doi:10.16104/j.issn.1673-1891.2021.01.017

基于增强现实和神经网络的校园展示应用设计

杨　路[1,2],祝天禹[3]

(1.安徽公安职业学院信息网络安全监察系,安徽 合肥 230031;2.南京工业大学机械与动力工程学院,江苏 南京 211816;3.凤台县公安局,安徽 凤台 232100)

摘　要:当前的校园展示多采用虚拟现实开发,该技术让用户完全沉浸入虚拟世界,但忽略了现实世界的信息。增强现实技术能够在真实世界的基础上叠加虚拟信息,进一步增强人们的感知体验。基于 Unity3D+ARCore+TensorFlow 设计了带平面检测、多图像和物体识别功能的增强现实应用,详细描述了虚拟对象、触屏交互、平面检测、图片和物体识别的实现过程和注意问题,说明了在脚本和 UI 设计中的性能优化方法。测试表明多标识物识别提升了应用的实用性,物体识别功能丰富了应用内涵,增加了校园展示的设计方法,拓展了增强现实技术的应用范围。

关键词:增强现实;神经网络;Unity3D;ARCore;TensorFlow

中图分类号:TP391.9;TP183　　**文献标志码**:A　　**文章编号**:1673-1891(2021)01-0081-07

Design of Campus Display Application Based on Augmented Reality and Neural Network

YANG Lu[1,2], ZHU Tianyu[3]

(1. Department of Information Network Security Control, Anhui Public Security College, Hefei, Anhui 230031, China; 2. School of Mechanical and Power Engineering, Nanjing Tech University, Nanjing, Jiangsu 211816, China; 3. Public Security Bureau of Fengtai County, Fengtai, Anhui 232100, China)

Abstract: Currently most of the campus displays are developed through virtual reality technology, which gets users immersed in the virtual world, but ignore information of the real world. In this paper, a campus display application is designed based on Unity3D+ARCore+TensorFlow, which provide functions of plane detection, multi-image and object recognition. The implementation process and problems of virtual objects, touch screen interaction, plane detection, image and object recognition are described in detail. Performance optimization methods in scripting and UI design are illustrated. The test shows that multi-marker recognition improves practicability, and object recognition enriches application implications, which together could increase the design methods of campus displays and expand the application scope of augmented reality technology.

Keywords: augmented reality; neural network; unity3D; ARCore; Tensor Flow

0 引言

高校是代表社会进步的重要场所,要有先进的技术支持其建设。伴随着虚拟现实(Virtual Reality,简称 VR)的发展,基于该技术设计的虚拟校园展示层出不穷,使得用户能够在一个纯粹的虚拟三维空间中对校园进行审视[1-4],但同时忽略了现实世界的信息。增强现实(Augmented Reality,简称 AR)是近年来随着 VR 技术的发展而产生的一项新技术,它在现实世界的基础上叠加虚拟信息,构建出一个信息增强的虚实融合混合体并被用户的感官感知。AR 技术连接起了真实与虚拟两个世界,进一步增强用户的感知体验,最终达到超越现实的体验。目前 AR 技术的应用主要在教育教学领域[5],在校园展示领域的应用研究相对欠缺[6-8]。文献[6]仅做了理论性综述,文献[7]研究了基于 Vuforia 的校园图片识别应用,文献[8]利用 GPS 和深度学习技术研究了校园定位技术。本文采用另一种增强现实

收稿日期:2020-01-17
基金项目:安徽省自然科学重点项目(KJ2019A1210);安徽公安职业学院校内重大项目(XN2018ZDA05)。
作者简介:杨路(1985—),女,安徽怀远人,讲师,博士研究生,研究方向:计算机应用技术。

技术 ARCore,辅以 TensorFlow 设计了具备平面检测、图片和物体识别的增强现实应用。丰富了校园展示的设计方法,拓展了 AR 技术的应用范围,具有一定的理论与实践意义。

1 设计思路

应用设计包括 4 部分[9-10]:(1)虚拟对象建立,包括三维模型,文本、图片等。(2)触屏交互功能,用户能够以手姿的方式与虚拟对象进行交互获得相关信息。(3)标识物识别,本文检测两种标识物:图片和平面,查找到标识物后,把虚拟对象与真实场景画面进行匹配并叠加显示。(4)物体识别,采用神经网络模型和训练集识别并标注摄像头图像上的物体。基于 AR 技术的校园展示应用设计完成后导出生成软件安装包,用户下载并安装软件包生成应用。应用运行后开启摄像头,扫描到标识物显示设计好的虚拟对象,用户可以通过手姿与虚拟对象交互,物体识别能够帮助用户更好地了解校园结构。

