

一种实用的移动 AR 实验解决方案

谢伙生^{1,2}, 杨 铮¹, 林 晶^{1,2}

(1. 福州大学 数学与计算机科学学院, 福建 福州 350116;

2. 福州大学 网络信息安全与计算机技术国家级实验教学示范中心, 福建 福州 350116)

摘要: 传统移动 AR 实验的底层开发实现较复杂, 不便于 AR 技术的普及, 同时在开发过程中也较难获取到移动端传感器数据。针对该问题提出了一种更为直观和便捷的移动 AR 实验解决方案, 利用 Android Studio 和 Unity3D 通信机制进行传感器数据传输, 并融合 Android Studio、Unity3D 和 EasyAR SDK 来实现移动 AR 实验。使用该解决方案实现了虚实光照一致的移动 AR 实验, 结果表明: 该方案取得了虚实光照一致性的良好效果, 实时性较高, 系统流畅性较好, 为移动 AR 实验教学提供了一种更为有效的实验教学方式。

关键词: 移动增强现实; Unity3D; Android Studio; 光照一致性; 传感器

中图分类号: TK448.21 **文献标识码:** A **文章编号:** 1002-4956(2020)07-0047-06

Resolution for practical mobile AR experiment

XIE Huosheng^{1,2}, YANG Zheng¹, LIN Jing^{1,2}

(1. College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China;

2. National Experimental Teaching Demonstration Center of Network Information Security and Computer Technology, Fuzhou University, Fuzhou 350116, China)

Abstract: The underlying development and realization of the traditional mobile AR experiment are complex, which is not conducive to the popularization of AR technology. At the same time, it is difficult to obtain the data of mobile sensors in the development process. Aiming at the above problems, a more intuitive and convenient mobile AR experiment solution is proposed, which utilizes Android Studio and Unity3D communication mechanism for sensor data transmission, and integrates Android Studio, Unity3D and EasyAR SDK to realize the mobile AR experiment. The solution is used to realize a mobile AR experiment having consistency of virtual and real illumination, and the experimental results show that the scheme has achieved better effect of consistency virtual and real illumination, higher real-time performance and better system fluency, and providing a more effective experimental teaching method for mobile AR experimental teaching.

Key words: mobile augmented reality; Unity3D; Android Studio; illumination consistency; sensors

增强现实 (augmented reality, AR) 技术作为一门新兴技术, 在许多领域都有着巨大的潜力和应用价值, 比如教育领域、医疗领域、娱乐游戏领域等^[1-3]。近年来随着移动设备飞速发展, 因其具有易携带、易使用、显示效果较好等特点, 使得基于移动设备开发的 AR 技术逐渐成为发展主流, 这种技术称为移动

AR^[4-6]技术。

随着人工智能的发展, 实验教学中的移动 AR 实验成为大学计算机相关专业的新兴课程, 熟练掌握移动 AR 开发技术已成为大学生未来就业、创业所要具备的技能之一。因此, 移动 AR 实验的教学地位越来越重要, 但传统移动 AR 实验的底层开发实现较复杂, 不便于 AR 技术的普及, 同时移动 AR 实验一般也较少利用移动设备的传感器数据进行结合开发。针对该问题, 本文提出一种更为直观和便捷的移动 AR 实验解决方案。为了验证该方案的有效性, 使用光照一致性的移动 AR 实验进行验证。由于构建光照模型的方法较复杂, 算法计算量较大, 目前关于光照一致性的

收稿日期: 2019-11-13

基金项目: 国家自然科学基金资助项目 (61801121)

作者简介: 谢伙生 (1964—), 男, 福建宁化, 硕士, 高级实验师, 研究方向为虚拟现实技术、图形图像处理。

E-mail: xiehs@fzu.edu.cn

通信作者: 林晶 (1989—), 女, 福建龙岩, 硕士, 实验师, 研究方向为图像处理、模式识别。

E-mail: 377676261@qq.com

AR 实验研究大多集中在 PC 端,较难在移动端中直接使用^[7-8]。本文利用移动设备的光传感器探测光源方向,通过 Android Studio 和 Unity3D 的通信机制传送传感器数据,同时结合 EasyAR SDK 开发实现了虚实光照一致的移动 AR 实验案例。

