

团 体 标 准

T/CESA XXXX—2020

信息技术 移动设备增强现实系统应用接口

Information technology - Application interfaces of augmented reality system for
mobile devices

征求意见稿

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。

已授权的专利证明材料为专利证书复印件或扉页，已公开但尚未授权的专利申请证明材料为专利公开通知书复印件或扉页，未公开的专利申请的证明材料为专利申请号和申请日期。

2020-XX- XX 发布

2020-XX- XX 实施

中国电子工业标准化技术协会 发布



版权保护文件

版权所有归属于该标准的发布机构，除非有其他规定，否则未经许可，此发行物及其章节不得以其他形式或任何手段进行复制、再版或使用，包括电子版，影印件，或发布在互联网及内部网络等。使用许可可于发布机构获取。

目 次

前 言.....	IV
1 范围.....	1
2 规范性引用文件.....	1
3 术语和定义.....	1
4 缩略语.....	2
5 总体结构.....	2
6 数据类型.....	3
6.1 结果.....	3
6.2 设备.....	3
6.3 数据流模式.....	3
6.4 节点.....	4
6.5 算法.....	4
6.6 算法模式.....	4
6.7 可用性.....	4
6.8 跟踪状态.....	5
6.9 跟踪运行模式.....	5
6.10 世界坐标对齐模式.....	5
6.11 顶点格式.....	6
6.12 像素格式.....	6
6.13 参数.....	6
6.14 左右眼.....	7
6.15 光照估计模式.....	7
6.16 光照估计状态.....	7
6.17 求交查询模式.....	7
6.18 云锚点状态.....	7
6.19 平面.....	8
6.20 手势.....	8
6.21 左右手.....	9
6.22 手掌心朝向.....	9
7 接口.....	9
7.1 功能扩展.....	9
7.2 能力检测.....	9
7.3 位姿.....	10
7.4 配置.....	11
7.5 增强现实会话.....	14

7.6	增强现实世界地图.....	21
7.7	帧画面.....	22
7.8	相机.....	27
7.9	相机内参.....	29
7.10	点云.....	30
7.11	稠密重建.....	30
7.12	光照估计.....	32
7.13	锚点列表.....	33
7.14	锚点.....	34
7.15	节点列表.....	35
7.16	节点.....	36
7.17	平面节点.....	37
7.18	查询结果列表.....	39
7.19	查询结果.....	40
7.20	参考图数据库.....	41
7.21	图像节点.....	43
7.22	手势节点.....	44

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。

本文件由浙江商汤科技开发有限公司提出。

本文件由中国电子技术标准化研究院、中国电子工业标准化技术协会归口。

本文件起草单位：浙江商汤科技开发有限公司、中国电子技术标准化研究院、浙江大学、山东大学、北京邮电大学、OPPO广东移动通信有限公司、北京小米移动软件有限公司、北京爱奇艺智能科技有限公司、北京百度网讯科技有限公司、网易（杭州）网络有限公司、杭州易现先进科技有限公司。

本文件主要起草人： 。

信息技术 移动设备增强现实系统应用接口

1 范围

本文件规定了移动设备增强现实系统应用开发的数据类型和接口。
本文件适用于移动设备增强现实系统的设计、开发和应用。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 38247-2019 信息技术 增强现实 术语
T/CESA XXXXX 信息技术 移动设备增强现实系统技术规范

3 术语和定义

GB/T 38247-2019和T/CESA XXXXX界定的以及下列术语和定义适合于本文件。为了便于使用，以下重复列出了GB/T 38247—2019中的某些术语和定义。

3.1

参考图像 reference image

在环境中可被识别的二维图像。

[来源：GB/T 38247-2019，2.2.18]

3.2

增强现实会话 augmented reality session

负责管理增强现实系统的整个运行过程，包括创建、启动、更新、关闭、销毁等各个过程。

3.3

增强现实地图 augmented reality map

增强现实运行过程中构建的地图结构，包括检测到的点云、平面节点、图像节点等信息，以及用户创建的锚点。

3.4

节点 node

增强现实运行过程中识别出的平面、图像、手势等具体对象信息。

3.5

点云 point cloud

在增强现实应用中，用于恢复三维场景而使用的场景物体表面的三维点数据集。

[来源：GB/T 38247-2019, 2.2.47]

3.6

稠密网格 dense mesh

稠密三维重建过程中，恢复的场景三角形网格。

4 缩略语

下列缩略语适用于本文件。

6DoF: 六自由度 (Six Degrees of Freedom)

AR: 增强现实 (Augmented Reality)

API: 应用程序编程接口 (Application Programming Interface)

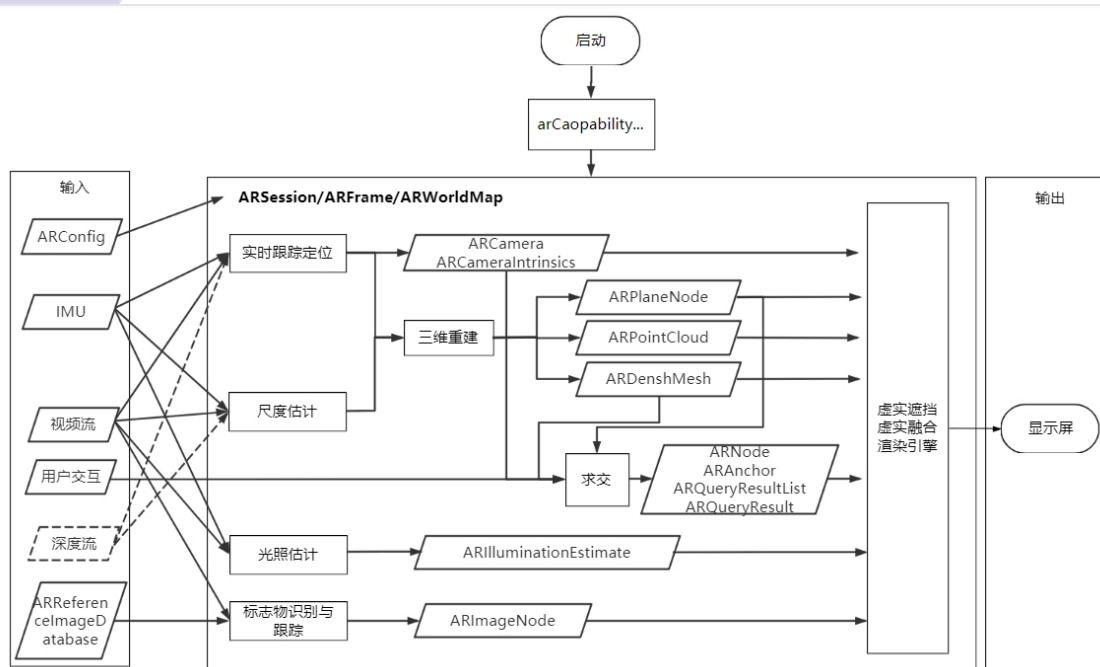
FPS: 每秒帧数 (Frames Per Second)

