

6-5-2020

## Augmented Reality System Based on Depth Image Segmentation and Object Tracking

Jiabo Yang

*School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China;*

Yang Gang

*School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China;*

Yang Meng

*School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

---

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

---

# Augmented Reality System Based on Depth Image Segmentation and Object Tracking

## Abstract

**Abstract:** In this paper, an augmented reality scheme that can achieve the virtual-real objects' interaction effectively is proposed. The depth camera, such as Kinect, is adopted to capture the depth image of scene, and then the segmentation and dynamic tracking of the scene objects are implemented based on the depth image. In the scene segmentation, *we propose a scene segmentation strategy based on the prior knowledge. The large plane objects existing in the scene are firstly recognized according to the prior knowledge. Then, the clustering segmentation is applied to the remaining points to get the scene objects. This strategy is very efficient for the indoor desktop-scene segmentation.* In the object tracking, we adopt the three dimensional tracking algorithm based on particle filtering. *The tracking process can also be accelerated by using the prior large plane information.* This paper provides an efficient scheme for implementing the virtual-real objects' interaction.

## Keywords

augmented reality, depth image segmentation, object tracking, virtual-real objects interaction

## Recommended Citation

Yang Jiabo, Yang Gang, Yang Meng. Augmented Reality System Based on Depth Image Segmentation and Object Tracking[J]. Journal of System Simulation, 2017, 29(11): 2788-2795.

# 基于深度图像分割与物体跟踪的增强现实系统

杨家博, 杨刚\*, 杨猛

(北京林业大学信息学院, 北京 100083)

**摘要:** 提出一种能实现虚、实物体互动的增强现实解决方案。该方案中, 采用 Kinect 等设备获取现实场景的深度图像, 并基于深度图像对实景中的物体进行分割及动态跟踪。在场景分割时, 提出一种基于先验知识的分割策略, 首先识别出场景中先验存在的大平面, 然后再对剩下的点云数据进行聚类, 该方式对室内桌面场景分割效果良好。在物体跟踪时, 采用了基于粒子滤波的三维跟踪算法, 该过程中同样可以通过先验的大平面信息进行加速。该方法为实现虚、实物体互动提供了一种高效、便捷的实现方案。

**关键词:** 增强现实; 深度图像分割; 物体跟踪; 虚实交互

中图分类号: TP391.41

文献标识码: A

文章编号: 1004-731X (2017) 11-2788-08

DOI: 10.16182/j.issn1004731x.joss.201711027

## Augmented Reality System Based on Depth Image Segmentation and Object Tracking

Yang Jiabo, Yang Gang\*, Yang Meng

(School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China)

**Abstract:** In this paper, an augmented reality scheme that can achieve the virtual-real objects' interaction effectively is proposed. The depth camera, such as Kinect, is adopted to capture the depth image of scene, and then the segmentation and dynamic tracking of the scene objects are implemented based on the depth image. In the scene segmentation, we propose a scene segmentation strategy based on the prior knowledge. The large plane objects existing in the scene are firstly recognized according to the prior knowledge. Then, the clustering segmentation is applied to the remaining points to get the scene objects. This strategy is very efficient for the indoor desktop-scene segmentation. In the object tracking, we adopt the three dimensional tracking algorithm based on particle filtering. The tracking process can also be accelerated by using the prior large plane information. This paper provides an efficient scheme for implementing the virtual-real objects' interaction.

**Keywords:** augmented reality; depth image segmentation; object tracking; virtual-real objects interaction

## 引言

随着软硬件技术水平的提高, 增强现实

(Augmented Reality, 简称 AR)系统逐渐走向实际应用, 并成为当前最受关注的技术热点之一。但是, 当前大部分 AR 应用只是通过三维注册实现了虚拟物体与实景图像的叠加, 这种叠加只是对虚、实物体在二维图像层面上的对齐, 却无法实现虚、实物体在三维空间中遮挡、碰撞等带有三维性质的互动。例如, 奇幻味味、小熊尼奥等应用中, 主要根据标记图案的特征点检测来实现相机的方位标定,



收稿日期: 2017-08-30 修回日期: 2017-09-15;  
基金项目: 中央高校基本科研业务费专项资金 (2015ZCQ-XX);