1 Unity3D 和 AR SDK

Unity3D 是一个功能丰富的轻量级图形渲染引擎,主要用于游戏的开发,是一个可跨平台的游戏引擎,支持基于 Android、iOS、PC、Linux、tvOS 等多种平台发布,可以使用户轻松实现图形渲染、三维动画和脚本编辑^[9]。Unity3D 界面包括五大基础模块:层级视图(hierarchy)、场景(scence)、游戏(game)、项目(project)、检测面板(Inspector),这些模块使用户方便、直观地开发 3D 应用项目。同时,Unity3D 还具有丰富的 Asset Store,可以下载资源并导入到 Unity3D 项目中,也可以使用外部资源导入。Unity3D 支持 JavaScript、C#、Boo 3 种脚本语言开发,其中 C# 具有强大的 .NET 库做支撑并且有着良好的跨平台性,已逐渐成为 Unity3D 所推荐的开发语言。

国内外有着许多优秀的 AR SDK,传统的有 ARToolkit、ARTag^[10]和 NyARToolkit^[11]等。ARToolkit 是最早流行起来的开源 AR 工具,使用 C/C++ 编辑语言。NyARToolkit 基于 ARToolkit 移植,具有良好的兼容性,支持 Java 语言和 C#。现如今国内外研发了多种基于 Unity3D 的 AR SDK,比如有 Vuforia SDK、Wikitude AR SDK、HiAR SDK、EasyAR SDK^[12]等。相较于传统的 AR SDK,使用基于 Unity3D 的 AR SDK 将极大地减少开发难度。Vuforia 是美国公司 Qualcomm 发布的一款移动终端 AR 开发引擎。EasyAR SDK 作为国内自主研发的一款增强现实工具包,已经具备完善的 Target 种类和较便捷的识别图设置模式,同时具有单目标识别、多目标识别、3D 物体识别、云端服务等多种功能以及丰富的 API 和完备的用户手册,深受国内外开发者的青睐。

Unity3D 是一个具有可视化操作界面的专业游戏引擎,Android SDK 又提供了开发移动端应用的各种组件^[13],这些都极大地满足了移动 AR 应用的各项需求。移动 AR 应用中通过增加虚实光照一致性能,较好地消除了虚拟模型显示的突兀感,增强了虚拟信息与现实场景的融合。

2 移动 AR 实验的解决方案

实现一个结合传感器数据的移动 AR 实验需要以下步骤:1)结合 Unity3D 和 AR SDK 搭建基本 AR 实验环境;2)在 Android Studio 中获取移动端传感器

数据;3)Unity3D 工程与 Android Studio 工程相结合进行通信、传输传感器数据。

2.1 结合 Unity3D 和 AR SDK 搭建基本 AR 实验环境

本文采用 EasyAR SDK 开发,它是国内自主研发的一款免费增强现实开发工具,能够在 Android、iOS、PC 等多平台实现。具备单目标识别、多目标识别、3D 物体识别、云端服务等功能,是一款完善的增强现实开发工具^[14]。利用 EasyAR SDK 在 Unity3D 中搭建基本 AR 实验环境的步骤如下:

1)在 EasyAR 官网下载 EasyAR SDK for Unity 工具包,并注册账号创建 License key。

2)将 EasyAR.unitypackage 资源包导入(Import) Unity3D 中,包内组件如图 1 所示。

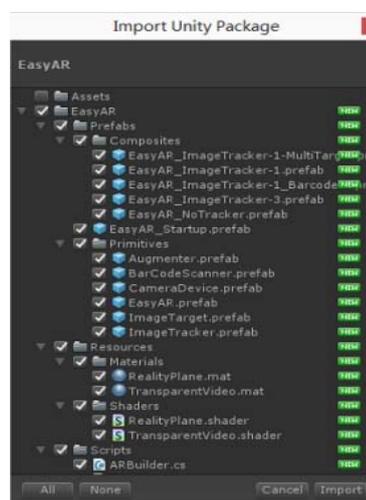


图 1 导入 EasyAR SDK 到 Unity3D

3)移除层级视图中初始摄像机 Main Camera,添加 Prefabs 文件夹中的 AR 相机组件 EasyAR_Startup,在其检测面板中输入 1)中获取到的 License key。

