解决方案实践

AR 现场作业解决方案

文档版本	1.0
发布日期	2023-06-20

华为技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 方案概述

应用场景

随着国家“出五进一”等用人标准的陆续出台，工业场景中的现场作业面临人员短缺、作业现场专家缺乏等问题。

- 传统的人工巡检劳动强度大、检测质量差、巡检结果多依靠纯手工纸质记录，后期再录入业务系统，全流程数字化程度低；
- 作业现场场地面积大，巡检设备众多，需要记录回传的信息类型多，如遇到巡检异常，作业人员缺乏专业知识，作业现场缺少专家支撑，存在安全隐患；
- 传统线下培训成本高、互动性较弱、经验传递难、知识积累难。
- 专业设备结构复杂，检修操作过程依靠感觉，缺少规划化监控，安全性较难保障。

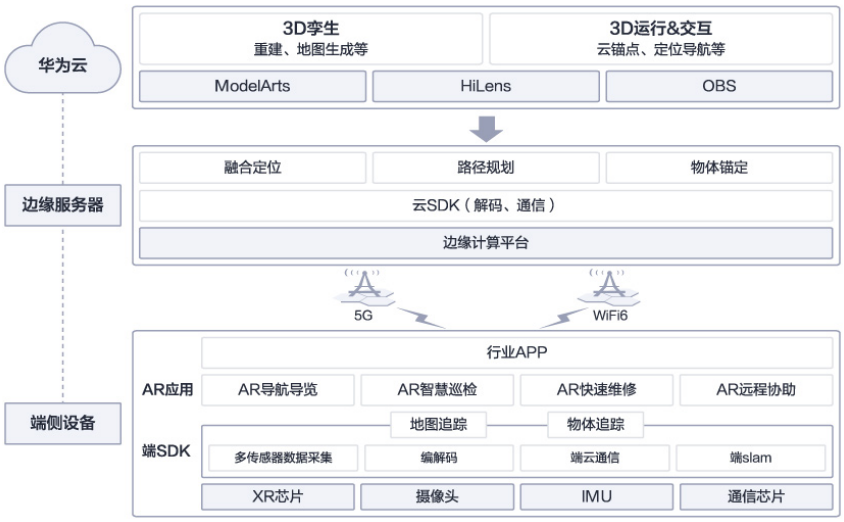
AR现场作业解决方案，联合AR眼镜硬件合伙作伴，可以实现如下业务效果：

- AR智慧巡检，用户穿戴AR设备融入3D数字化信息，完成设备巡检、异常上报、信息录入等工作。设备自动化识别，当用户走到设备附近，设备参数标准参数自动触发显示，和后台数据在视野中实时呈现；3D隐藏结构展示，快速赋能巡检作业新员工；巡检信息后台数据比对，异常数据自动上报，提升巡检效率、降低工作负荷；
- AR导航导览，工业场景下，场站面积大，场景复杂，巡检点位众多，而且有许多相同设备需要巡检。依靠纸质记录、工人经验很容易出现漏检、错检等现象。AR导航导览可彻底解决这一问题：根据工单，所有巡检点位自动编排，巡检路径最优化，提高作业效率；巡检过程中，根据距离感应，周围设备信息自动化呈现，彻底解决设备贴牌、扫码记录的繁琐步骤；通过定位导航技术，工人到达巡检点位自动打卡，并指引到达下一个巡检点，解决漏检、错检问题；
- AR辅助维修，借助AR设备现场作业，用户可在产线中对多个设备，高效率、精准地故障发现，从而快速维修，以减少由于设备故障而带来的生产损失。维修任务可视化呈现，与现实中的设备虚实融合，检修员可直观找到待检修区域及设备；维修步骤3D可视化引导，精准指引用户维修过程，按照眼镜端提示操作，全程自动录像，方便问题回溯和实时查看；工单数据智能化回传后台，减少传统纸质信息记录中的错记、漏记问题，提升设备维修效率；
- AR远程协作，现场作业过程中遇到紧急问题时，作业人员无法第一时间找到解决方法，可通过AR远程协助请求远程联络专家，现场作业状况实时同步，远程专家第一视角作业指导。通过AR标注、屏幕共享，指导现场员工进行作业，快速解决现场问题。多部门、多人协同作业，指定用户知识分享、精准空间内容锚定。

方案架构

该解决方案基于端边云部署架构构建AR现场作业工作流，云端部署三维重建、模型训练等算力消耗型服务，管理和更新边缘服务。用户按需在边缘部署融合定位、物体定位跟踪等AR能力；基于端侧SDK自定义开发AR面板、AR导航、设备拆解等功能。

图 1-1 方案架构图



该解决方案会部署如下资源：

- 创建用于上传训练数据和结果数据导出的OBS桶，开发者可以自主通过华为云账号向该桶上传更新数据和下载结果数据；
- 用户在HiLens技能市场订阅所需的边侧AR服务技能；用户自购边缘服务器，使用HiLens云服务纳管边缘服务器并将技能下发到边缘服务器。

方案优势

- 算法高效高精度

  采用先进的深度学习、AR算法，深入研究各行业业务场景，提高技术在复杂场景中的适配度，支持多终端视觉定位导航，高效高精度大场景重建，厘米级定位精度、毫秒级位姿输出。