作者简介: 杨家博(1991-), 男, 广西南宁, 硕士生, 研究方向为虚拟现实; 杨刚(通讯作者 1977-), 男, 山西长治, 博士, 副教授, 研究方向为虚拟现实; 杨猛(1982-), 男, 河北秦皇岛, 博士, 副教授, 研究方向为计算机图形学。

<http://www.china-simulation.com>

• 2788 •

由此实现虚、实空间的三维注册, 完成虚实内容的叠加。这其中并没有获取现实场景的三维信息, 无法对现实场景中的物体进行分割、定位, 也就无法实现虚拟物体与实景物体的遮挡判断、碰撞等交互效果。

要完成虚、实物体的互动, 关键在于获取现实场景的三维信息。一些研究者将机器人领域的视觉 SLAM 技术引入到 AR 系统中<sup>[1-3]</sup>, 力图通过 SLAM 技术完成现实场景的三维信息获取。但这些技术尚处于实验阶段, 结果不稳定、计算量较大, 尚无法实用。近年来, 深度感知设备获得了长足进步, 出现了很多消费级别的小型深度感知设备, 如 Kinect, Real Sense 等。人们可以利用这些深度感知设备获得场景到深度信息。这就为场景的识别、理解和重建研究带来了巨大遍历。本文即基于深度摄像设备, 提出了一种实现虚、实物体互动的增强现实解决方案。我们利用 Kinect 等目前流行的深度摄像设备获取现实场景的深度图像, 基于该深度图像进行实景物体的分割与追踪, 从而达到对现实场景的更深入感知; 在此基础上, 设计开发了一个增强现实系统, 能够有效实现虚拟物体与现实物体的遮挡、碰撞等交互效果。

本文要解决的关键问题是基于深度图像的场景分割与目标物体跟踪。基于图像的分割或基于深度图像的分割问题已经有大量研究, 但这些面向通用场景的分割方法一直存在准确率不高、计算量大的问题, 很难应用于实际系统。考虑到本文 AR 系统的一种典型应用场景是室内桌面场景, 针对这种室内桌面场景, 我们提出了一种有效的场景分割策略。首先快速识别出场景中桌面、墙面等大平面, 在去掉这些占据大量点云的大平面后, 再对剩下的点云进行聚类分割。实践表明这个方案是非常有效的, 效率高, 分割效果好。在目标跟踪方面, 我们将分割得到的物体作为跟踪目标, 采用三维粒子跟踪算法实现了对场景中多个物体的实时跟踪。在此过程中, 我们同样采用了首先去除大平面区域, 再

对剩余点云进行跟踪的方式。

在系统开发方面, 本文将 Unity3D 平台与外部动态链接库结合进行整个 AR 系统的构建。场景分割及跟踪等核心算法都封装在 DLL 中, 在运行时该 DLL 会将分割、跟踪得到的数据传入 Unity 中; 然后在 Unity 中进行虚实内容的融合, 并对虚、实物体的遮挡和碰撞等进行计算。

## 1 方法总览

本文方法流程如图 1 所示。

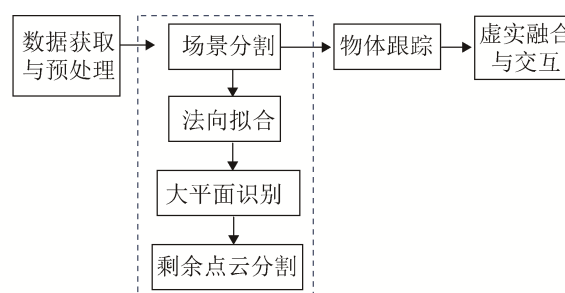


图 1 方法流程

Fig. 1 Method flow

本文方案包含四个步骤(如图 1 所示):

(1) 数据获取与预处理。本文首先通过 Kinect 获取得到彩色图像和深度图像, 然后将深度图像转化为点云数据; 对转化得到的点云数据进行预处理, 降低数据密度, 并去除噪声点、离群点, 为下一步的分割做好准备。

(2) 场景分割。首先对点云进行法向估计, 并借助法向, 采用 RANSAC 算法进行大平面识别; 然后对去除大平面后的剩余点云进行聚类分割。

(3) 物体跟踪。将分割出来的点云物体作为跟踪目标, 采用三维粒子滤波算法, 实现同时对多个目标物体的实时跟踪。

(4) 虚实融合与交互。依据深度图像所提供的三维信息, 可以构建出视频实景的三维投影坐标系, 由此实现实景与虚拟物体的三维注册与融合。进一步地, 为分割得到的实景物体构建虚拟的代理几何体, 由此进行虚、实物体的遮挡、碰撞等交互计算。

