInteractiveMode *mode)
获取 HMD 定位工作状态	bool vrConfigGetHMDPositionSystemWorkingStatus(const vrConfig *config, vrHMDPositionSystemWorkingStatus *status)
获取 HMD 云同步状态	bool vrConfigGetHMDCloudSyncStatus(const vrConfig *config, vrHMDCloudSyncStatus *status)
获取控制器定位工作状态	bool vrConfigGetControllerPositionSystemWorkingStatus(const vrConfig *config, vrControllerPositionSystemWorkingStatus *status)
获取设备的基本信息	bool vrConfigGetSystemInfo(const vrConfig *config, vrSystemInfo *system_info)
获取休眠倒计时设定	bool vrConfigGetDormancyTime(const vrConfig *config, int32_t *system_info)

6.3 工作状态控制

工作状态控制接口函数见表 72。

表 72 工作状态控制接口函数

接口描述	接口名称
设置系统功耗模式	bool vrConfigSetSystemCostMode(vrConfig *config, const vrSystemCostMode *mode)
设置系统睡眠模式	bool vrConfigSetSystemDormancyMode(vrConfig *config, const vrSystemDormancyMode *mode)
设置系统关机模式	bool vrConfigSetSystemShutDownMode(vrConfig *config, const vrSystemShutDownMode *mode)
设置 HMD 定位工作模式	bool vrConfigSetHMDPositioningMode(vrConfig *config, const vrHMDPositioningMode *mode)
设置 HMD 重定位工作模式	bool vrConfigSetHMDRelocalizationMode(vrConfig *config, const vrHMDRelocalizationMode *mode)
设置控制器定位工作模式	bool vrConfigSetControllerPositioningMode(vrConfig *config, const vrControllerPositioningMode *mode)
设置控制器交互模式	bool vrConfigSetControllerInteractiveMode(vrConfig *config, const vrControllerInteractiveMode *mode)
设置 HMD 定位工作状态	bool vrConfigSetHMDPositionSystemWorkingStatus(vrConfig *config, const vrHMDPositionSystemWorkingStatus *status)
设置 HMD 云同步状态	bool vrConfigSetHMDCloudSyncStatus(vrConfig *config, const vrHMDCloudSyncStatus *status)
设置目标 LED 状态	bool vrLEDSetStatus(const vrLED *led, const vrLEDStatus led_status)
设置目标 SonicUnit 状态	bool vrSonicUnitSetStatus(const vrSonicUnit *sonicUnit, const vrSonicUnitStatus sonicUnit_status)
设置控制器定位工作状态	bool vrConfigSetControllerPositionSystemWorkingStatus(vrConfig *config, const vrControllerPositionSystemWorkingStatus status)
设置休眠计时	bool vrConfigSetDormancyTime(const float time)
设置控制器震动参数	bool vrControllerSetVibration(const float vibrate)
重启控制(请增加获取 device_id 的接口)	bool vrDeviceRebootDevice(const int32_t device_id)
关机控制	bool vrDeviceShutDownDevice(const int32_t device_id)
静止锁设置	bool vrDeviceStationaryLock(const int32_t device_id, const uint8_t lock_state)
设置系统功耗模式	bool vrConfigSetSystemCostMode(vrConfig *config, const vrSystemCostMode *mode)

6.4 定位设备状态信息获取

定位设备状态信息获取接口见表 73。

表 73 定位设备状态信息获取接口函数

接口描述	接口名称
获取定位设备列表	bool vrDeviceGetPositionableDeviceList(vrPositionableDevice **device, int32_t *out_size)
获取定位设备类型	bool vrDeviceGetPositionableDeviceType(const vrPositionableDevice *device, vrPositionableDeviceType *type)
获取定位设备位姿及运动状态	bool vrDeviceGetPositionableDeviceMotionPose(const vrPositionableDevice *device, vrPose *pose, vrMotionData *motion)
获取当前的 Frame Id 列表	bool vrFrameGetFrameIdList(int32_t **id_list, int32_t *out_size)
获取 ID 对应的 Frame 实例	bool vrFrameGetFrame(const int32_t frame_id, vrFrame *frame)
获取最新 Frame 实例	bool vrFrameGetCurrentFrame(vrFrame *frame)
获取目标帧的相机图片数据	bool vrFrameAcquireCameraImage(const vrFrame *frame, const int32_t camera_id, vrImage **out_image, int32_t *out_size)
获取目标帧的点云数据	bool vrFrameAcquirePointCloud(const vrFrame *frame, vrPointCloud **out_point_cloud, int32_t *out_size)
获取目标帧的位姿及其估计质量	bool vrFrameGetPose(const vrFrame *frame, vrPose *out_pose, float *quality)
获取目标帧的时间戳	bool vrFrameGetTimestamp(const vrFrame *frame, int64_t *out_timestamp_ns)
获取平面实例列表及个数	bool vrFrameGetPlaneList(const vrFrame *frame, vrPlane **out_plane_list, int32_t *out_size)
获取平面类型	bool vrPlaneGetPlaneType(const vrPlane *plane, vrPlaneType *out_type)
获取平面法向量	bool vrPlaneGetNormal(const vrPlane *plane, vrPosition* out_plane_normal)
获取平面中心点位姿	bool vrPlaneGetCenterPose(const vrPlane *plane, vrPose* center_pose)
获取平面包围盒的延伸大小	bool vrPlaneGetExtent(const vrPlane *plane, float* out_extent_xyz)
获取平面边界多边形的定点坐标数量,每个顶点包含 3 个浮点数量	bool vrPlaneGetPolygon3DSize(const vrPlane *plane, vrPosition **position_list, int32_t *out_size)
判断给定点是否在平面包围盒内	bool vrPlaneIsPoseInExtent(const vrPlane *plane, const vrPose* pose, int* out_pose_in_extents)
判断给定点是否在平面多边形内	bool vrPlaneIsPoseInPolygon(const vrPlane *plane, const vrPose* pose, int* out_pose_in_polygon)
获取控制器列表	bool vrControllerGetControllerList(vrControllerList *controller_list, int32_t *out_size)
获取控制器 Id 列表	bool vrControllerGetControllerIdList(const vrControllerList *controller_list, int32_t **id_list, int32_t *out_size)
获取 ID 对应的控制器实例	bool vrControllerGetController(const vrControllerList *controller_list, const int32_t controller_id, vrController *controller)
获取目标控制器的类型	bool vrControllerGetType(const vrController *controller, vrControllerType *type)
获取目标控制器的位姿	bool vrControllerGetPose(const vrController *controller, vrPose *pose, float *quality)
获取 ID 对应的传感器实例	bool vrSensorGetSensor(const int32_t sensor_id, vrSensor *sensor)
获取传感器数据	bool vrSensorGetSensorData(const vrSensor *sensor, uint8_t **data, int32_t *data_size)

参 考 文 献

- [1] GB/T 38259—2019 信息技术 虚拟现实头戴式显示设备通用规范
 - [2] IEC 63308:2021 Virtual reality equipment and systems—Market, technology and standards requirements
-

中 华 人 民 共 和 国
国 家 标 准
虚拟现实设备接口 定位设备
GB/T 44220—2024

*

中国标准出版社出版发行
北京市朝阳区和平里西街甲2号(100029)
北京市西城区三里河北街16号(100045)

网址: www.spc.net.cn

服务热线: 400-168-0010

2024年7月第一版

*

书号: 155066 · 1-76839

版权专有 侵权必究



GB/T 44220-2024