- 端边云协同架构

  端边云协同架构支持端侧使用到最新最优的云上AI服务，边侧算力充足，支持快速部署充分发挥AI算法优势，缓解低算力终端计算压力。

- 适配复杂场景

  该解决方案考虑了多种复杂的环境，如场站中大量重复结构、相似设备，支持弱纹理设备的精准识别及定位跟踪，帮助用户在复杂场景中高效率完成现场作业。

- 提供通用API接口

  提供符合RESTful规范的API访问接口，不同行业不同应用场景用户，均可通过接口直接调用场景重建等云服务，接口通用性强，对接快速、简单易用。

约束与限制

- 该方案当前仅支持华北-北京四区域。在开始之前，请确认您已经拥有一个可以访问该区域的华为云账号，并完成实名认证，帐号不能处于欠费或冻结状态。

- 边缘服务服务器：系统：ubuntu18.04，不修改内核，显卡3080、3090（支持cuda11.1），安装新版的nvidia驱动（如：470.63.01）

2 资源和成本规划

表 2-1 资源和成本规划

华为云服务	计费模式	数量	价格	备注
场景重建	一次性收费	N/A	N/A	5万平方米以内，收费XX万；超过5万平后，每增加1万平方米，收费X万； 包含点云地图采集、视觉地图制作、导航地图制作
大空间视觉定位服务	按年收费	1	XX万元	1年内，服务10并发以内，收费XX万，之后每增加1并发，收费X万
物体锚定服务	按年收费	1	XX万元	10个以内物体，1年，服务10并发，收费XX万；之后每增加1并发，收费X万；每增加10个物体，收费X万； （不包含物体图像数据采集）
路径规划与导航服务	按年收费	1	XX万元	10万平米以内，1年内，服务10并发以内，收费XX万，之后每增加10并发，收费X万

3 准备工作

- **注册华为帐号**并开通华为云，完成**实名认证**。
- 使用华为帐号登录**HiLens控制台**，选择专业版，选择“北京四”区域。
- 技能订阅：用户提供北京四的project_id和iam_id，用于共享定制技能。提供账号信息后，等待技能提供方共享技能；共享成功后，用户在HiLens技能市场购买相关的技能（如：AR巡检演示Demo需要skill-xr3d-3ddetection和skill-xr3d-edge-server两个技能）
- 边缘服务服器：系统：ubuntu18.04，不修改内核，显卡3080、3090（支持cuda11.1），安装新版的nvidia驱动（如：470.63.01）

4 实施步骤

4.1 边缘服务部署

4.1.1 边缘服务器安装 Docker

1. docker

   查看docker版本，版本建议20以上

   ```
   sudo docker --version
   ```

   安装教程

   ```
   sudo apt update
   sudo apt install docker-ce docker-ce-cli containerd.io
   ```

2. nvidia-docker

   查看nvidia-docker版本

   ```
   sudo nvidia-docker --version
   ```

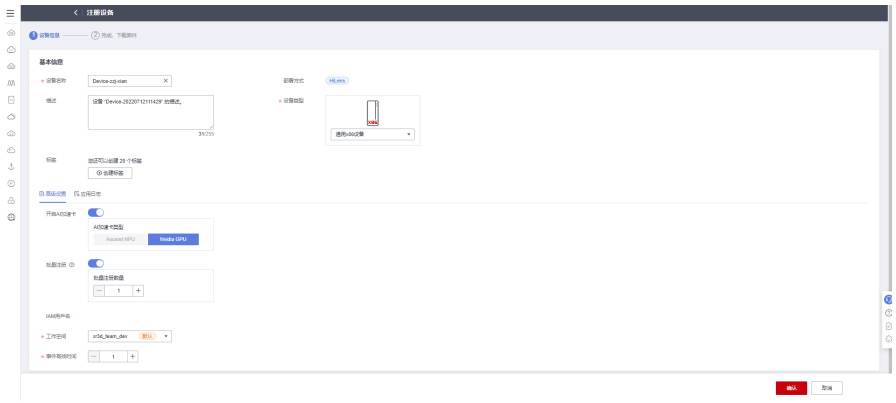
   安装教程参考：**https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/install-guide.html#install-guide**

   ```
   distribution=$(. /etc/os-release;echo $ID$VERSION_ID) \
   && curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg \
   && curl -s -L https://nvidia.github.io/libnvidia-container/experimental/$distribution/libnvidia-container.list | \
   sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg] https://#g' | \
   sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list
   sudo apt-get update
   sudo apt-get install -y nvidia-docker2
   sudo systemctl restart docker
   ```