2 开发环境和设计流程

2.1 软硬件环境

采用 Unity3D+ARCore+TensorFlow 开发校园展示的增强现实应用,Unity3D 是 IDE 开发环境,用于设计虚拟对象。常用的增强现实开发包有 Vuforia、ARKit 和 ARCore,Vuforia 支持 2D/3D 识别,扫描真实物体进行识别、虚拟按键,尤其是使用 Vuforia 检测图片,但考虑到图片标识物在校园并不是随处可见,增加平面作为第二种标识物。Vuforia 的 Ground plane 可用于平面检测,但在测试中发现其功能不稳定,漂移和抖动现象时常出现,且运行时设备的发热量较高。ARKit 是苹果环境下的开发包,不支持安卓平台。ARCore 是 Google 开发的用于创建 AR 的 SDK,可进行位置跟踪、标识关键点并构建虚实融合的世界,还可以实现平面检测。融合 TensorFlow 提供物体识别功能,辅以 C#编写的脚本,提供了平面检测、图片识别和物体检测功能,提高了应用的实用性。

2.2 设计流程

如图 1 所示:(1)以官方提供的 AugmentedImage 图片识别项目为原型开发,导入 TensorFlow 插件。项目已导入 arcore-unity-sdk 插件包,添加了 ARCore Device 和 Environment Light。前者负责与设备相关的操作(更新检测的平面,设备位置等),后者负责光照相关事宜。在 Assets 创建 scenes(场景)、scripts(脚本)、prefabs(预设件)等文件夹。(2)在 Unity3D 内建立平面检测和图片识别需要的虚拟对象,制作 prefab,编写触屏交互脚本并挂载到相关对象上。(3)平面检测,制作平面 prefab,创建平面生成器和平面控制器。前者用于平面生成,后者对项目的各种情况进行诊断和控制。编写并挂载生成器和控制器脚本,把 ARCore Device 的 First Person Camera 和要显示的虚拟对象 prefab 赋给控制器脚本对应的参数,测试效果。(4)图片识别,导入要识别的图片,创建图片数据库,修改 sessionconfig,编写多图片识别的控制器脚本并挂载测试效果。(5)物体识别,编写摄像头和识别脚本,使用移动设备优化模型集和精简标签集训练,把脚本挂载到项目场景摄像机上测试效果,完成应用开发。

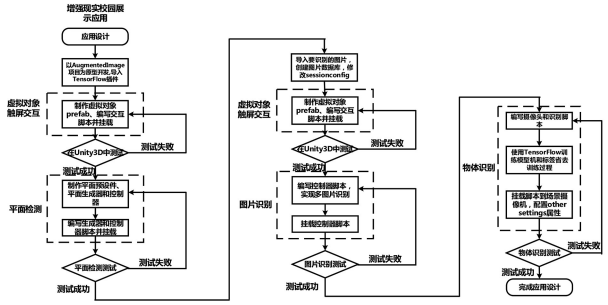

图 1 设计流程

3 应用实现

3.1 虚拟对象

应用涉及的虚拟对象主要包括三维模型、显示图片的 Image、显示文本信息的 Text 和显示按钮的 Button 等。后三者由 UGUI 提供,这是 Unity3D 内置的用于虚拟对象开发的系统,包含 Canvas 和 EventSystem 两个基本组件。其中要显示的图片需要先设置为 Sprite 类型,因为 Image 只支持该类型显示。三维模型自定义设计。测试时发现当导入.fbx类型的模型时可能会出现没有贴图的问题,解决方法如下:选中模型,在 Materials -> Location 选中 Use External Materials,但是模型显示依然有问题。因为之前的操作仅仅是把贴图导入了项目,但还没有正确的贴到模型身上。新建一个 Material,在 Main Maps 的 Albedo 和 Normal Map 选择正确的贴图,再把 Material 拖到模型上即可。