IMU: 惯性测量单元 (Inertial Measurement Unit)

SLAM: 同步定位与地图构建 (Simultaneous Localization And Mapping)

5 总体结构

增强现实系统总体结构如图1所示。



说明:

- > 可选模块
- > 必选模块

图 1 增强现实系统总体结构

6 数据类型

6.1 结果

结果数据类型命名为ARResult，其值定义如表1 所示。

表 1 结果数据类型 ARResult 定义

ARResult 值	描述
ARRESULT_SUCCESS=0	成功或者正确。
ARRESULT_ERROR_INVALID_ARGUMENT=1	包含无效参数。
ARRESULT_ERROR_FATAL=2	严重错误。
ARRESULT_ERROR_SESSION_PAUSED=3	当前操作 ARSession 暂停状态错误。
ARRESULT_ERROR_SESSION_NOT_PAUSED=4	当前操作 ARSession 未暂停状态错误
ARRESULT_ERROR_NOT_TRACKING=5	目前不处于跟踪状态导致的错误。
ARRESULT_ERROR_TEXTURE_NOT_SET=6	未设置纹理参数导致的错误。
ARRESULT_ERROR_UNSUPPORTED_VIDEOFORMAT=7	指定的视频格式不支持错误。
ARRESULT_ERROR_CAMERA_NOT_AVAILABLE=8	摄像机服务不可用错误。
ARRESULT_ERROR_CAMERA_PERMISSION_NOT_GRANTED=9	摄像机服务未授权导致的错误。
ARRESULT_ERROR_UNSUPPORTED_CONFIG=10	设置的配置选项不支持导致的错误。
ARRESULT_ERROR_RESOURCE_EXHAUSTED=11	没有可用资源导致的错误。
ARRESULT_ERROR_CLOUD_ANCHORS_NOT_CONFIGURED=12	没有对云锚点模式进行设置导致的错误。
ARRESULT_ERROR_ANCHOR_NOT_SUPPORTED_FOR_HOSTING=13	该锚点不支持托管导致的错误。
ARRESULT_ERROR_DEVICE_UNUPPORT=14	移动设备不支持增强现实导致的错误。
ARRESULT_ERROR_ALGORITHM_UNUPPORT=15	使用不支持的算法导致的错误。
ARRESULT_ERROR_APPLICATION_AUTHORIZED=16	该应用未授权使用增强现实功能导致的错误。

6.2 设备

设备数据类型命名为ARDeviceType，其值定义如表2所示。

表 2 设备数据类型 ARDeviceType 定义

ARDeviceType 值	描述
ARDEVICE_TYPE_HANDHOLD=100	手持式移动设备。
ARDEVICE_TYPE_HEADMOUNTED=200	头戴式移动设备。

6.3 数据流模式

数据流模式数据类型命名为ARStreamMode，其值定义如表3所示。

表 3 数据流模式数据类型 ARStreamMode 定义

ARStreamMode 值	描述
ARSTREAM_MODE_FRONT_RGB=100	数据流采用前置 RGB 相机模式。
ARSTREAM_MODE_BACK_RGB=2000	数据流采用后置 RGB 相机模式。
ARSTREAM_MODE_BACK_RGBD=2100	数据流采用后置 RGBD 相机模式。
ARSTREAM_MODE_BACK_STEREO=2200	数据流采用后置立体相机模式。

6.4 节点

节点数据类型命名为 ARNodeType，其值定义如表4所示。

表 4 节点数据类型 ARNodeType 定义

ARNodeType 值	描述
ARNODE_TYPE_NOT_VALID=0	节点类型无效。
ARNODE_TYPE_PLANE=1	平面节点类型。
ARNODE_TYPE_IMAGE=2	图像节点类型。
ARNODE_TYPE_HAND_GESTURE=3	手势节点类型。

6.5 算法

算法数据类型命名为 ARAlgorithmType，其值定义如表5所示。

表 5 算法数据类型 ARAlgorithmType 定义

ARAlgorithmType 值	描述
ARALGORITHM_TYPE_UNDEFINED=0	非法算法类型。
ARALGORITHM_TYPE_SLAM=1	SLAM 算法类型。
ARALGORITHM_TYPE_HAND_GESTURE=2	手势算法类型。
ARALGORITHM_TYPE_DENSE_RECON=3	稠密网格重建算法类型。
ARALGORITHM_TYPE_IMAGE_TRACKING=4	图像跟踪算法类型。
ARALGORITHM_TYPE_PLANE_DETECTION=5	平面检测算法类型。
ARALGORITHM_TYPE_ILLUMINATION_ESTIMATION=6	光照估计算法类型。
ARALGORITHM_TYPE_CLOUD_ANCHOR=7	云锚点算法类型。

6.6 算法模式

算法模式数据类型命名为 ARAlgorithmMode，其值定义如表6所示。

表 6 算法模式数据类型 ARAlgorithmMode 定义

ARAlgorithmMode 值	描述
ARALGORITHM_MODE_DISABLE=0	算法关闭模式。
ARALGORITHM_MODE_ENABLE=1	算法激活模式。

6.7 可用性

可用性数据类型命名为ARAvailability，其值定义如表7所示。

表 7 可用性数据类型 ARAvailability 定义

ARAvailability 值	描述
ARAVAILABILITY_UNKNOWN_ERROR=0	未知错误。
ARAVAILABILITY_UNSUPPORTED_DEVICE_NOT_CAPABLE=100	设备不支持 AR。
ARAVAILABILITY_SUPPORTED_NOT_INSTALLED=201	设备支持，模块未安装导致不可用。
ARAVAILABILITY_SUPPORTED_VERSION_TOO_OLD=202	设备支持，模块版本太旧导致不可用。
ARAVAILABILITY_SUPPORTED=203	设备支持，并且可用。

6.8 跟踪状态

跟踪状态数据类型命名为ARTrackingState，其值定义如表8所示。

表 8 跟踪状态数据类型 ARTrackingState 定义

ARTrackingState 值	描述
ARTRACKING_STATE_SUCCESS=0	正常跟踪状态。
ARTRACKING_STATE_LOST=1	跟踪丢失状态。
ARTRACKING_STATE_STOPPED=2	停止跟踪状态。
ARTRACKING_STATE_INITIALIZING=10	跟踪初始化中状态。

6.9 跟踪运行模式

跟踪运行模式数据类型命名为ARTrackingRunMode，其值定义如表9所示。

表 9 跟踪运行模式数据类型 ARTrackingRunMode 定义

ARTrackingRunMode 值	描述
ARTRACKING_RUN_MODE_AUTO=0	自动跟踪模式，增强现实会话恢复运行后，算法自动启动运行。
ARTRACKING_RUN_MODE_MANUAL=1	手动跟踪模式，增强现实会话恢复运行后，算法默认不启动，需要手动启动运行。