4.1.2 HiLens 注册设备

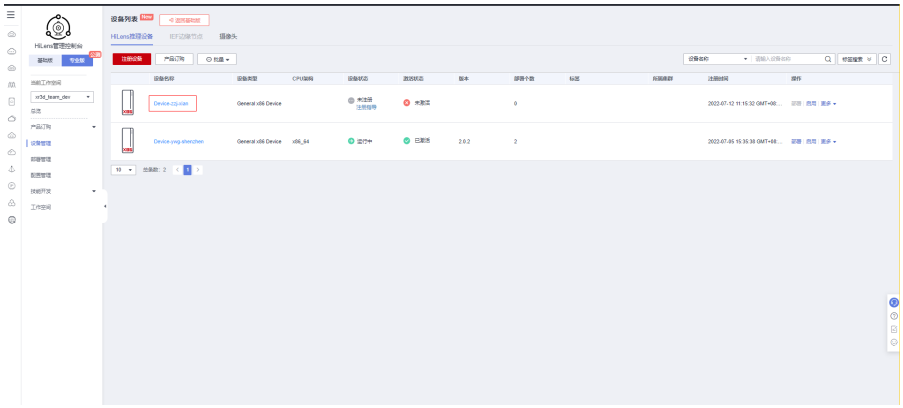
1. 注册页面

图 4-1 注册



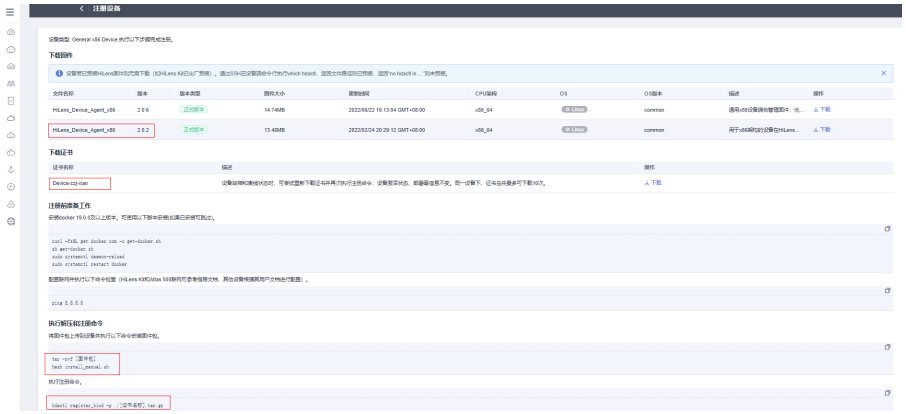
2. 注册成功页面（未激活）

图 4-2 注册成功



3. 下载固件和证书

图 4-3 下载



4. 安装固件包

   安装新版本固件（旧版本固件存在宿主机重启技能丢失，无法再次拉起本地容器的情况）

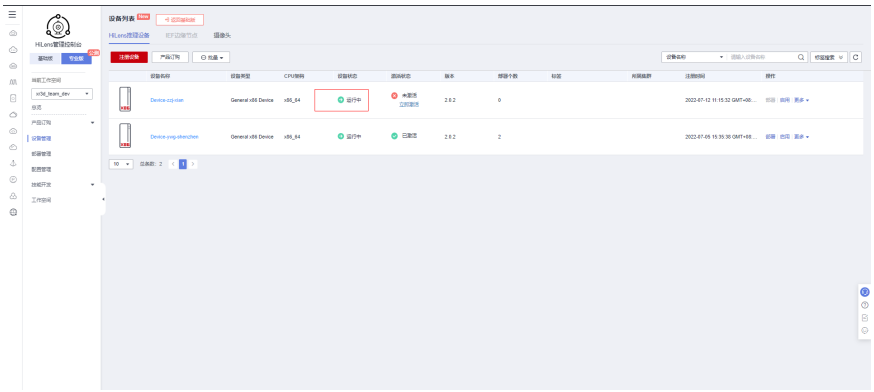
```
tar -xvf [固件包]
bash install_manual.sh
ps -ef | grep hdad      # 查看hdad进程是否存在
```

5. 注册

```
hdactl register_bind -p [证书名称].tar.gz --ip [docker0 ip]
```

注册成功的状态：

图 4-4 状态



说明

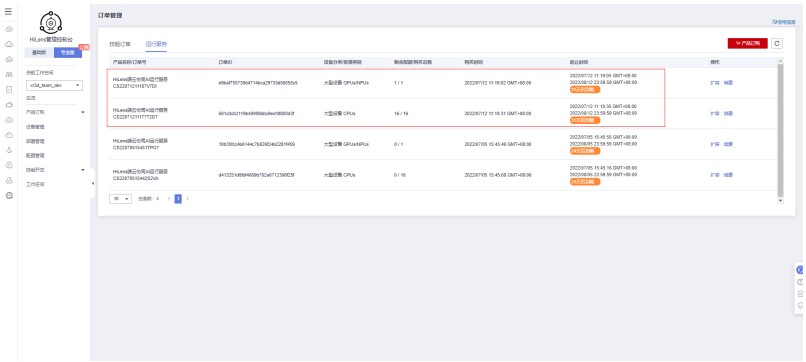
容器内的Agent与宿主机中的固件交互，缺省情况下会把宿主机的IP挂载到容器内，如果宿主机的IP是动态获取的，会导致容器内的Agent与宿主机中的固件无法交互；需要使用“ifconfig”命令查看docker0的ip（172.17.0.1）