3.2 触屏交互

Unity 的 INPUT 类提供了包含手机移动设备在内的各项系统事件功能,本文基于 INPUT 类实现模型的触屏移动缩放功能的 movescale 脚本,流程如图 2 所示:首先判断触屏点个数,如果为 0 表示没有触屏动作,脚本结束,如果不为 0 则进行下一步判定。

如果触屏个数为 1 表示单点触摸,调用 GetTouch(0)获取单点信息,如果触点的 phase 是 TouchPhase.Began 表明开始单点触摸移动,记录起始位置,如果触点的 phase 是 TouchPhase.Ended 表明移动操作结束,记录当前的位置,分别计算当前和起始位置在空间上的差值,调用 Translate 方法对物体进行移动。如果触屏个数为 2 则进行两点触摸的放大缩小操作,更新上一次的触点信息,计算并比较两次触点间的距离差,为正值则执行放大操作,反之执行缩小操作,操作最后要更新最新的触摸点信息以备下次使用。值得注意的是,触屏操作和模型缩放都存在边界限制,所以在更新 localScale 前要做判定:对放大操作,如果触屏位置不在屏幕边缘且当前模型大小不大于最大值,反之亦然。除此之外,考虑到除了模型外还包含其他 UI 元素,为避免对它们的误操作,增加 EventSystem.current.IsPointerOverGameObject(Input.GetTouch(0).fingerId)和 EventSystem.current.IsPointerOverGameObject(Input.GetTouch(0/1).fingerId)判定语句,前者用于单点操作,如果是 false 表示不是 UI,可以执行旋转,后者用于缩放操作,完成后把脚本挂载到相关模型上即可。

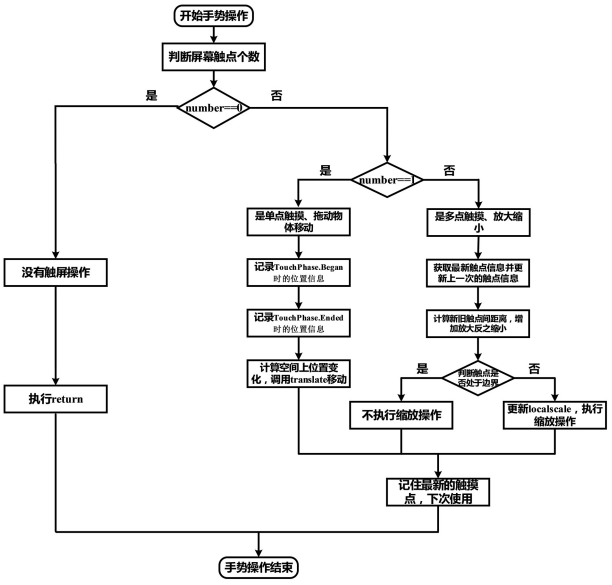

图 2　交互设计流程

也可采用 FingerGestures 插件,通过 C#代理形式实现手姿操作。FingerGestures Unity 插件在 OnEnable 中绑定各种事件方法来监听操作事件,以移动操作为例,FingerGestures.OnFingerDragMove + = OnFingerDragMove。对应的在 OnDisable 中注销手势操作事件,FingerGestures.OnFingerDragMove - = OnFingerDragMove,之后实现 void OnFingerDragMove 方法即可,基本思想同上。在 FingerGestures 的操作对象上要添加一个 2D 碰撞区域,否则会没有效果。

3.3 平面检测

如图 3 所示,平面检测操作流程如下。

1)新建一个 plane,自定义名称 newplane,Position 坐标都设置为 0,Scale 都为 1,否则在识别时可能会出现偏移,在 Materials->Element 0 选择合适的材质。之后进行平面的渲染操作,挂载渲染器脚本 Detected Plane Visualizer,完成平面 prefab 的制作。

2)创建平面检测的生成器 planegenerator,挂载 Detected Plane Generator 脚本,并把上一步制作的平面 prefab newplane 赋给对脚本的 Detected Plane prefab 参数。Detected Plane Generator 脚本用于可视化检测到的平面,基本思想如下:定义 GameObject 类型的 detectedplane 变量保存平面的引用,实例化两个 List,一个保存新检测到的平面(List < detectedplane> newplanes),一个保存所有已检测到的平面(List<detectedplane> allplanes)。从 Session 获取 new 的平面并保存到 newplanes 表,用(1)步制作好的平面 prefab 实例化新平面,把新实例化的平面赋值给 detectedplane 变量用于显示和使用,并保留一份所有已检测到的平面副本。