6.10 世界坐标对齐模式

世界坐标对齐模式数据类型命名为ARWorldAlignmentMode，其值定义如表10所示。

表 10 世界坐标对齐模式数据类型 ARWorldAlignmentMode 定义

ARWorldAlignmentMode 值	描述
ARWORLD_ALIGNMENT_MODE_GRAVITY=0	根据初始化时候屏幕的朝向和重力方向构建世界坐标系。
ARWORLD_ALIGNMENT_MODE_GRAVITY_HEADING=1	根据指南针和重力方向构建世界坐标系，y 轴同重力方向平行，(0, -1, 0)朝下，z 轴同南北方向平行，(0, 0, 1)指向所在位置的朝南方向。

6.11 顶点格式

顶点格式数据类型命名为ARVertexFormat，其值定义如表11所示。

表 11 顶点格式数据类型 ARVertexFormat 定义

ARVertexFormat 值	描述
ARVERTEX_FORMAT_3D_POSITION=0	顶点数据包含 3 个 float 值，表示该点坐标。
ARVERTEX_FORMAT_3D_POSITION_NORMAL=1	顶点数据包含 6 个 float 值，表示该点坐标和经过该点的平面法线向量。
ARVERTEX_FORMAT_3D_POSITION_NORMAL_COLOR=2	顶点数据包含 10 个 float 值，表示该点坐标、经过该点的平面法线向量和该点的 RGBA 数据。
ARVERTEX_FORMAT_NOT_VALID=3	非法顶点数据。

6.12 像素格式

像素格式数据类型命名为ARPixelFormat，其值定义如表12所示。

表 12 像素格式数据类型 ARPixelFormat 定义

ARPixelFormat 值	描述
ARPIXEL_FORMAT_UNKNOWN=-1	非法图像数据。
ARPIXEL_FORMAT_LUMINANCE8=0	图像数据采用 LUMINANCE8 格式。
ARPIXEL_FORMAT_RG16=1	图像数据采用 RG16 格式。
ARPIXEL_FORMAT_RGB24=2	图像数据采用 RGB24 格式。
ARPIXEL_FORMAT_RGBA32=3	图像数据采用 RGBA32 格式。
ARPIXEL_FORMAT_DEPTH16=4	图像数据采用 Depth16 格式。
ARPIXEL_FORMAT_YUV_NV21=1000	图像数据采用 YUV NV21 格式。

6.13 参数

参数数据类型命名为ARParameterEnum，其值定义如表13所示。

表 13 参数数据类型 ARParameterEnum 定义

ARParameterEnum 值	描述
ARPARAMETER_VIDEO_VERTICAL_FOV=50	视频垂直方向 FOV。
ARPARAMETER_ALGORITHM_SLAM_INFO=100	SLAM 算法运行信息。
ARPRARMETER_ALGORITHM_SLAM_MAP_QUALITY=101	SLAM 地图质量。
ARPARAMETER_ALGORITHM_SLAM_DETAIL_STATE=102	SLAM 算法内部详细的运行状态。
ARPARAMETER_CAMERA_TEXTURE_TARGET=3000	视频流纹理 ID。

6.14 左右眼

左右眼数据类型命名为AREyeSide，其值定义如表14所示。

表 14 左右眼数据类型 AREyeSide 定义

AREyeSide 值	描述
AREYE_LEFT=0	左眼。
AREYE_RIGHT=1	右眼。

6.15 光照估计模式

光照估计模式数据类型命名为ARilluminationEstimateMode，其值定义如表15 所示。

表 15 光照估计模式数据类型 ARilluminationEstimateMode 定义

ARilluminationEstimateMode 值	描述
ARILLUMINATION_ESTIMATE_MODE_AMBIENT_INTENSITY=1	环境光亮度模式，包含单一亮度值。

6.16 光照估计状态

光照估计状态数据类型命名为ARilluminationEstimateState，其定义如表16 所示。

表 16 光照估计状态数据类型 ARilluminationEstimateState 定义

ARilluminationEstimateState 值	描述
ARILLUMINATION_ESTIMATE_STATE_NOT_VALID = 0	光照估计结果暂时还不可用。
ARILLUMINATION_ESTIMATE_STATE_VALID = 1	光照估计结果可用

6.17 求交查询模式

求交查询模式数据类型命名为ARQueryMode，其值定义如表17 所示。

表 17 求交查询模式数据类型 ARQueryMode 定义

ARQueryMode 值	描述
ARQUERY_MODE_POLYGON_ONLY=0	只能跟多边形平面求交查询。
ARQUERY_MODE_POLYGON_AND_HORIZONPLANE=1	能跟多边形平面以及水平多边形所在整个无限平面求交查询。
ARQUERY_MODE_POLYGON_PERSISTENCE=2	能跟多边形平面以及多边形平面所在的无限平面求交查询，多边形平面优先。

6.18 云锚点状态

云锚点状态数据类型命名为ARCloudAnchorState，其值定义如表18 所示。

表 18 云锚点状态数据类型 ARCloudAnchorState 定义

ARCloudAnchorState 值	描述
ARCLOUD_ANCHOR_STATE_NONE=0	该锚点不是云锚点。
ARCLOUD_ANCHOR_STATE_TASK_IN_PROGRESS=1	正在托管或是解析中。
ARCLOUD_ANCHOR_STATE_SUCCESS=2	托管或解析成功。
ARCLOUD_ANCHOR_STATE_ERROR_INTERNAL=-1	托管或解析发生内部错误。
ARCLOUD_ANCHOR_STATE_ERROR_NOT_AUTHORIZED=-2	托管或解析没有被授权。
ARCLOUD_ANCHOR_STATE_ERROR_SERVICE_UNAVAILABLE=-3	当前服务不可用。
ARCLOUD_ANCHOR_STATE_ERROR_RESOURCE_EXHAUSTED=-4	资源已经被用尽。
ARCLOUD_ANCHOR_STATE_ERROR_HOSTING_DATASET_PROCESSING_FAILED=-5	云锚点数据错误，不能托管。
ARCLOUD_ANCHOR_STATE_ERROR_CLOUD_ID_NOT_FOUND=-6	该云锚点 ID 没有被找到。
ARCLOUD_ANCHOR_STATE_ERROR_RESOLVING_LOCALIZATION_NO_MATCH=-7	解析失败，环境无法匹配。
ARCLOUD_ANCHOR_STATE_ERROR_RESOLVING_SDK_VERSION_TOO_OLD=-8	解析锚点的 SDK 版本过旧。
ARCLOUD_ANCHOR_STATE_ERROR_RESOLVING_SDK_VERSION_TOO_NEW=-9	解析锚点的 SDK 版本过新。
ARCLOUD_ANCHOR_STATE_ERROR_UNKNOWN=-11	未知错误。

6.19 平面

平面数据类型命名为ARPlaneType，其值定义如表19 所示。

表 19 平面数据类型 ARPlaneType 定义

ARPlaneType 值	描述
ARPLANE_TYPE_UPWARD_FACING=0	水平平面朝上类型。
ARPLANE_TYPE_DOWNWARD_FACING=1	水平平面朝下类型。
ARPLANE_TYPE_VERTICAL=2	垂直平面类型。
ARPLANE_TYPE_NOT_VALID=3	无效平面类型。