4)添加 ImageTarget 组件,在该组件下创建的 3D 物体是作为虚拟显示模型,添加虚拟光源组件 VirtualLight,如图 2 所示。

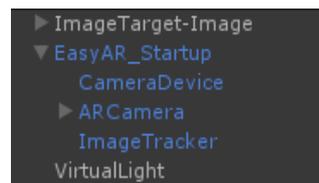


图 2 各组件层级视图

5)选择 ImageTarget 组件,在检测面板中设置 Marker 图标,输入 Marker 的 path 和 name 属性并调整位置和大小。Marker 图标路径位于 StreamingAssets 文件夹下,如图 3 所示。

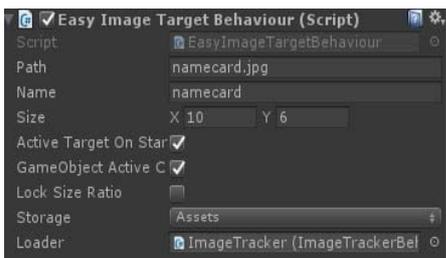


图 3 设置 Marker 图标属性

6) 选择菜单 File>Build Settings 设置 Android 运行模式, 点击 Player Settings>Other Settings>Identification 填写 SDK 包名, 点击 Build 即可生成 Android APK。

2.2 Android Studio 中获取传感器数据

Google Android API 上的传感器访问通过传感器管理类 SensorManager 进行, 并使用 SensorListener 来检索测量^[15]。SensorManager 是用来获取传感器服务和选择传感器类型。SensorListener 是传感器监听器, 用于监听传感器数据的变化。Sensor 是传感器信息抽象类, 通过 getType() 方法获取不同的传感器类型, 包括光传感器、方向传感器、磁场传感器、重力加速度传感器等。获取的传感器数据以长度为 3 的数组形式存在。常见传感器表示方法如表 1 所示, 不同的传感器类型如表 2 所示。

表 1 常见的传感器表示方法

方法	含义
getMaximumRange()	最大取值范围
getRotationMatrix()	计算旋转矩阵
getDefaultSensor()	初始化传感器类型
getType()	传感器类型
remapCoordinateSystem()	坐标系转换

表 2 常见类型传感器表示方法

传感器类型	名称
Sensor.TYPE_ACCELEROMETER	加速度传感器
Sensor.TYPE_MAGNETIC_FIELD	磁场传感器
Sensor.TYPE_ORIENTATION	方向传感器
Sensor.TYPE_GYROSCOPE	陀螺仪
Sensor.TYPE_LIGHT	光传感器

2.3 Unity3D 与 Android Studio 结合进行通信传输

Android Studio 是一款 Android 集成和调试的开发工具, Unity3D 导出的 Android 工程可以很完美地结合到 Android Studio 工程中进行二次开发并传输数据。图 4 为 Unity3D 和 Android Studio 通信机制图。

如图 4 所示, Android Studio 是通过 com.unity3d.player 包里提供的函数和类调用 Unity3D 方法。其中

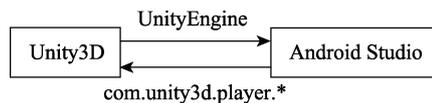


图 4 Unity3D 和 Android Studio 通信机制

UnityPlayer.java 是 Android Studio 中图像渲染和信息传递功能的重要类, UnityPlayer.java 类中封装有函数 UnitySendMessage (gameObject,fuction, param), 该函数用于 Android Studio 和 Unity3D 之间通信传输, 输入的 3 个成员变量分别为游戏对象、需要进行通信的函数名以及需要传递的数据参数。Unity3D 是通过 UnityEngine 工具包里所提供的各类 API 调用 Android Studio 方法, 主要通过 AndroidJavaObject 对象及 AndroidJavaClass 类来实现。AndroidJavaObject 是 Android Java 对象, 通过构造函数实例化 Android 类, 并进行函数及成员变量的调用。实例化方式: AndroidJavaObject obj=new AndroidJavaObject (“className”)。AndroidJavaClass 是 Android Java 类, 里面包含着各种 java 类的操作。

Unity3D 与 Android Studio 相结合的具体实现步骤如下:

1) 在 Unity3D 中设置 Player Settings 导出安卓工程。以 Unity3D 5.5.0 版本为例, 如图 5 所示, 将 Build System 选项设置为 Gradle, 勾选 Export Project, 点击 Export 导出项目。导出后的目录结构如图 6 所示。

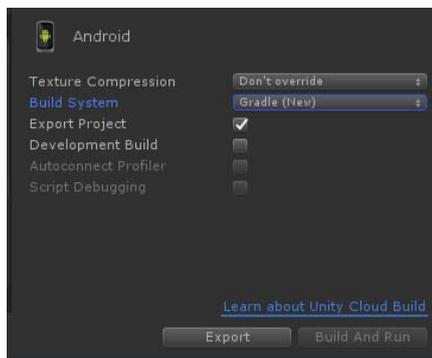


图 5 Unity3D 中设置导出 Android 工程

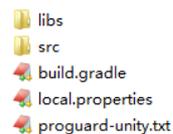


图 6 Unity3D 导出工程目录

2) 在 Android Studio 中新建一个项目, 将 Unity3D 导出工程中的文件全部替换到 Android Studio 中相应的文件夹中, 包括 java、jniLibs、assets、res 文件夹以及 AndroidManifest.xml 配置文件。配置好的文件目录如图 7 所示。这样就初步完成了一个 Unity3D 结合

到 Android Studio 中的工程搭建。需要注意的是, Unity3D 端导出的文件夹只有在首次配置中才需要全部替换, 而后当需要修改 Unity3D 工程中的文件时只需要将 Unity3D 导出工程中的 assets->bin->Data 文件替换到 Android studio 对应的 Data 文件夹即可。路径如图 8 所示, 图 8(a)为 Unity3D 端 Data 文件路径, 图 8(b)为 Android Studio 端 Data 文件路径。



图 7 Android Studio 配置好的文件目录

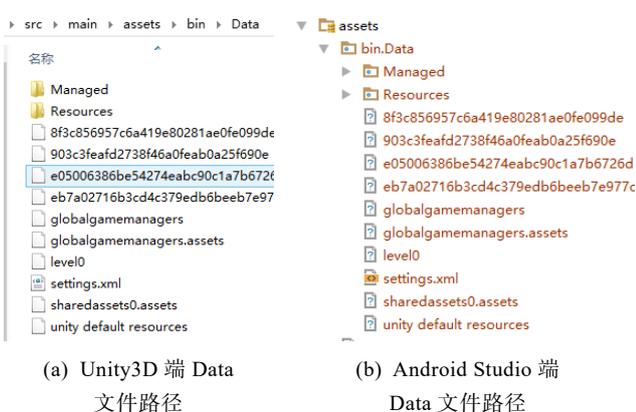


图 8 文件路径

3 光照一致的移动 AR 实验案例

AR 技术有别于 VR 技术, AR 技术更注重在真实世界中叠加虚拟信息。虚拟信息高度融合于现实世界主要体现为几何一致性、光照一致性和时间一致性^[16]。其中光照一致性指的是在虚拟世界中的光照强度、方向等信息与现实世界中的光源信息保持一致, 从而消除虚拟模型显示的飘浮感, 这是影响虚拟模型显示效果的重要因素。

本文利用移动 AR 解决方案, 实现了带有虚实光照一致性效果的移动 AR 实验。通过转动手机, 利用移动设备自带的光传感器采集光强数据, 探测到最大光强后近似将摄像头的位置方向当作现实场景中的光源位置方向, 将摄像头位置的旋转数据作为虚拟场景

中的光源位置的旋转数据更新光源方向, 最终渲染出三维模型及阴影。实验能实时渲染与现实光照方向较为一致的阴影并与虚拟三维模型融合, 达到虚拟场景和现实场景光照一致的视觉效果。

3.1 案例实验架构

利用 Unity3D 和 Android Studio 结合开发并通过 Android 传感器采集现实世界光源信息, 其中 Marker 图的追踪识别和摄像机姿态的获取是利用 EasyAR SDK 工具包来实现, 本案例实验的整体架构如图 9 所示。Android Studio 端主要负责光传感器的注册和监听, 获取真实场景中 Marker 图周边环境的光强值并迭代出最大的光照强度, 将最大光强数据传递给 Unity3D 端并同时触发改变虚拟光源的函数; Unity3D 端主要负责 EasyAR SDK 图像识别追踪包的搭建和使用, 获取相机标定过程中 AR 摄像机的旋转数据, 同时开启陀螺仪监测功能, 获取手机在三维空间的姿态数据。根据追踪结果的不同使用不同的旋转数据赋值给虚拟光源, 改变虚拟光源在虚拟世界中的方向位置, 渲染三维模型和光照方向正确的阴影效果, 最后融合到真实场景中。