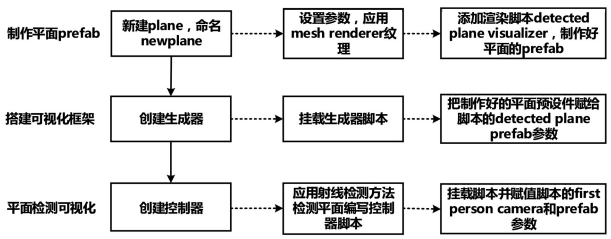

图 3　平面检测流程

3)创建控制器 planecontroller,编写 planecontroller 脚本负责对项目的各种情况进行诊断和控制。控制器脚本有 start 和 update 两个基本方法,前者在挂载控制器脚本的对象开始时执行一次,后者在 AR 应用运行的每一帧都执行。定义 checkphone 函数,检测当前设备是否支持 ARCore、是否授权使用摄像头和是否发生其他错误。在 start 方法中调用 checkphone,如果没有问题,那么就进入 AR 应用循环,在 update 方法中调用 checkstatus 函数,它的主要作用就是跟踪设备状态,设备正在跟踪说明应用正在有效执行,不让应用进入休眠并进入主处理环节,反之则让应用休眠一会,如果应用正在退出,调用 return 不再进行下一步操作。

主处理环节解决虚拟对象显示问题,应用了

ARCore 提供的射线检测方法,在触屏用户触摸的点和摄像机位置间构建一条射线,检测射线与场景中的平面是否发生了碰撞,在检测到平面碰撞的位置放置并显示虚拟对象。关键属性 TrackableHitFlags 用于过滤需要进行碰撞检测的对象类型,本文只检测在边界内和多边形内的平面,根据用户点击点构建射线做碰撞检测:如果发生了碰撞,击中的是平面且不是平面的背面,则实例化虚拟对象 prefab,并生成一个 anchor,把对象的 prefab 挂载到 anchor 以便 ARCore 跟踪定位物体位置;如果击中的是背面,则不做其他操作。关键语句 Vector3.Dot(FirstPersonCamera.transform.position - hit.Pose.position, hit.Pose.rotation * Vector3.up) < 0)的作用是对从摄像机发射到碰撞点的向量与碰撞点的法向量做点积,如果小于 0,说明角度大于 90 度,击中的是背面。把该脚本挂载到控制器 planecontroller 上,并把 ARCore Device 的 First Person Camera 和要显示的虚拟对象 prefab 赋给 First Person Camera 和 Prefab 参数。

在项目开发过程中,要显示的虚拟对象 prefab 不要放在 Hierarchy 窗口,全部放到 Prefabs 文件夹下,否则应用运行后模型会自动出现,跳过了扫描平面的过程。

3.4 图片识别

ARCore 基于图片数据库实现图片识别,支持 PNG 和 JPG 格式的图片,对 JPG 图片要尽量避免过度的压缩。在选择识别图片时,避免含有高度重复或稀疏特征的图片,黑白还是彩色并不会影响识别效果。Unity 目前已经集成了 arcoreimg 图片打分功能(0-100),帮助进行辅助选择。

1)创建图片数据库,把图片导入 Images 文件夹,选中图片,右键 Create-> GoogleARCore ->AugmentedImageDatabase,完成后会生成 New Database 文件。点击该文件即可看到导入图片和评分,尽量选择评分高的。如果在该文件中看到有 error,那么一定要把 error 的图片删除掉,否则会导致其他图片识别出错。之后修改配置文件,选中 ARCore Device ->SessionConfig,把它的 Augmented Image Database 改成新建的 Database 文件即可。

2)根据 4.1+4.2 创建好要显示的虚拟对象 prefab,添加 Augmented Image Visualizer 脚本,这个官方脚本的作用是显示一个相框,所以挂载脚本后会看到 Frame Lower Left 等四个参数,要对脚本进行修改,删除掉这几个参数相关的所有语句,完成虚拟对象 prefab 的更新操作。