6.20 手势

手势数据类型命名为ARHandGestureType，其值定义如表20 所示。

表 20 手势数据类型 ARHandGestureType 定义

ARHandGestureType 值	描述
ARHAND_GESTURE_TYPE_UNKNOWN=-1	无效手势获取类型。
ARHAND_GESTURE_TYPE_OK=0	OK 手势获取类型。
ARHAND_GESTURE_TYPE_SCISSORS=1	剪刀手势获取类型。
ARHAND_GESTURE_TYPE_THUMBS_UP=2	竖起大拇指手势获取类型。
ARHAND_GESTURE_TYPE_PAPER=3	布手势获取类型。
ARHAND_GESTURE_TYPE_GUN=4	手枪手势获取类型。
ARHAND_GESTURE_TYPE_ROCK=5	石头手势获取类型。
ARHAND_GESTURE_TYPE_FINGER_HEART=6	比心手势获取类型。
ARHAND_GESTURE_TYPE_FINGERTIP=7	指尖手势获取类型。
ARHAND_GESTURE_TYPE_WELL_PLAYED=8	做得好手势获取类型。

ARHAND_GESTURE_TYPE_THREE_FINGERS=9	3个手指手势获取类型。
ARHAND_GESTURE_TYPE_FOUR_FINGERS=10	4个手指手势获取类型。
ARHAND_GESTURE_TYPE_I_LOVE_YOU=11	我爱你手势获取类型。
ARHAND_GESTURE_TYPE_INDEX_FINGER_AND_LITTLE_FINGER=12	竖起食指与小拇指手势获取类型。
ARHAND_GESTURE_TYPE_LITTLE_FINGER=13	小拇指手势获取类型。

6.21 左右手

左右手数据类型命名为ARHandSide，其值定义如表21所示。

表 21 左右手数据类型 ARHandSide 定义

ARHandSide 值	描述
ARHAND_SIDE_HAND_UNKNOWN=-1	无法区分左右手。
ARHAND_SIDE_RIGHT_HAND=0	右手。
ARHAND_SIDE_LEFT_HAND=1	左手。

6.22 手掌心朝向

手掌心朝向数据类型命名为ARHandTowards，其值定义如表22所示。

表 22 手掌心朝向数据类型 ARHandTowards 定义

ARHandTowards 值	描述
ARHAND_TOWARDS_UNKNOWN=-1	无效的手朝向。
ARHAND_TOWARDS_PALM=0	手心方向。
ARHAND_TOWARDS_THE_BACK_OF_HAND=1	手背方向。
ARHAND_TOWARDS_SIDE_HAND=2	侧手方向。

7 接口

7.1 功能扩展

7.1.1 获取扩展函数地址

接口定义：PFN_xrVoidFunction arGetProcAddress(const char* name)

功能描述：通过给定的字符串，获取功能扩展的函数指针。

输入参数：

类型：const char*，值：函数指针的名字。

输出参数：

类型：PFN_xrVoidFunction，值：typedef void (ARAPI_PTR *PFN_arVoidFunction)(void)，定义了函数指针的类型，ARAPI_PTR 定义的是调用方式，当无法获取到指定名字的函数指针的时候，返回空指针。

7.2 能力检测

7.2.1 检测设备是否支持

接口定义：void arCapabilityCheckAvailability(void* env, void* context, ARAvailability*

out_availability)

功能描述：检测设备是否支持运行增强现实模块。

输入参数：

类型：void*，值：运行环境。

类型：void*，值：运行上下文。

输出参数：

类型：ARAvailability*，值：是否可用，参照 ARAvailability 定义。

7.2.2 检测是否支持算法

接口定义：ARResult arCapabilityCheckAlgorithm(void* env, void* context, ARAlgorithmType type, ARStreamMode mode)

功能描述：检测设备是否支持给定数据流模式的算法，返回 ARRESULT_SUCCESS 表示成功。

输入参数：

类型：void*，值：运行环境。

类型：void*，值：运行上下文。

类型：ARAlgorithmType，值：算法类型，参考 ARAlgorithmType 定义。

类型：ARStreamMode，值：数据流模式，参考 ARStreamMode 定义。

输出参数：

类型：ARResult，值：返回值，参考 ARResult 定义。

7.2.3 检测是否授权

接口定义：ARResult arCapabilityCheckAuthorized(void* env, void* context, const char* appId)

功能描述：检测应用程序是否授权运行增强现实模块，返回 ARRESULT_SUCCESS 表示成功。

输入参数：

类型：void*，值：运行环境。

类型：void*，值：运行上下文。

类型：const char*，值：应用程序授权字符串。

输出参数：

类型：ARResult，值：返回值，参考 ARResult 定义。

7.3 位姿

7.3.1 创建位姿

接口定义：void arPoseCreate(const float* pose_raw, ARPose** out_pose)

功能描述：创建位姿结构，数组参数为 6DoF 信息 (qx, qy, qz, qw, tx, ty, tz)，数组长度为 7。

输入参数：

类型：const float*，值：6DoF 信息。

输出参数：

类型：ARPose**，值：创建新的位姿结构，结构地址保存到*out_pose 中。

7.3.2 释放位姿

接口定义：void arPoseDestroy(ARPose* pose)

功能描述：销毁位姿结构。

输入参数：

类型：ARPose*，值：位姿结构指针。

输出参数：不作定义。

7.3.3 获取位姿结构的 6DoF 信息

接口定义：void arPoseGetPoseRaw(const ARPose* pose, float* out_pose_raw)

功能描述：获取位姿结构对应的 6DoF 信息 (qx, qy, qz, qw, tx, ty, tz)，数组长度为 7。

输入参数：

类型：const ARPose*，值：位姿结构指针。

输出参数：

类型：float*，值：6DoF 信息。

7.3.4 获取位姿结构的矩阵

接口定义：void arPoseGetMatrix(const ARPose* pose, float* out_matrix_col_major_4x4)

功能描述：获取位姿结构对应的 4×4 矩阵。

输入参数：

类型：const ARPose*，值：位姿结构指针。

输出参数：

类型：float*，值：矩阵，该矩阵为列矩阵形式。

7.4 配置

7.4.1 创建配置

接口定义：void arConfigCreate(ARConfig** out_config)

功能描述：创建配置结构。

输入参数：不做定义。

输出参数：

类型：ARConfig**，值：创建新的配置结构，结构地址保存到*out_config 中。

7.4.2 销毁配置

接口定义：void arConfigDestroy(ARConfig* config)

功能描述：销毁配置结构。

输入参数：

类型：ARConfig*，值：配置结构指针。

输出参数：不做定义。

7.4.3 获取配置中的设备类型

接口定义：void arConfigGetDeviceType(const ARConfig* config, ARDeviceType* out_device_type)

功能描述：获取配置结构中的设备类型。

输入参数：

类型：const ARConfig*，值：配置结构指针。

输出参数:

类型: ARDeviceType*, 值: 设备类型指针, 参考 ARDeviceType 定义。

7.4.4 设置配置中的设备类型

接口定义: void arConfigSetDeviceType(ARConfig* config, ARDeviceType device_type)