4.1.3 HiLens 激活设备

1. 购买运行服务

基于边缘设备的规格购买对应的CPU和GPU，分别按照CPU资源和GPU资源类型下两个订单

图 4-5 购买运行服务



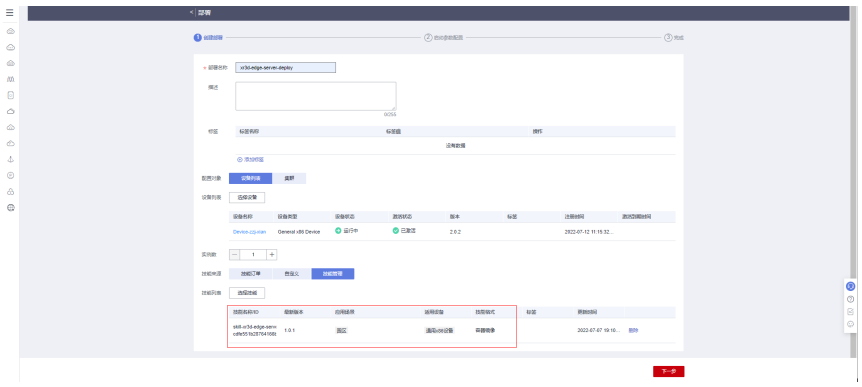
2. 激活

点击“立即激活”按钮，进入设备激活页面，选择对应的“CPU订单”和“GPU订单”，点击确认即可激活。

4.1.4 HiLens 部署技能

4.1.4.1 部署跟踪服务

图 4-6 部署跟踪服务



1. 计算资源配置
   按需配置，推荐内存8G以上，加速卡缺省1个（暂时无法精确到小数）

图 4-7 计算资源配置

2. 环境变量配置
   新版本不需要配置DETECTION_3D_ENDPOINT，可自动从HiLens配置的环境变量中获取

表 4-1 环境变量配置

环境变量名称	示例	描述
IS_SGLE_INSTANCE	True	部署模式，单实例版本需要配置IS_SINGLE_INSTANCE参数；多实例版本不需要配置该参数

环境变量名称	示例	描述
MODELS_CONFIG	{"models":{"0":{"object_key": "models/shuifo/shuifo.bin","bucket_name": "xr3d-models-bucket"}},"maps":{"0":{"object_key": "maps/F1-2F/F1-2F.bin","bucket_name": "xr3d-models-bucket"}}}	配置模型文件和地图文件在OBS桶的位置
DEFAULT_OBJ_NAME	ShuShuiFa	缺省物体名称
LOG_LEVEL	3	日志级别
DETECTION_3D_PORT	8080	定位服务端口

3. 网络配置

   单实例网络配置

   图 4-8 计算资源配置

4. 数据存储配置

   a. 宿主机新建目录并改变权限

   sudo mkdir -p /opt/xr3d/models/

   图 4-9 改变权限

```
ei-server@EI-Server-PC:/opt$
ei-server@EI-Server-PC:/opt$
ei-server@EI-Server-PC:/opt$
ei-server@EI-Server-PC:/opt$ sudo mkdir -p /opt/xr3d/models/
ei-server@EI-Server-PC:/opt$ sudo mkdir -p /opt/xr3d/ai_models/
ei-server@EI-Server-PC:/opt$ sudo mkdir -p /opt/xr3d/obj_models/
ei-server@EI-Server-PC:/opt$
ei-server@EI-Server-PC:/opt$ sudo chmod 777 -R /opt/xr3d/
ei-server@EI-Server-PC:/opt$
ei-server@EI-Server-PC:/opt$
```

   表 4-2 数据存储配置

挂载地址	说明	备注
/opt/xr3d/models	3ddetection技能的物体模型文件和ai模型文件	-

挂载地址	说明	备注
/opt/xr3d/ai_models	edge_server技能的地图文件	-
/opt/xr3d/ai_models	edge_server技能的物体模型文件	-

b. 在线部署

部署时，通过配置从OBS下载对应的地图和模型文件（需要确认OBS桶中有对应的模型文件）

c. 离线运行

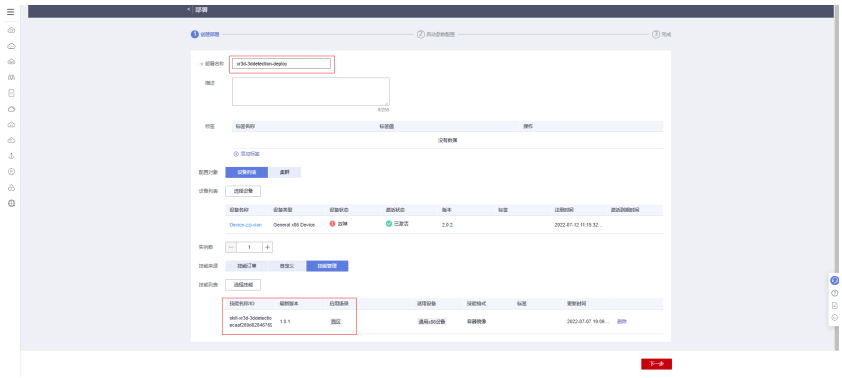
离线状态下无法从OBS通获取模型文件，可以将地图或模型文件拷贝到对应的挂载路径

d. 离线重启电脑

离线重启电脑，技能无法从OBS下载模型文件，可以直接使用挂载路径已有的模型文件

4.1.4.2 部署物体定位服务

图 4-10 部署物体定位服务



1. 计算资源配置

按需配置，推荐内存4G以上，加速卡缺省1个（暂时无法精确到小数）

图 4-11 部署物体定位服务

2. 环境变量配置

图 4-12 部署物体定位服务

表 4-3 环境变量配置

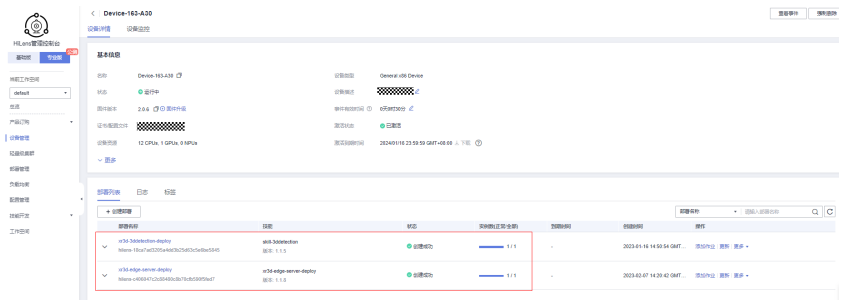
名称	示例	描述
MODELS_CONFIG	{"models":{"0": {"object_key": "models/shuifo/shuifo.bin","bucket_name": "xr3d-models-bucket"}}}	配置模型文件（包含对应的AI模型）在OBS桶的位置