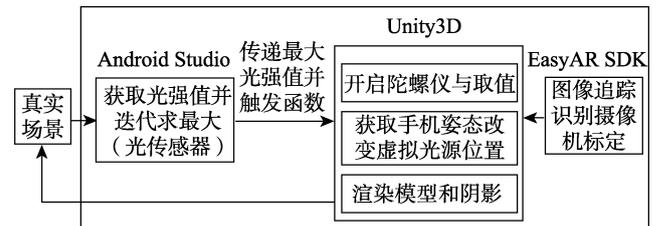


图 9 案例实验架构

3.2 案例实验流程

基于光照一致性的移动 AR 案例实验流程如图 10 所示, 该实验基于 Android Studio 端和 Unity3D 端同时进行。Android Studio 端首先初始化注册光照传感器, 同时监听光照强度变化, 对比当前光强值是否大于最大光强值, 若小于最大光强值则不做操作继续监听; 若大于最大光强值则更新最大光强数据, 将光强值进行通信传输传递给 Unity3D 端。Unity3D 端利用 EasyAR SDK 搭建 AR 环境, 主要负责摄像头图像帧的获取, 对 Marker 标记进行检测和识别。Unity3D 端的函数每一帧都在更新, 如果 Android Studio 端监听到最大光强值改变时则会触发 Unity3D 端更改虚拟光源方向的函数。在函数中判断 Marker 标记的检测匹配是否成功, 如果成功, 获取当前手机 AR 相机姿态旋转数据后赋值给虚拟光源组件的旋转值; 如果失败, 则获取手机陀螺仪数据并矫正手机当前姿态, 当 Marker 标记识别成功后更新虚拟光源组件的位置和光照强度, 最终渲染阴影输出模型融合效果。