3)选中项目的 ExampleController,进入默认挂载的控制器脚本,该脚本的作用是当识别出图片后,跟数据库中的图片进行比对校验,任何一个图片库里的图片被识别后都会创建一个锚点并加载要显示的虚拟对象。这里只需要做一个修改,把(2)步制作好的 prefab 赋值给默认控制器脚本的 Augmented Image Visualizer 参数。

上述操作实现了扫描到图片数据库包含的图片就显示虚拟对象的效果,但存在一个问题,就是不管扫描到图片库的哪个图片显示的都是同一个虚拟空间。原因在于 ExampleController 的默认脚本只加载一个 AugmentedImageVisualizer(public AugmentedImageVisualizer AugmentedImageVisualizerPrefab),现在要加载多个,需要一个集合对象。还有就是 Update 方法也要修改,根据扫描到图片在图片库中的索引显示对应的虚拟对象。要重新编写脚本并对 ExampleController 和 Augmented Image Visualizer 进行修改。

新建 imagevisualizeroverride 和 imagecontroller 脚本,imagevisualizeroverride 关键代码如下所示,然后用该脚本替代上述(2)步挂载的 Augmented Image Visualizer 脚本。

```
namespace GoogleARCore.Examples.AugmentedImage
{
……
public class imagevisualizeroverride:MonoBehaviour
{
public AugmentedImage Image;
public void Update()
{
if (Image == null || Image.TrackingState != TrackingState.Tracking)
return;
}
}
}
```

在 imagecontroller 脚本的定义了一个 List(public List<imagevisualizeroverride> AugmentedImageVisualizerPrefab),private Dictionary<int, imagevisualizeroverride> m_Visualizers = new Dictionary<int, imagevisualizeroverride>(),并在 Update 函数重写了方法,根据数据库图片的索引号显示不同的虚拟对象,关键代码如下:

```
public void Update()
```

```
{
……
foreach (var image in m_TempAugmentedImages)
    {
imagevisualizeroverride visualizer = null;
        m_Visualizers.TryGetValue(image.DatabaseIndex, out visualizer);
if (image.TrackingState == TrackingState.Tracking && visualizer == null)
{
Anchor anchor = image.CreateAnchor(image.CenterPose);
        visualizer = (imagevisualizeroverride)Instantiate(AugmentedImageVisualizerPrefab[image.DatabaseIndex], anchor.transform);
        visualizer.Image = image;
        m_Visualizers.Add(image.DatabaseIndex, visualizer);
    }
}
……
}
```

删除 ExampleController 默认挂载的脚本,挂载新建的 imagecontroller 脚本,Augmented Image Visualizer Prefab 的 Size 参数设置要显示的虚拟空间的数量,把更新脚本后的虚拟对象 prefab 放到对应的 Element X 下,并把 Canvas 的 FitToScanOverlay 拖拽上去。这里要注意,选中 Canvas 的 FitToScanOverlay,在 Inspector 窗口勾选一下,保证应用运行时扫描图片的 UI 组件能够正常显示。

3.5 物体识别

TensorFlow 是当前比较成熟的机器学习、神经网络框架,支持 WINDOWS、MAC 和 LINUX 平台,如果要在安卓移动端使用则要下载对应平台的 dll 文件(TensorFlow.Android.dll)。TensorFlow 使用数据流图 graphs 和会话 sessions 实现功能,前者代表代码间的依赖关系,后者执行操作。首先要创建一个 graph 并设置,从 graph 创建一个 session,通过 session 的 runner 设置输入输出,执行管线流程。

导入 Unity TensorFlow Plugin 插件,在 Assets 下可以看到 ML-Agents 文件夹,里面包含了各种类型平台所需的 TensorFlowSharp dll 文件和代码库。这里请注意,如果导入了较新版本的 TensorFlow 插件(如 0.4),那么要使用 TensorFlow 1.7.1 训练后的模型,并把后缀修改成.bytes,因为需要包含 checkpoint 和本身的 pt 图,可以查阅官方文档。新建 camera 和 detectimg 脚本,前者用于从摄像头获取图像信息,连同训练模型和标签一起传递给 TensorFlow。后者是物体检测的实现代码,利用训练好的模型,调用 TensorFlow 对摄像头输入的图像进行识别。camera 脚本负责摄像头图像的获取和处理、传递训练模型和标签给 TensorFlow 识别并给图像画矩形框。针对前者,如果图像是镜像则翻转回来,如果图像发生了转动,则转回正确显示方向。针对后者,调用在 detectimg 脚本定义的识别方法 detectimgAsync,采用异步方式把从摄像头抓取的图像(taketexturesnap())传输给 detectimgAsync 识别图像中的物体并画出矩形框。