3. 网络配置

图 4-13 网络配置

登入华为云HiLens页面，在设备管理查看对应设备部署服务的状态，跟踪服务与定位服务状态为成功即可。

图 4-14 查看



4.2 端侧 SDK 接入

4.2.1 开发准备

4.2.1.1 开发环境

操作系统：

Windows 10

Unity开发环境：

1. Unity版本：2020.1.17
2. Android SDK版本：API Level 31
3. Android JDK版本：JDK 1.8
4. Android NDK版本：NDK-r19
5. Android Gradle版本：gradle-6.7.1

Unity工程设置（终端为眼镜需设置，PAD不涉及）：

1. 删除原相机，点击菜单栏的“ARCamera/Create Camera For ARGlass”创建AR相机。
2. 设置分辨率：将分辨率调整为2560*720

UI设置：

Canvas：

Rec Transform：Width:1280, Heigh:720

Render Mode：World Space

Event Camera：None(Camera)

ARInspect SDK软件包：

ARInspect for unity_{version}.unitypackage

说明

SDK软件包，由华为方提供。

4.2.1.2 运行环境

设备要求：

终端设备（眼镜/PAD），云/边侧服务器（按需选择）

配置文件：

ARInspect SDK初始配置文件，包括：

ARInspect.config，Unity相关配置

slam_sdk_config.json，SLAM算法适配层相关配置

initParam.json，SLAM算法相关配置

hwServerUrl.json，Android层配置

initParam.schema.json，SLAM算法相关参数格式

说明

默认配置文件跟随ARInspect SDK软件包一起提供，配置文件中关键配置项将在4.2.3.4中介绍。

4.2.2 接口说明

4.2.2.1 接口总览

表 4-4 接口总览

接口	说明
初始化	完成SDK准备工作
获取定位信息	获取端SLAM的定位信息
模型识别跟踪	获取3Dtracking结果
获取当前帧图像	获取一帧摄像头采集的实时图像
注册状态回调	通过此接口接收运行时状态与错误码
销毁	停止ARInspect SDK，释放资源

4.2.2.2 初始化

定义：bool InitSdk(int mode)

描述：初始化SDK以及必要组件。

参数：

表 4-5 参数

名称	类型	描述
mode	int	初始化模式。 1：表示运行端侧SLAM 2：表示运行云侧SLAM 3：表示运行端侧SLAM +云侧TRACK 4：表示运行云侧SLAM+云侧TRACK

返回值：

true：初始化成功

false：初始化失败，参数错误。

4.2.2.3 获取定位信息

定义：Pose GetCameraPose (long time)

描述：获取未来时间SLAM的定位信息，time为0时表示不带预测，获取当前时间SLAM的定位信息。当此值能够正常获取数据时，说明已经完成定位，可进行后续的接口调用。

参数：

表 4-6 参数

名称	类型	描述
time	long	目标时间与当前时间的差值 (单位为纳秒)，为0时为不带预测。 例如：想得到的目标位姿的时间是20点30分40秒6000ns，当前时间为20点30分40秒5000ns，那么time的值为1000。即得到time纳秒后的位姿。

返回值：

Pose：返回对应时间的6dof 位姿数据。

4.2.2.4 模型识别跟踪

定义：void StartTrack (TrackingCallBack<ModelData> callback)

描述：获取3Dtracking结果。上层实现回调处理，SDK通过回调返回识别跟踪结果，callback在程序工作期间被连续回调，更新最新结果，注意当跟踪失败时，需要重新调用识别方法。

参数：

表 4-7 参数

名称	类型	描述
callback	TrackingCallback	实现回调函数，接收、处理、使用识别跟踪结果，callback与ModelDate定义如下

callback定义为：

```
public delegate void TrackingCallback(ModelData modelData);
```

ModelData定义为：

```
public class ModelData
{
public bool hasTracked; //识别跟踪结果，true:成功；false:失败
public string trackMsg; //识别跟踪信息
public string modelName; //模型名称
public string url; //模型下载链接，暂不支持
public Pose modelPose; //6dof位姿
public long time; //时间戳
}
```

返回值：NA

4.2.2.5 获取当前帧图像

定义：Void GetPictureData(SavePictureCallback callback)

描述：在ARInspect SDK工作期间，摄像头被占用，如果想获取摄像头采集的图像，可调用该接口获取一帧实时图像。

参数：

表 4-8 参数

名称	类型	描述
callback	SavePictureCallback	实现回调函数，接收图片数据。callback定义如下

callback定义：

public delegate void SavePictureCallback(byte[] data);

返回值：NA

4.2.2.6 注册状态回调

定义：Void RegisterStateCallback(StateCallback callback)

描述：在ARInspect SDK工作期间，通过该接口实时获取SDK状态，主要包括运行状态与错误码信息，在注册时立即回调一次当前状态，后续由事件触发。

参数：

表 4-9

名称	类型	描述
callback	StateCallback	实现回调函数，处理状态与错误码信息。callback定义如下

callback定义：

public delegate void StateCallback (int stateCode);

stateCode定义

1. 端SLAM定位中。表示端SLAM定位成功，GetCameraPose返回的数据来自端SLAM。
2. 云SLAM定位中。表示云Slam定位成功，GetCameraPose返回的数据来自云SLAM。