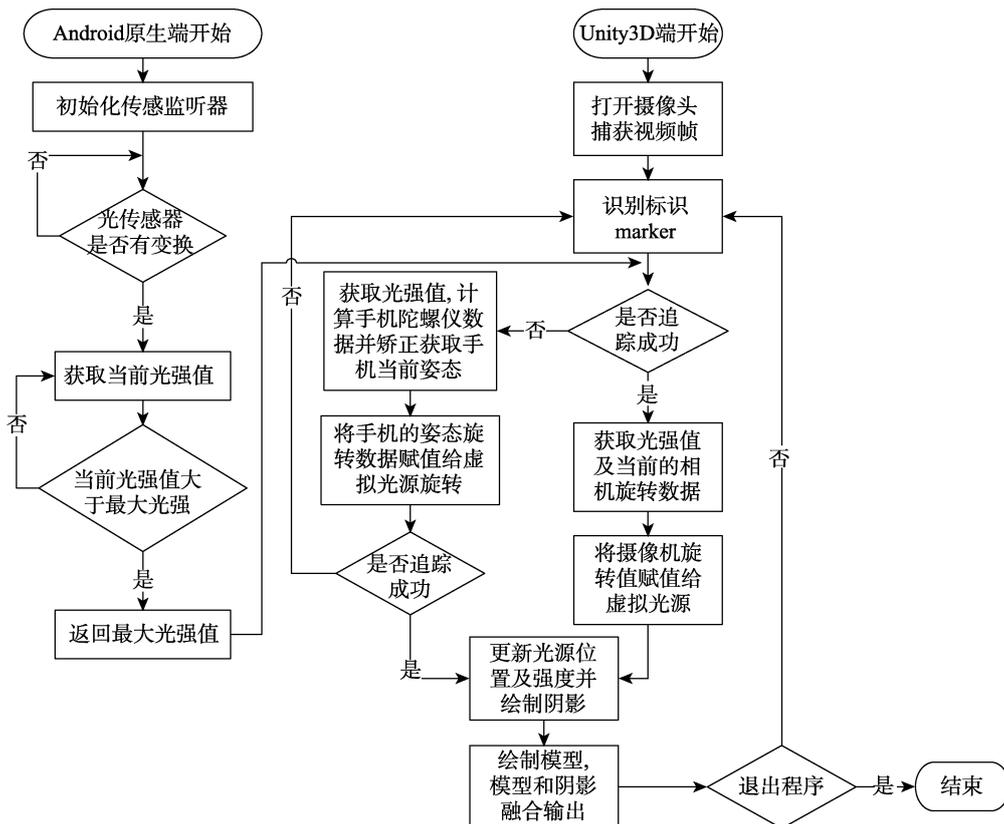


图 10 案例实验流程图

3.3 案例实验的结果与分析

本实验使用 Android Studio 和 Unity3D 结合开发, 使用 Java 程序语言开发生成 APK, 采用 Android 手机运行测试, 具体运行环境软硬件配置如表 3 所示。

表 3 运行环境配置表

配置类型	具体描述
软件设备	Windows 端 集成开发环境: Android Studio、Unity3D 操作系统: 64 位 Microsoft Windows 7
	手机客户端 操作系统: Android4.0 及以上版本
硬件设备	Windows 端 内存 8 G、处理器 Intel Core i5 主频 3.30 GHz
	手机客户端 内存 2.0 G、处理器四核、主频 2.5 GHz 后置摄像头 800 万、分辨率 1 920×1 080

采取单个类似聚灯光源进行实验, 虚拟模型采用现实场景中常见的物体形状, 并与现实世界中类似的物体模型进行比较。分别在模型追踪成功和失败的 2 种条件下转动手机采集周边光照信息, 观察阴影实时变化情况。

1) 阴影效果测试。

采取简单球体模型作为虚拟信息, 一般的 AR 实验无阴影效果如图 11(a)所示, 虚拟模型加了阴影后的效果如图 11(b)所示, 通过对比可以看出, 加了阴影后的模型显得更加逼真, 减少了模型的漂浮感, 增强了

用户的视觉体验。

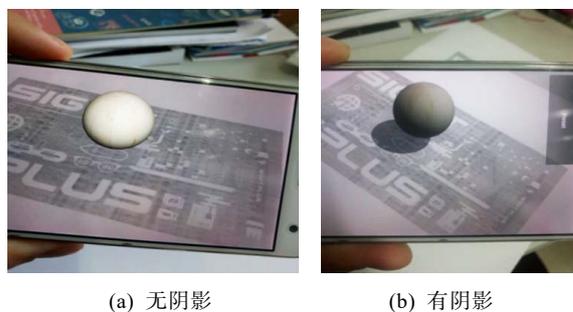


图 11 阴影效果对比

2) 阴影实时性和模型光照强度动态变化测试。

图 12 为手机逆时针绕着 Marker 图中心位置转动时阴影和模型的变化情况, 光源位置在 Marker 图的右侧上方。图 12(a)是摄像头打开的初始状态, 此时还未转动手机进行光照强度的探测, 实验默认此时为最大光强位置, 渲染该方向阴影; 当手机逆时针向右移动到图 12(b)时, 实验探测到更大的光强值, 更新了阴影方向及模型光照亮度; 当手机再次移动到图 12(c)时探测到当前环境的最大光强, 实验显示模型亮度达到最大并且重新渲染更新阴影方向; 此时再次转动手机探测不到更大的光强值, 实验保持图 12(c)时的模型和阴影方向状态的渲染。实验可以看出虚拟模型的阴影方向和模型的亮度都是根据现实场景中的光照情况实时

变化的。实验结果表明：该方案能够较快地实时更新阴影的方向，并且更新模型的光照强度状态，运行较流畅。

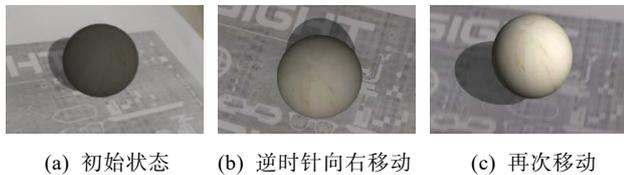


图 12 阴影实时性与模型光照强度动态变化测试

3) 阴影方向准确性测试。

测试使用不同形状的虚拟模型与现实世界中相近的物体进行比较。光照一致性的实验渲染效果如图 13 所示，图 13(a)选用球体模型，图 13(b)选用柱体模型，分别在现实场景探测光照强度后实时渲染阴影。实验结果表明：Android 光传感器能够敏感地探测周边光强变化，虚拟物体能够较好地渲染与真实物体方向较一致的阴影，达到虚实光照一致的目的。