## 2 关键技术

### 2.1 点云数据预处理

从 Kinect 拍摄得到的深度图像可以转化成为三维点云数据,但这种点云数据还无法直接使用,需要经过一系列的预处理。预处理有三个目的:切割出有效区域、去除噪声点和离群点、降低数据密度,分别论述如下。

#### (1) 切割有效区域。

Kinect 深度相机和彩色相机的位置有一定差别,FOV 角度也不一样,使其拍摄得到的画面并不完全对应;并且 Kinect 深度相机有一定的有效范围,对不同位置和角度扫描的精度也不一样。为此我们首先需要切割出深度相机所拍图像的有效范围,之后才能进行后续处理。在本文实验中,我们采用了一个经验性的坐标范围来进行截取: X 轴范围是 $[-1.0\text{ m}, 1.0\text{ m}]$ ; Y 轴是 $[-0.3\text{ m}, 0.5\text{ m}]$ ; Z 轴,即深度为 $[0.0\text{ m}, 1.5\text{ m}]$ 。

#### (2) 去除噪声点和离群点。

由于硬件性能及拍摄环境的问题, Kinect 扫描所产生的深度图像会带有很多噪声,并可能会产生一些稀疏的离群点。这些噪声点和离群点会对后面的场景分割产生影响,需要去除。对于噪声点,本文选用了非线性滤波中常用的中值滤波进行去噪处理。对于离群点,我们通过分析点云间的距离进行判断和去除。离群点一般处于图像中场景物体的边缘,密度非常稀疏。可以通过计算某个点与其一定范围内的邻域点的平均距离(我们简称之为邻域距离)来衡量该点所在区域的稀疏程度。首先统计出点云集合中所有点的邻域距离,这些邻域距离应构成一个高斯分布,计算出该高斯分布的均值和标准差  $\sigma$ ; 则我们将邻域距离超出  $2 \times \sigma$  的点认定为离群点予以去除。

#### (3) 降低点云密度。

Kinect 所获取的点云数据中两点间的距离是毫米级的,点云非常密集。过多的点云数量会对后续的分割及跟踪带来困难。为此,我们需要尽量减

少点云数量,但同时又要保持各区域点云的形状特征。我们采用一种体素格滤波器对点云进行采样。体素格滤波器会采用一个三维的立方体体素来对点云数据进行处理,该立方体的大小可以根据拟保留的点云密度而决定。在每个立方体体素内,根据体素内所有点云可以计算出一个重心位置,用该重心点来代表该体素的所有点,由此得到经过稀疏化的点云数据。在本文实验中,体素格滤波前的点云数量可达 20 万,滤波后减少为 5 万,可大大降低后续的计算工作量。如图 2 所示,左图为体素格滤波前的点云,含 217088 个点;右图为经过体素格滤波后的点云,简化为 51466 个点。

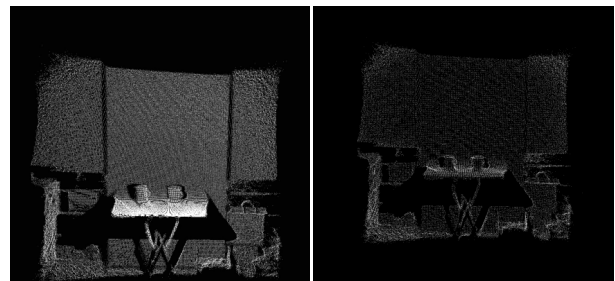


图 2 体素格滤波  
Fig. 2 Voxel grid filter

### 2.2 场景分割

对点云数据进行预处理后得到了较精简、有效的点云数据。在此基础上我们对场景进行分割。本文方法的一个典型应用场景是室内的桌面场景,在这种场景中,桌面,有时还有背景墙面往往会占据相当大的空间。针对这种先验性的场景特征,我们的策略是:先识别出场景中的大平面,然后对去除大平面之后的剩余点云进行聚类分割。这种策略可以提高分割的效率和准确性,原因在于:平面物体易于识别,而去除大平面之后的点云数据则可以被更快速、准确地分割出来。