1：未找到License

2：License校验失败

11：SD卡未挂载

12：SD卡空间不足

21：网络未连接

22：IP配置错误

23：服务器不可达

24：端口配置错误

31：Unity配置错误

32：帧率配置错误

返回值：NA

4.2.2.7 注销

定义：void StopSdk()

描述：停止ARInspect SDK，资源销毁释放。

参数：NA

返回值：NA

4.2.3 应用开发

4.2.3.1 新建工程或使用已有工程导入 SDK

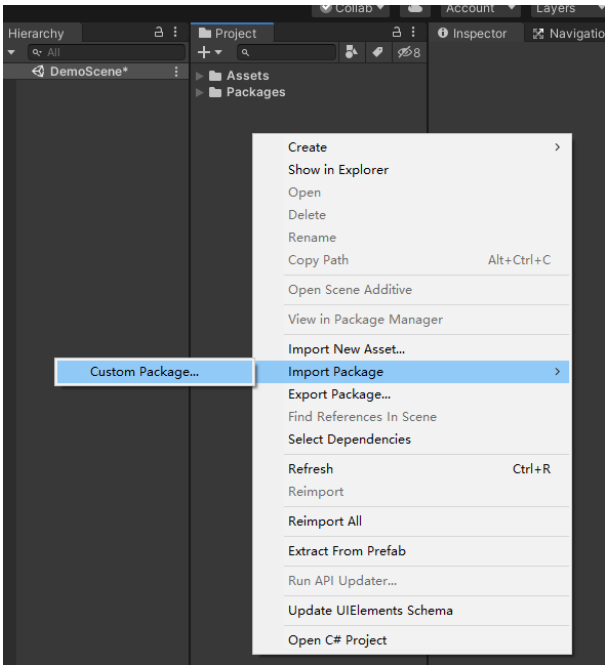
1. 打开Unity开发软件，创建工程：在Unity中创建3d类型的工程。

   ⚠ 注意

   项目路径中禁止中文、特殊字符、空格等。

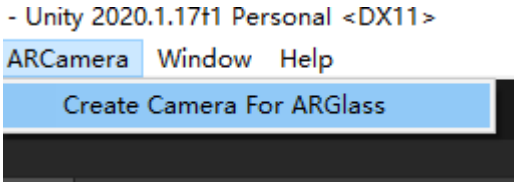
2. 导入ARInspect SDK：在Project面板，右键选择“Improt Package” ->“Custom Package…”，然后在弹出来的窗口中选择相应的SDK进行导入。

图 4-15 导入



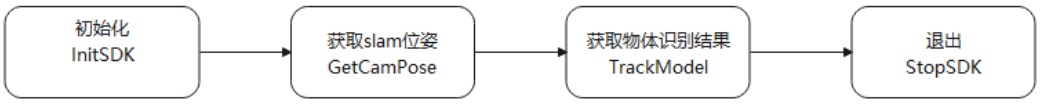
3. 创建新的Camera：删除原工程中的camera，然后在Unity项目中选择”ARCamera” ->“Create Camera For ARGlass” 或者 “Create Camera For ARPad”。

图 4-16 创建新的 Camera



4.2.3.2 SDK 接口使用说明

图 4-17 SDK 接口使用说明



1. 添加引用：” ARInspect SDK”。
2. 在程序生命周期开始时调用SDK初始化方法，对SDK进行初始化：“InspectSDK.Instance.InitSdk(int mode)”。
3. 在Unity中的Update系列方法中进行SLAM数据的调用及获取：“InspectSDK.Instance.GetCameraPose(null)”。
4. 在可以获取到正常位姿后，可以调用物体识别接口：“InspectSDK.Instance.TrackModel(callback)”。
5. 在程序调用前需调用关闭SDK接口：” InspctSDK.Instance.StopSdk()”。
6. 整体代码示例如下图：

图 4-18 代码示例

```
using ARInspectSDK;
using UnityEngine;

public class DemoScript : MonoBehaviour
{
    public GameObject TrackModel;
    private bool IsCalled = false;
    private TrackingCallBack callback;//物体识别回调  按需调用

    private void OnEnable()
    {
        InspctSDK.Instance.InitSDK();
        callback = new TrackingCallBack(CallBackResult);
    }

    private void CallBackResult(ModelData modelData)
    {
        if (modelData.TrackResult)
        {
            TrackModel.transform.position = modelData.ModelPose.position;
            TrackModel.transform.rotation = modelData.ModelPose.rotation;
        }
        else
        {
            InspctSDK.Instance.TrackModel(callback, TrackModel.name);
        }
    }

    private void FixedUpdate()
    {
        Pose pos = InspctSDK.Instance.GetCamPose(null);//空间定位  世界坐标系下的位姿
        if (pos.position != Vector3.zero && !IsCalled)
        {
            IsCalled = true;
            InspctSDK.Instance.TrackModel(callback, TrackModel.name);//物体识别  按需调用
        }
    }

    private void OnDestroy()
    {
        InspctSDK.Instance.StopSDK();
    }
}
```