功能描述: 设置配置结构中的设备类型。

输入参数:

类型: ARConfig*, 值: 配置结构指针。

类型: ARDeviceType, 值: 设备类型, 参考 ARDeviceType 定义。

输出参数: 不做定义。

7.4.5 获取配置中的算法数据流模式

接口定义: void arConfigGetAlgorithmStreamMode(const ARConfig* config, ARAlgorithmType type, ARStreamMode* out_stream_mode)

功能描述: 获取配置结构中指定算法类型的数据流模式。

输入参数:

类型: const ARConfig*, 值: 配置结构指针。

类型: ARAlgorithmType, 值: 算法类型, 参考 ARAlgorithmType 定义。

输出参数:

类型: ARStreamMode*, 值: 数据流模式指针, 参考 ARStreamMode 定义。

7.4.6 设置配置中的算法数据流模式

接口定义: void arConfigSetAlgorithmStreamMode(ARConfig* config, ARAlgorithmType type, ARStreamMode stream_mode)

功能描述: 设置配置结构中指定算法类型的数据流模式。

输入参数:

类型: ARConfig*, 值: 配置结构指针。

类型: ARAlgorithmType, 值: 算法类型, 参考 ARAlgorithmType 定义。

类型: ARStreamMode, 值: 数据流模式, 参考 ARStreamMode 定义。

输出参数: 不做定义。

7.4.7 获取配置中的算法开关模式

接口定义: void arConfigGetAlgorithmMode(const ARConfig* config, ARAlgorithmType type, ARAlgorithmMode* out_algorithm_mode)

功能描述: 获取配置结构中指定算法类型的算法开关模式。

输入参数:

类型: const ARConfig*, 值: 配置结构指针。

类型: ARAlgorithmType, 值: 算法类型, 参考 ARAlgorithmType 定义。

输出参数:

类型: ARAlgorithmMode*, 值: 算法开关模式指针, 参考 ARAlgorithmMode 定义。

7.4.8 设置配置中的算法开关模式

接口定义: void arConfigSetAlgorithmMode(ARConfig* config, ARAlgorithmType type,

ARAlgorithmMode algorithm_mode)

功能描述：设置配置结构中指定算法类型的算法开关模式。

输入参数：

类型：ARConfig*，值：配置结构指针。

类型：ARAlgorithmType，值：算法类型，参考 ARAlgorithmType 定义。

类型：ARAlgorithmMode，值：算法开关模式，参考 ARAlgorithmMode 定义。

输出参数：不做定义。

7.4.9 获取配置中光照估计模式

接口定义：void arConfigGetIlluminationEstimateMode(const ARConfig* config, ARIlluminationEstimateMode* out_mode)

功能描述：获取配置结构中的光照估计模式。

输入参数：

类型：const ARConfig*，值：配置结构指针。

输出参数：

类型：ARIlluminationEstimateMode*，值：光照估计模式指针，参考 ARIlluminationEstimateMode 定义。

7.4.10 设置配置中光照估计模式

接口定义：void arConfigSetIlluminationEstimateMode(ARConfig* config, ARIlluminationEstimateMode mode)

功能描述：设置配置结构中的光照估计模式。

输入参数：

类型：ARConfig*，值：配置结构指针。

类型：ARIlluminationEstimateMode，值：光照估计模式，参考 ARIlluminationEstimateMode 定义。

输出参数：不做定义。

7.4.11 获取配置中的跟踪运行模式

接口定义：void arConfigGetTrackingRunMode(const ARConfig* config, ARTrackingRunMode* tracking_run_mode)

功能描述：获取配置结构中的跟踪运行模式。

输入参数：

类型：const ARConfig*，值：配置结构指针。

输出参数：

类型：ARTrackingRunMode*，值：跟踪运行模式指针，参考 ARTrackingRunMode 定义。

7.4.12 设置配置中的跟踪运行模式

接口定义：void arConfigSetTrackingRunMode(ARConfig* config, ARTrackingRunMode tracking_run_mode)

功能描述：设置配置结构中的跟踪运行模式。

输入参数：

类型：ARConfig*，值：配置结构指针。

类型：ARTrackingRunMode，值：跟踪运行模式，参考 ARTrackingRunMode 定义。
输出参数：不做定义。

7.4.13 获取配置中的世界坐标对齐模式

接口定义：void arConfigGetWorldAlignmentMode(const ARConfig* config, ARWorldAlignmentMode* world_alignment_mode)

功能描述：获取配置结构中的世界坐标对齐模式。

输入参数：

类型：const ARConfig*，值：配置结构指针。

输出参数：

类型：ARWorldAlignmentMode*，值：世界坐标对齐模式指针，参考 ARWorldAlignmentMode 定义。

7.4.14 设置配置中的世界坐标对齐模式

接口定义：void arConfigSetWorldAlignmentMode(ARConfig* config, ARWorldAlignmentMode world_alignment_mode)

功能描述：设置配置结构中的世界坐标对齐模式。

输入参数：

类型：ARConfig*，值：配置结构指针。

类型：ARWorldAlignmentMode，值：世界坐标对齐模式，参考 ARWorldAlignmentMode 定义。

输出参数：不做定义。

7.4.15 获取配置中的参考图像数据集

接口定义：void arConfigGetReferenceImageDatabase(const ARConfig* config, ARReferenceImageDatabase* out_image_database)

功能描述：获取配置结构中的参考图像数据集结构。

输入参数：

类型：const ARConfig*，值：配置结构指针。

输出参数：

类型：ARReferenceImageDatabase*，值：参考图像数据集结构的指针。

7.4.16 设置配置中的参考图像数据集

接口定义：void arConfigSetReferenceImageDatabase(ARConfig* config, const ARReferenceImageDatabase* image_database)

功能描述：设置配置结构中的参考图像数据集结构。

输入参数：

类型：ARConfig*，值：配置结构指针。

类型：const ARReferenceImageDatabase*，值：参考图像数据集结构指针。

输出参数：不做定义。

7.5 增强现实会话

7.5.1 创建增强现实会话

接口定义: `ARResult arSessionCreate(void* env, void* context, ARSession** out_session)`

功能描述: 创建增强现实会话结构, 返回 `ARRESULT_SUCCESS` 表示成功。

输入参数:

类型: `void*`, 值: 运行环境。

类型: `void*`, 值: 运行上下文。

输出参数:

类型: `ARSession**`, 值: 创建新的增强现实会话结构, 结构地址保存在 `*out_session` 中, 如果创建失败, `*out_session` 指向的值为空指针。

类型: `ARResult`, 值: 返回值, 参考 `ARResult` 定义。

7.5.2 销毁增强现实会话

接口定义: `void arSessionDestroy(ARSession* session)`

功能描述: 销毁增强现实会话结构。

输入参数:

类型: `ARSession*`, 值: 增强现实会话结构指针。

输出参数: 不做定义。

7.5.3 增强现实会话设置云服务密钥

接口定义: `void arSessionSetKeyAndSecret(ARSession* session, const char* key, const char* secret)`