detectimg 脚本的核心是导入图像和物体识别,public detectimg(byte[] model, string[] labels…)是导入图像部分,因为运行在安卓平台,在开头要添加#if UNITY_ANDROID #endif。定义 labels、graph 等,import 模型。public Task<List<BoxOutline>> detectimgAsync(Color32[] data)是识别部分,采用 Task 异步调用和 InvokeRepeating 延迟方法。如果采用 Unity 的 yield+IEnumerator,那么代码要写在 Update 中。需要注意的是,因为是在 Update 中执行,要用多线程函数循环的方式避免造成程序卡顿。

```
……private detectimg detectimger;……
private void Start()
{
……InvokeRepeating(nameof(detectimgcheck), 1f, 1f);……
}
……private async void detectimgcheck()
{
var snap = TakeTextureSnap();
    var scaled = Scale(snap, detectImageSize);
    var rotated = await RotateAsync(scaled.GetPixels32(), scaled.width, scaled.height);
    this.boxOutlines = await this.detectimger.detectimgAsync(rotated);
    Destroy(snap);
    Destroy(scaled);
}……
```

使用 TensorFlow 训练模型机和标签省去训练过程,考虑到是移动端应用,使用 ssd_mobilenet_v1_android_export 移动设备优化模型集和 coco_labels_list 精减标签集。把 camera 脚本挂载到场景摄像机下,并把 ssd_mobilenet_v1_android_export 和 coco_la-

bels_list 赋值给脚本对应的参数。在 Other settings (Unity->Edit->Project Settings->Player)进行属性配置,Scripting Runtime Version 选择.NET 4.x Equivalent、Api Compatibility Level 选择.NET 4.x、Scripting Define Symbols * 填写 ENABLE_TENSORFLOW。

3.6 性能优化

移动端设备的硬件性能相比 PC 端有较大差距,因此,性能优化就显得尤为重要,否则可能出现卡顿、掉帧等问题,导致体验变差。本文主要涉及脚本优化和 UI 优化,脚本优化着重在垃圾回收和对象池。内存管理是脚本设计的关键,针对不同的变量类型,Unity 在栈和堆两种内存池中存取数据,数值类型存储在栈,引用类型存储在堆。栈上内存的分配和释放简单且快速,根据后进先出的原则进行。栈上变量不在作用域时,占用的内存空间马上被返还给栈。堆上的分配则慢得多,而当堆上的对象不在作用域时,所占用的堆内存却不会立刻释放。Unity 的自动内存管理提供了垃圾回收器,确认堆上对象是否还在作用域,如果不在就删除。这个操作会带来较大的系统开销,导致内存空间的碎片化,严重会造成程序卡顿。所以在编程时要从以下几点着手降低垃圾生成:(1)缓存,重复的调用会造成堆内存的重复分配,带来垃圾,可采用缓存解决这个问题,即保存结果的引用并复用。(2)字符串的使用,C#中字符串是固定的,创建和丢弃会产生垃圾,如果某个字符串要多次使用,创建后应该缓存起来。如果脚本包含频繁字符串操作,应该使用动态字符串处理的 stringbuilder 类,该类不产生堆内存分配,可减少垃圾的产生,同时删除掉所有的 debug.log()语句。(3)共用集合,创建集合会在堆上分配内存,所以在脚本中应共用一些集合对象并缓存引用,执行 clear()清空内容复用,减小垃圾。(4)避免在频繁执行的函数内分配堆内存,如 update 和 lateupdate 方法,会造成垃圾的快速堆积,应在 start、awake 等方法中缓存引用。(5)装拆箱,包括装箱和拆箱,因为 C#的数据类型都继承自 system.object,彼此间可以通过显示或隐式操作转换,前者称之为装箱,后者是拆箱。装箱时会在堆上临时创建 system.object 包装类型变量,产生了垃圾,拆箱也是如此。即使在脚本中没有进行装拆箱操作,但插件或间接调用的使用也可能执行了装拆箱操作,所以,在脚本编写时最好少使用可能导致操作的函数,如 object.equals()、string.format()等。