4.2.3.3 旧版本文件清除

仅当SDK更新时，其中更新说明中特别注明需要清除旧文件，才需要进行此步骤。

1. 删除Unity工程下的“/Assets/plugins”文件夹和“/Assets/Editor”文件夹。
2. 将新版本SDK导入到Unity中。

4.2.3.4 配置文件

配置文件中常用参数描述如下

- ARInspect.config：
  a. "slam_client"：使用端SLAM还是云SLAM。
  b. "save_log": 是否保存SDK日志。调试阶段建议为true，否则false。
- slam_sdk_config.json：
  a. "save_image": 是否保存传给端侧SLAM算法模块的图像。建议false，否则会导致log过大，快速占满磁盘。
  b. "log_level"：日志等级，0=trace, 1=debug, 2=info，3=warn，4=error，5=critical，6=off，建议选择info。
  c. " flush_level "：日志写到文件的触发等级，如4表示遇到error，则将日志写到文件，建议选择3（warn）
- initParam.json：

a. "offlineMapName"：配置地图名称，如："F1-2F.bin"

b. " logLevelSwitch"：日志等级开关，0表示关，1表示开。建议调试阶段开启info及以上即[0, 0, 0, 1, 1, 1]，否则全0

c. " logModuleSwitch"：模块日志开关，0表示关，1表示开。建议调试阶段开启所有模块[1, 1, 1, 1, 1, 1]，否则全0

- hwServerUrl.json：

a. "serverIp": 配置边侧服务器ip地址，如"192.168.3.23"。

b. "vio_type": 配置VIO类型。常用值：1=云SLAM+云TRACK；4=端SLAM+云TRACK。通常有地图选择1，无地图选择4。

c. " log_level ": 日志等级，取值：0~4，对应debug，info，warn，error，fatal，建议调试阶段选择0(debug)，否则为1(info)