功能描述: 授权云服务。

输入参数:

类型: `ARSession*`, 值: 增强现实会话结构指针。

类型: `const char*`, 值: Key 字符串。

类型: `const char*`, 值: Secret 字符串。

输出参数: 不做定义。

7.5.4 增强现实会话获取版本号

接口定义: `void arSessionGetVersion(const ARSession* session, char* version, int versionSize)`

功能描述: 获取版本号。

输入参数:

类型: `ARSession*`, 值: 增强现实会话结构指针。

类型: `int`, 值: 版本号字符串数组的长度。

输出参数:

类型: `char*`, 值: 填充版本号的字符串。

7.5.5 设置增强现实会话配置

接口定义: `ARResult arSessionConfigure(ARSession* session, const ARConfig* config)`

功能描述: 设置增强现实会话结构的配置, 返回 `ARRESULT_SUCCESS` 表示成功。

输入参数:

类型: `ARSession*`, 值: 增强现实会话结构指针。

类型: `const ARConfig*`, 值: 配置指针。

输出参数:

类型: ARResult, 值: 返回值, 参考 ARResult 定义。

7.5.6 获取增强现实会话配置

接口定义: void arSessionGetConfig(const ARSession* session, ARConfig* out_config)

功能描述: 获取增强现实会话结构的配置。

输入参数:

类型: const ARSession*, 值: 增强现实会话结构指针。

输出参数:

类型: ARConfig*, 值: 配置指针。

7.5.7 恢复运行增强现实会话

接口定义: ARResult arSessionResume(ARSession* session)

功能描述: 恢复运行增强现实会话, 进入运行状态, 该状态下才会对场景进行跟踪识别, 返回 ARRESULT_SUCCESS 表示成功。

输入参数:

类型: ARSession*, 值: 增强现实会话结构指针。

输出参数:

类型: ARResult, 值: 返回值, 参考 ARResult 定义。

7.5.8 暂停运行增强现实会话

接口定义: ARResult arSessionPause(ARSession* session)

功能描述: 暂停运行增强现实会话, 进入暂停状态, 该状态下停止对物体场景的跟踪识别, 返回 ARRESULT_SUCCESS 表示成功。

输入参数:

类型: ARSession*, 值: 增强现实会话结构指针。

输出参数:

类型: ARResult, 值: 返回值, 参考 ARResult 定义。

7.5.9 更新增强现实会话

接口定义: ARResult arSessionUpdate(ARSession* session, ARFrame* out_frame)

功能描述: 更新增强现实会话的内部数据, 例如更新设备姿态信息, 识别的平面节点、图像节点。该函数需要按照一定的频率进行调用, 使增强现实会话内部信息不断更新, 返回 ARRESULT_SUCCESS 表示成功。

输入参数:

类型: ARSession*, 值: 增强现实会话结构指针。

输出参数:

类型: ARFrame*, 值: 帧画面结构指针, 需要预先创建, 该结构内容会被更新。

类型: ARResult, 值: 返回值, 参考 ARResult 定义。

7.5.10 启动运行增强现实会话算法

接口定义: ARResult arSessionStartAlgorithm (ARSession* session, ARAlgorithmType type, int option)

功能描述：手动方式启动算法，返回 ARRESULT_SUCCESS 表示成功。

输入参数：

类型：ARSession*，值：增强现实会话结构指针。

类型：ARAlgorithmType，值：算法类型，参考 ARAlgorithmType 定义。

类型：int，值：启动算法选项。

输出参数：

类型：ARResult，值：返回值，参考 ARResult 定义。

7.5.11 停止运行增强现实会话算法

接口定义：ARResult arSessionStopAlgorithm(ARSession* session, ARAlgorithmType type, int option)

功能描述：手动方式停止算法，返回 ARRESULT_SUCCESS 表示成功。

输入参数：

类型：ARSession*，值：增强现实会话结构指针。

类型：ARAlgorithmType，值：算法类型，参考 ARAlgorithmType 定义。

类型：int，值：停止算法选项。

输出参数：

类型：ARResult，值：返回值，参考 ARResult 定义。

7.5.12 设置增强现实会话显示区域

接口定义：void arSessionSetDisplayGeometry(ARSession* session, int rotation, int width, int height)

功能描述：设置屏幕显示参数。

输入参数：

类型：ARSession*，值：增强现实会话结构指针。

类型：int，值：手机界面朝向模式。

类型：int，值：显示区域宽的像素数量。

类型：int，值：显示区域高的像素数量。

输出参数：不做定义。

7.5.13 获取增强现实会话地图

接口定义：ARResult arSessionAcquireWorldMap(ARSession* session, ARWorldMap** out_map)

功能描述：获取增强现实会话当前的增强现实世界地图结构。

输入参数：

类型：ARSession*，值：增强现实会话结构指针。

输出参数：

类型：ARWorldMap**，值：创建新的增强现实世界地图结构，结构地址保存在*out_map 中。

类型：ARResult，值：返回值，参考 ARResult 定义。

7.5.14 获取增强现实会话跟踪状态

接口定义：void arSessionGetTrackingState (ARSession* session, ARTrackingState* out_state)

功能描述：获取跟踪状态，不同状态表示内部的算法运行情况，比如：初始化、正常跟踪等。

输入参数:

类型: ARSession*, 值: 增强现实会话结构指针。

输出参数:

类型: ARTrackingState*, 值: 跟踪状态指针, 参考 ARTrackingState 定义。

7.5.15 增强现实会话托管锚点

接口定义: ARResult arSessionHostAnchor(ARSession* session, const ARAnchor* anchor, ARAnchor** out_cloud_anchor)

功能描述: 将本地锚点托管至云端, 创建对应的云锚点结构。

输入参数:

类型: ARSession*, 值: 增强现实会话结构指针。

类型: ARAnchor*, 值: 本地锚点。

输出参数:

类型: ARAnchor**, 值: 创建新的云锚点结构, 结构地址保存在*out_cloud_anchor。

类型: ARResult, 值: 返回值, 参考 ARResult 定义。

7.5.16 增强现实会话解析锚点

接口定义: ARResult arSessionResolveAnchor(ARSession* session, const char* cloud_anchor_id, ARAnchor** out_cloud_anchor)

功能描述: 通过云锚点 ID 解析并获取云锚点。

输入参数:

类型: ARSession*, 值: 增强现实会话结构指针。

类型: const char*, 值: 云端锚点 id。

输出参数:

类型: ARAnchor**, 值: 欣创建的 id 对应的云锚点结构, 结构地址保存在*out_cloud_anchor。

类型: ARResult, 值: 返回值, 参考 ARResult 定义。

7.5.17 设置增强现实会话整型数据类型参数值

接口定义: void arSessionSetIntValue(ARSession* session, ARParameterEnum type, int* value, int size)

功能描述: 设置指定的整型数据类型参数的值。

输入参数:

类型: ARSession*, 值: 增强现实会话结构指针。

类型: ARParameterEnum, 值: 参数枚举, 参考 ARParameterEnum 定义。

类型: int*, 值: 参数对应的整形数组。

类型: int, 值: 整形数组长度。

输出参数: 不做定义。

7.5.18 设置增强现实会话浮点型数据类型参数值

接口定义: void arSessionSetFloatValue(ARSession* session, ARParameterEnum type, float* value, int size)

功能描述: 设置指定的浮点型数据类型参数的值。

输入参数:

类型: ARSession*, 值: 增强现实会话结构指针。

类型: ARParameterEnum, 值: 参数枚举, 参考 ARParameterEnum 定义。

类型: float*, 值: 参数对应的浮点数组。

类型: int, 值: 浮点数组长度。

输出参数: 不做定义。

7.5.19 设置增强现实会话字符串数据类型参数值

接口定义: void arSessionSetStringValue(ARSession* session, ARParameterEnum type, char* value, int size)

功能描述: 设置指定的字符数据类型参数的值。

输入参数:

类型: ARSession*, 值: 增强现实会话结构指针。

类型: ARParameterEnum, 值: 参数枚举, 参考 ARParameterEnum 定义。

类型: char*, 值: 参数对应的字符串数组。

类型: int, 值: 字符串数组长度。

输出参数: 不做定义。

7.5.20 获取增强现实会话整型数据类型参数值

接口定义: void arSessionGetIntValue(const ARSession* session, ARParameterEnum type, int* out_value, int size)

功能描述: 获取指定的整型数据类型参数的值。

输入参数:

类型: const ARSession*, 值: 增强现实会话结构指针。

类型: ARParameterEnum, 值: 参数枚举, 参考 ARParameterEnum 定义。

类型: int, 值: 整型数组长度。

输出参数:

类型: int*, 值: 保存获取结果的整形数组。

7.5.21 获取增强现实会话获取浮点型数据类型参数值

接口定义: void arSessionGetFloatValue(const ARSession* session, ARParameterEnum type, float* out_value, int size)

功能描述: 获取指定的浮点型数据类型参数的值。

输入参数:

类型: const ARSession*, 值: 增强现实会话结构指针。

类型: ARParameterEnum, 值: 参数枚举, 参考 ARParameterEnum 定义。

类型: int, 值: 参数数组长度。

输出参数:

类型: float*, 值: 保存获取结果的浮点数组。

7.5.22 获取增强现实会话字符串数据类型参数值

接口定义: void arSessionGetStringValue(const ARSession* session, ARParameterEnum type, char* out_value, int* out_length, int size)

功能描述: 获取指定的字符数据类型参数的值。

输入参数:

类型: `const ARSession*`, 值: 增强现实会话结构指针。

类型: `ARParameterEnum`, 值: 参数枚举, 参考 `ARParameterEnum` 定义。

类型: `int`, 值: 字符串数组长度。

输出参数:

类型: `char*`, 值: 保存获取结果的字符串数组。

类型: `int*`, 值: 字符串数组被填充的长度。

7.5.23 开始增强现实会话头戴式设备左右眼绘制

接口定义: `ARResult arSessionRenderBeginEye(ARSession* session, AREyeSide eye)`

功能描述: 开始头戴式设备左右眼绘制流程状态。

输入参数:

类型: `ARSession*`, 值: 增强现实会话结构指针。

类型: `AREyeSide`, 值: 左右眼类型, 参考 `AREyeSide` 定义。

输出参数:

类型: `ARResult`, 值: 返回值, 参考 `ARResult` 定义。

7.5.24 结束增强现实会话头戴式设备左右眼绘制

接口定义: `ARResult arSessionEndEye(ARSession* session, AREyeSide eye)`

功能描述: 结束头戴式设备左右眼绘制流程状态。

输入参数:

类型: `ARSession*`, 值: 增强现实会话结构指针。

类型: `AREyeSide`, 值: 左右眼类型, 参考 `AREyeSide` 定义。

输出参数:

类型: `ARResult`, 值: 返回值, 参考 `ARResult` 定义。

7.5.25 提交增强现实会话头戴式设备左右眼绘制结果

接口定义: `ARResult arSessionRenderSubmitFrame(ARSession* session, const int leftEyeTextureId, const int rightEyeTextureId)`

功能描述: 提交左右眼绘制结果给增强现实会话, 实现异步时间扭曲 (ATW) 绘制。

输入参数:

类型: `ARSession*`, 值: 增强现实会话结构指针。

类型: `const int`, 值: 左眼离屏绘制结果纹理参数。

类型: `const int`, 值: 右眼离屏绘制结果纹理参数。

输出参数:

类型: `ARResult`, 值: 返回值, 参考 `ARResult` 定义。

7.5.26 设置增强现实会话头戴式设备绘制窗口环境

接口定义: `ARResult arSessionRenderSetWindow(ARSession* session, void* window)`

功能描述: 设置窗口环境, 使异步时间扭曲 (ATW) 绘制框架能创建所需的绘制环境。

输入参数:

类型: `ARSession*`, 值: 增强现实会话结构指针。

类型: `void*`, 值: 异步时间扭曲 (ATW) 绘制所需的窗口环境。

输出参数:

类型: ARResult, 值: 返回值, 参考 ARResult 定义。

7.6 增强现实世界地图

7.6.1 获取增强现实世界地图中新锚点

接口定义: ARResult arWorldMapAcquireNewAnchor(ARSession* session, ARWorldMap* ar_map, const ARPose* pose, ARAnchor** out_anchor)

功能描述: 使用指定的位姿来创建锚点。

输入参数:

类型: ARSession*, 值: 增强现实会话结构指针。

类型: ARWorldMap*, 值: 增强现实世界地图结构指针。

类型: const ARPose*, 值: 姿态信息。

输出参数:

类型: ARAnchor**, 值: 创建新的锚点结构, 结构地址保存在*out_anchor。

类型: ARResult, 值: 返回值, 参考 ARResult 定义。

7.6.2 获取增强现实世界地图中所有锚点

接口定义: void arWorldMapGetAllAnchors(const ARSession* session, const ARWorldMap* ar_map, ARAnchorList* out_anchor_list)

功能描述: 获取增强现实世界地图结构中所有锚点。

输入参数:

类型: const ARSession*, 值: 增强现实会话结构指针。

类型: const ARWorldMap*, 值: 增强现实世界地图结构指针。

输出参数:

类型: ARAnchorList*, 值: 锚点列表, 需要预先创建锚点列表结构。

7.6.3 获得增强现实世界地图中节点

接口定义: void arWorldMapGetAllNodes(const ARSession* session, const ARWorldMap* ar_map, ARNodeType filter_type, ARNodeList* out_node_list)

功能描述: 获取增强现实世界地图中所有节点, 包括平面、图像、手势节点。ARNodeType 参数用于过滤需要返回的节点类型。

输入参数:

类型: const ARSession*, 值: 增强现实会话结构指针。

类型: const ARWorldMap*, 值: 增强现实世界地图结构指针。

类型: ARNodeType, 值: 指定获取节点的类型, 参考 ARNodeType 定义。

输出参数:

类型: ARNodeList*, 值: 节点列表, 需要预先创建节点列表结构。

7.6.4 释放增强现实世界地图

接口定义: void arWorldMapRelease(ARWorldMap* ar_map)

功能描述: 释放增强现实世界地图。

输入参数:

类型: ARWorldMap*, 值: 增强现实世界地图结构指针。
输出参数: 不做定义。

7.7 帧画面

7.7.1 创建帧画面

接口定义: void arFrameCreate(ARSession* session, ARFrame **out_frame)

功能描述: 创建帧画面结构。

输入参数:

类型: ARSession*, 值: 增强现实会话结构指针。

输出参数:

类型: ARFrame**, 值: 创建新的帧画面结构, 结构地址保存在*out_frame 中。

7.7.2 销毁帧画面

接口定义: void arFrameDestroy(ARFrame* out_frame)

功能描述: 销毁帧画面结构。

输入参数:

类型: ARFrame*, 值: 帧画面结构指针。

输出参数: 不做定义。

7.7.3 判断显示区域是否发生变化

接口定义: void arFrameGetDisplayGeometryChanged(const ARSession* session, const ARFrame* frame, int* ut_geometry_changed)

功能描述: 判断显示区域是否发生了变化。

输入参数:

类型: const ARSession*, 值: 增强现实会话结构指针。

类型: const ARFrame *, 值: 帧画面结构指针。

输出参数:

类型: int*, 值: 内容是否发生变化, 0 表示没有发生变化, 非 0 为发生了变化。

7.7.4 根据设备旋转模式转化纹理坐标

接口定义: void arFrameTransformDisplayUvCoords(const ARSession* session, const ARFrame* frame, int num_elements, const float* uvs_in, float* uvs_out);

功能描述: 根据设备旋转模式, 将纹理坐标转化为对应旋转模式下正确的纹理坐标。

输入参数:

类型: const ARSession*, 值: 增强现实会话结构指针。

类型: const ARFrame*, 值: 帧画面结构指针。

类型: int, 值: 需要转化的浮点数数量。

类型: const float*, 值: 需要转化的纹理坐标数组。

输出参数:

类型: float*, 值: 输出转化后的纹理坐标信息。

7.7.5 设置点求交查询模式

接口定义: `void arFrameSetPointQueryMode(const ARSession* session, const ARFrame* frame, ARQueryMode mode)`

功能描述: 设置求交查询模式。

输入参数:

类型: `const ARSession*`, 值: 增强现实会话结构指针。

类型: `const ARFrame*`, 值: 帧画面结构指针。

类型: `ARQueryMode`, 值: 求交查询模式, 参考 `ARQueryMode` 定义。

输出参数: 不做定义。

7.7.6 获取点求交查询结果列表

接口定义: `void arFramePointQuery(const ARSession* session, const ARFrame* frame, float pixel_x, float pixel_y, ARQueryResultList* query_result_list)`

功能描述: 从从视点到屏幕交点投射光线, 同节点求交, 获得求交查询结果列表。

输入参数:

类型: `const ARSession*`, 值: 增强现实会话结构指针。

类型: `const ARFrame*`, 值: 帧画面结构指针。

类型: `float`, 值: 屏幕点击位置的 X 坐标。

类型: `float`, 值: 屏幕点击位置的 Y 坐标。

输出参数:

类型: `ARQueryResultList*`, 值: 求交获得的查询结果列表, 需要预先创建该结构。

7.7.7 获取线求交查询结果列表

接口定义: `void arFrameRayQuery(const ARSession* session, const ARFrame* frame, const float* ray_origin_3, const float* ray_direction_3, ARQueryResultList* hit_result_list)`

功能描述: 发射任意一条射线, 同节点求交, 获得求交查询结果列表。

输入参数:

类型: `const ARSession*`, 值: 增强现实会话结构指针。

类型: `const ARFrame*`, 值: 帧画面结构指针。

类型: `const float*`, 值: 射线的起点坐标, 包含 3 个浮点数。

类型: `const float*`, 值: 射线的方向向量, 包含 3 个浮点数。

输出参数:

类型: `ARQueryResultList*`, 值: 求交获得的查询结果列表, 需要预先创建该结构。

7.7.8 获取当前帧的光照估计

接口定义: `void arFrameGetIlluminationEstimate(const ARSession* session, const ARFrame* frame, ARIlluminationEstimate* out_light_estimate)`

功能描述: 获取当前帧的光照估计结构。

输入参数:

类型: `const ARSession*`, 值: 增强现实会话结构指针。

类型: `const ARFrame*`, 值: 帧画面结构指针。

输出参数:

类型: `ARIlluminationEstimate*`, 值: 光照估计结构指针, 需要预先创建该结构。

7.7.9 获取当前帧的稀疏点云

接口定义: `ARResult arFrameAcquirePointCloud(const ARSession* session, const ARFrame* frame, ARPointCloud** out_point_cloud)`

功能描述: 创建当前帧新的稀疏点云结构。

输入参数:

类型: `const ARSession*`, 值: 增强现实会话结构指针。

类型: `const ARFrame*`, 值: 帧画面结构指针。

输出参数:

类型: `ARPointCloud**`, 值: 创建新的稀疏点云结构, 结构地址保存在`*out_point_cloud`中。

类型: `ARResult`, 值: 返回值, 参考 `ARResult` 定义。

7.7.10 获取当前帧的新相机

接口定义: `void arFrameAcquireCamera(const ARSession* session, const ARFrame* frame, ARCamera** out_camera)`

功能描述: 创建当前帧新的相机结构。

输入参数:

类型: `const ARSession*`, 值: 增强现实会话结构指针。

类型: `const ARFrame*`, 值: 帧画面结构指针。

输出参数:

类型: `ARCamera**`, 值: 创建新的相机结构, 结构地址保存在`*out_camera`中。

7.7.11 获取当前帧的稠密网格

接口定义: `ARResult arFrameAcquireDenseMesh(const ARSession* session, const ARFrame* frame, ARDenseMesh **out_dense_mesh)`

功能描述: 创建当前帧新的稠密网格结构。

输入参数:

类型: `const ARSession*`, 值: 增强现实会话结构指针。

类型: `const ARFrame*`, 值: 帧画面结构指针。

输出参数:

类型: `ARDenseMesh**`, 值: 创建新的稠密网格结构, 结构地址保存在`*out_desh_mesh`中。

7.7.12 获取更新的锚点列表

接口定义: `void arFrameGetUpdatedAnchors(const ARSession* session, const ARFrame* frame, ARAnchorList* out_anchor_list)`

功能描述: 获得更新的锚点列表, 增强现实会话更新过程中, 部分锚点位置等信息会发生变化。

输入参数:

类型: `const ARSession*`, 值: 增强现实会话结构指针。

类型: `const ARFrame*`, 值: 帧画面结构指针。

输出参数:

类型: `ARAnchorList*`, 值: 锚点列表, 该结构预先创建。

7.7.13 获取更新的节点列表

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/526221141150011005>