对象池是一种对象的集合,没有内存分配和创建堆中对象的开销,也没有内存释放和销毁对象的开销,特别适合费时的 constructor 和 finalize,可避免对象实例化和销毁时的内存垃圾问题。Unity 中第一次实例化 prefab,要把硬盘或网络的内容加载到内存,将耗费较多的时间,预先把对象加载到对象池可避免频繁的加载和卸载操作。

UI 优化主要包括 3 个方面:(1)drawcalls,图集的使用可以有效降低 drawcalls,把常用图片放在公共图集,单独的放在另外的图集,一个 UI 应用的图集一般不超过 4 个为宜。UI 的 mask 组件和 layout 组件对 drawcall 有较大影响,前者把 UI 分割,导致无法一次性渲染。后者被标记为 dirty 时,要重新计算所有子节点的坐标和尺寸。所以用带通道的图片和 recttransform-based layout 分别代替。此外,设计时要尽量减少 UI 的层级,层级越深 drawcalls 就会越大。(2)动静分离,动态 UI 会导致 canvas 和 batch 的更新,所以应该把它们分开,比如把涉及到移动类的按钮放在一起,动态 UI 变化时只需要重构该部分,不涉及静态部分,这样可以减小计算。(3)图片优化,Unity 默认图片会转化成 texture2d 格式,也就是说最终打包的图片可能会超出原图。所以在设计时,选择合适的图片,如尺寸在 2 的幂次方的图片就比较好。还可以根据应用发布平台选择压缩格式,android 下是 etc,IOS 是 pvrtc4。

除此之外,Unity 的 player setting 参数也要进行正确设置:minimum api level 必须选择安卓 7.0 以上的版本,不选择 multithreaded rendering,XRsettings 一定要选择 ARCore supported。

4 测试与小结

以办公桌面和图片为识别对象,测试效果如图 4~5 所示,扫描到平面显示校舍的三维模型,检测到不同的图片显示不同的虚拟对象。基于 TensorFlow 神经网络模型和移动端优化模型集的物体识别效果如图 6 所示,识别效果尚可。

图 4 平面检测

图 5　多图片识别

图 6　物体识别

本文基于增强现实技术,融合神经网络设计了校园展示应用,两种不同的标识物检测和多图片识别提升了应用的实用性。物体识别功能丰富了应用的内涵,帮助用户更好地了解校园信息。增强现实技术将真实环境与虚拟信息相结合,使得校园展示信息更加全面和方便,进一步增强人们的感知体验,较好地树立学校形象,对未来的高校建设有一定的指导意义。

参考文献:

[1] 吴涛,杨甲乐,陶欣.基于 VR 全景技术的扬州大学校园漫游导航系统的应用研究[J].华中建筑,2019,37(12):28-31.
[2] 姬喆.基于 VR 虚拟漫游技术的交互设计应用研究[J].现代电子技术,2019,42(15):86-90.
[3] 葛岩,吴帆,王泽华,等.基于 Unity3D 的虚拟校园漫游系统设计与开发[J].数字技术与应用,2019,37(6):167-167+169.
[4] 高猛.三维虚拟校园漫游系统中的角色建模与行为控制[J].现代电子技术 2019,42(14):104-107.
[5] 鲁文娟.移动增强现实在开放大学教育教学中的应用探索研究[J].中国教育信息化,2019(22):80-84.
[6] 林茂威,周学栋,林坚,等.增强现实技术在智慧校园建设中的应用[J]. 科技经济导刊,2019,27(16):176.
[7] 郭仁春,钱子扬,王锦宏,等.增强现实技术在校史馆建设中的应用[J]. 沈阳化工大学学报, 2019,33(1):93-96.
[8] 秦胜伟,李重,李金锋,等.校园漫游互动 AR 系统设计与实现[J]. 系统仿真学报,2019,31(7):1367-1376.
[9] 李亮,朱津津,祝凌宇. 虚拟现实与移动增强现实复合性教学环境设计[J]. 中国电化教育,2019(5):98-105.
[10] 林晓凡,朱倩仪,吴倩意,等.增强现实体验式教学资源的科学教育应用:策略与案例[J].中国电化教育,2019(9):60-67.