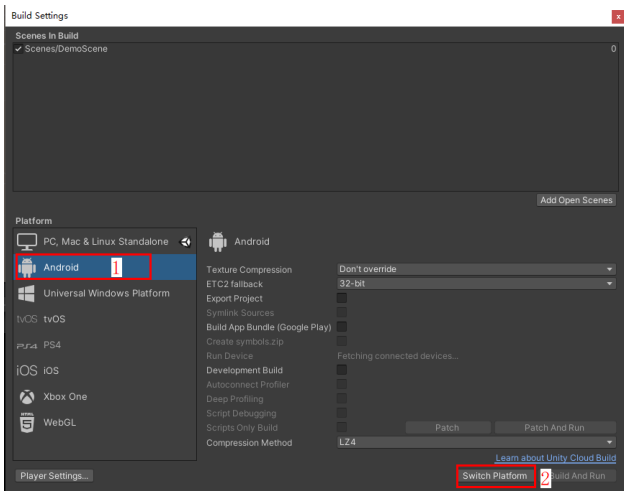
4.2.3.5 构建打包

打包的设置，相关路径，产出文件，文件使用方法描述。

1. 平台设置

   将打包平台设置为Android，等待Unity重新编译完成。

   图 4-19 平台设置

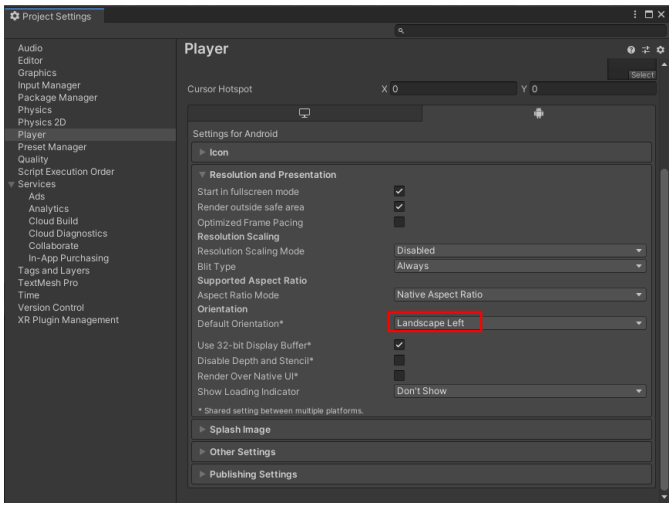


2. 打包设置

   PlayerSettings设置：

   - **Resolution and Presentation**：Default Orientation 选择 Landscape Left。

图 4-20 打包设置



- **Other Settings:**

  Auto Graphic API 勾选。

  Minimum API Level 选择Android 9.0 ‘Pie’(API Level 28)。

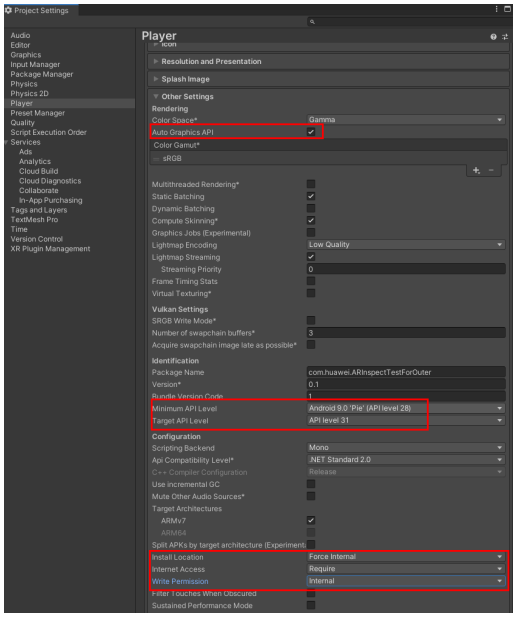
  Target API Level 选择API level 31。

  Install Location 选择 Force Internal。

  Internet Access 选择 Require。

  Write Permission 选择 Internal。

图 4-21 其他设置



- **Publishing Settings:**

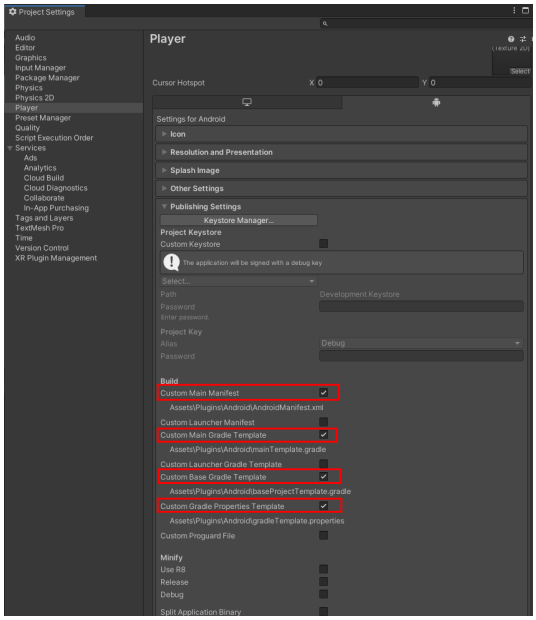
  Custom Main Manifest 勾选。

  Custom Main Gradle Template 勾选。

  Custom Base Gradle Template 勾选。

  Custom Gradle Properties Template 勾选。

图 4-22 发布设置

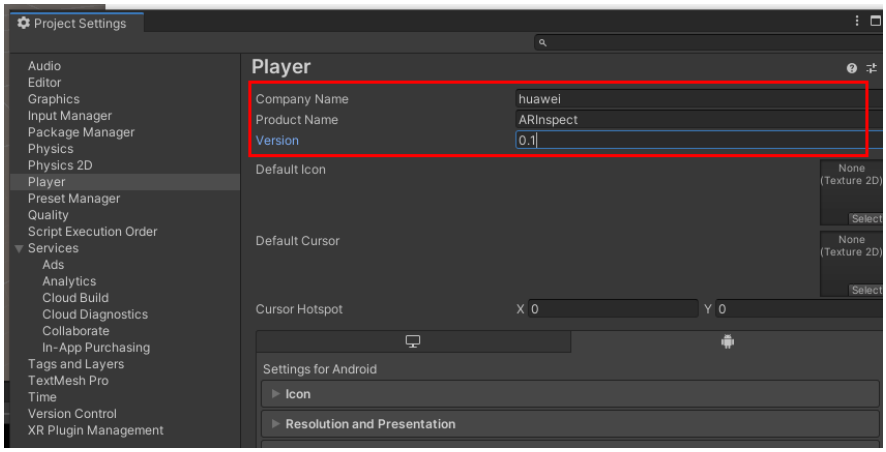


3. 包名设置

   “PlayerSettings > Player”中分别设置“Company Name”，“Product Name”，“Version”。

   “PlayerSettings > Other Settings”中更改“Package Name”。

图 4-23 包名设置

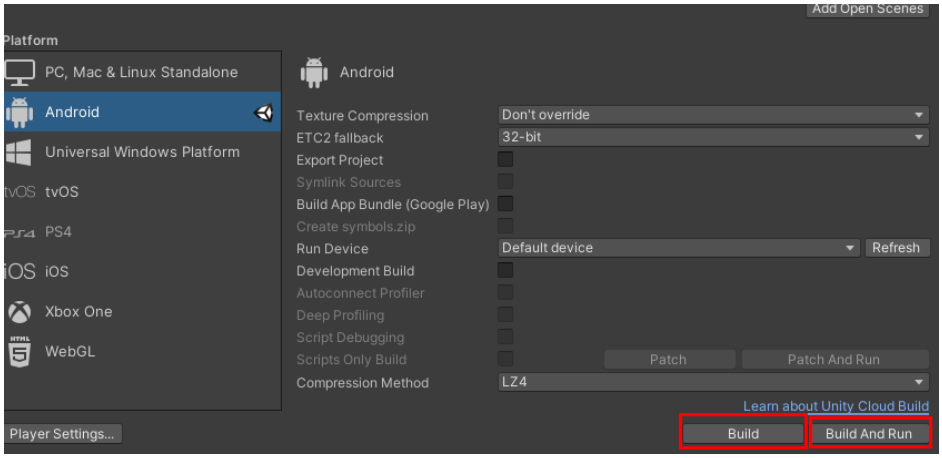


4. 打包及运行

   点击“Build Settings > Build”可以打出apk文件。

   点击“Build Settings > Build And Run”可以打出apk文件并且直接安装到设备中运行。

图 4-24 打包及运行



4.2.4 常见问题

汇总常见问题，持续更新补充。

1. 无法与服务器建立连接

   检查端侧与云侧是否能够ping通，如果能ping通则检查hwServerUrl.json中的ip地址以及端口是否正确配置。

   图 4-25 图示 1

   ```
   {
       "serverIp": "192.168.43.33",
       "tcpPort": 8903,
       "videoPort": 8904,
       "udpPort": 8905,
       "ntpPort": 123,
   ```

   如果无法ping通则检查网络、路由器等具体设置。

2. 服务端出现OOM

   在包含地图的场景中，服务端对内存空间要求较高，建议调整到8G内存。

5 修订记录

表 5-1

发布日期	修订记录
2023-02-03	第一次正式发布。