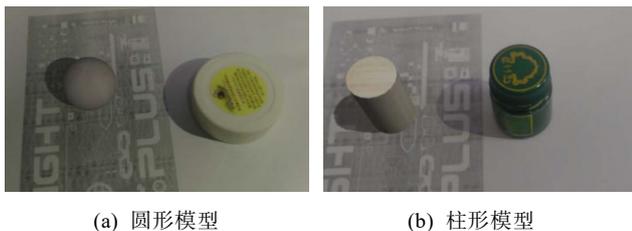


图 13 光照一致性的实验阴影效果

4 结语

利用 Unity3D 能够较为快速地实现 AR 实验，同时与 Android 原生开发结合能够较轻松地实现信息采集和数据通信，极大地降低开发难度，便于实验教学和 AR 技术的普及。本文结合 Unity3D 和 Android Studio 同时使用 EasyAR SDK 设计实现了一个更为直观和便捷的移动 AR 实验解决方案，并通过该解决方案实现了光照一致性的移动 AR 实验，通过与现实场景中的物体光照效果比较实验，表明该方案能够渲染

出与现实场景中较为一致的阴影方向，实时性较好，系统较流畅，能达到预期效果。

参考文献 (References)

- [1] 侯颖, 许威威. 增强现实技术综述[J]. 计算机测量与控制, 2017, 25(2): 1-7.
- [2] 钟毅. 增强现实(AR)技术的教育应用略谈[J]. 电子世界, 2018, 546(12): 63-64.
- [3] 李帅. VR/AR 技术的机遇和挑战[J]. 文化创新比较研究, 2018(22): 52-53.
- [4] 郭婉莹. 增强现实技术的发展与应用[J]. 科技创新与应用, 2017(20): 180, 182.
- [5] 李丹, 程耕国. 基于 Android 平台的移动增强现实的应用与研究[J]. 计算机应用与软件, 2015, 32(1): 16-19.
- [6] CHATZOPOULOS D, BERMEJO C, HUANG Z, et al. Mobile augmented reality survey: From where we are to where we go[J]. IEEE Access, 2017(4): 6917-6950.
- [7] 彭先玲, 陈一民, 黄晨. 基于光线传感器的光照一致性算法[J]. 微型电脑应用, 2017(4): 69-73.
- [8] WU X F, JIN G F, ZHU J. Freeform illumination design model for multiple light sources simultaneously[J]. Applied Optics, 2017, 56(9): 2405-2410.
- [9] 陈泽婵, 陈靖, 严雷, 等. 基于 Unity3D 的移动增强现实光学实验平台[J]. 计算机应用, 2015(s2): 194-199.
- [10] 王宇希, 张凤军, 刘越. 增强现实技术研究现状及发展趋势[J]. 科技导报, 2018, 36(10): 75-83.
- [11] 周国众. 移动增强现实关键技术及应用[J]. 测绘与空间地理信息, 2012, 35(9): 140-144.
- [12] 罗永东. 基于 Unity3D 的移动增强现实技术与应用研究[D]. 青岛: 青岛科技大学, 2015.
- [13] 齐越, 马红妹. 增强现实: 特点、关键技术和应用[J]. 小型微型计算机系统, 2004, 25(5): 900-903
- [14] 郑明月, 李颖, 焦梓, 等. 基于 Android 的增强现实应用研究[J]. 现代计算机(专业版), 2017(13): 35-39, 43.
- [15] MARKUS K, DAVID J, MURPHY. MARA-sensor based augmented reality system for mobile imaging[C]//ISMAR 2006, New York, USA: IEEE, 2006: 67-73.
- [16] 刘万奎, 刘越. 用于增强现实的光照估计研究综述[J]. 计算机辅助设计与图形学学报, 2016, 28(2): 197-207.