这其中的一个关键点是如何识别场景中的大平面。最小二乘法可以用来拟合平面,但其只适用于没有大量噪声点或局外点(outlier)的情况;随机采样一致性算法<sup>[4]</sup>(RANSAC 算法)则可以有效克服这个问题,但其依赖于随机点的选取,可能需要

大量的迭代次数。我们对二者进行了一个结合: 首先采用最小二乘法对点的局部区域进行拟合, 计算出所有点的法向; 然后, 在 RANSAC 算法中, 根据法向来选取合适的内点, 实现大平面的快速拟合。下面对场景分割中的几个关键步骤分别进行论述。

### 2.2.1 法向估计

求取点云数据中点的法向已有许多比较成熟的方法, 我们采用了最为经典的局部表面拟合法<sup>[5-6]</sup>进行法向估计。该方法的主要思想是, 假设被采样的点云表面处处光滑, 基于此我们可以利用任意点的邻域通过最小二乘法拟合出一个平面, 把平面法向作为这个点的法向量。假设任意点  $p$ , 获取其最相近的  $m$  个相邻点, 而后通过最小二乘法计算局部平面  $P$ , 此平面可以用如下公式表示:

$$P(n, d) = \arg \min \sum_{i=1}^m (n \cdot p_i - d)^2 \quad (1)$$

其中:  $n$  为  $P$  的法向量, 也就是点  $p$  的法向量;  $d$  为  $P$  到坐标原点的距离。如图 3 中显示了采用上述方法估算出的点的法向。图中绿色表示点云, 白色短线指示点的法向, 左、右图分别是不同角度观察场景的效果。

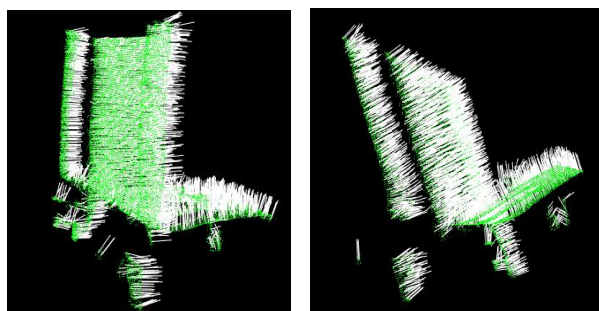


图 3 法向估计结果。

Fig. 3 Result of normal estimation.

### 2.2.2 大平面识别

按照 RANSAC 算法, 大平面识别的基本步骤为:

(1) 在点云中随机选取 3 个不共线的点, 根据此三个点求出当前的平面方程; 这 3 个点组成了初始内点集  $S$ ;

(2) 计算所有其他点到该平面的距离  $d$ , 如果  $d$  小于某个阈值  $t$ , 则将此点加入到内点集  $S$  中;

(3) 若内点集的数量大于某个阈值  $N$ , 则认为该平面成立, 并利用内点集  $S$ , 采用最小二乘等方法重新计算新的平面; 否则放弃该内点集  $S$ ;

(4) 重新随机抽取新的  $S$ , 重复以上过程; 如此重复迭代  $M$  次, 从中选取最大的  $S$ , 以此  $S$  拟合得到的平面即为所求平面。

在此算法中, 初始内点的选取很重要, 如果采用完全随机的选取方法, 往往需要很大的迭代次数  $M$ , 才能得到满意的结果, 效率较低。本文根据场景的特点对初始内点的选取进行限制, 以达到快速拟合的目的。

在上一节中, 我们已经求出了所有点的法向。对这些法向进行一个直方图统计分析, 可以找出分布最集中的法向区间。这些法向区间所对应的点集往往就是场景中的大平面点集。当进行初始内点选取时, 我们可以限定只从属于某个法向区间的点中选取, 由此可以具有更高的成功率。从而可以减少重复迭代的次数。

在我们的实验中, 如果采用传统的随机方式选取初始内点集, 则迭代次数  $M$  需要 500 次~1000 次左右才能得到合适的结果; 而采用本文上述方法, 则可使迭代次数降低到 10 次, 即使算上法向估计所花费的时间, 也可提升效率达 40%。具体实验数据见表 1。

表 1 计算效率统计

Tab. 1 Computational Efficiency statistics		/ms			
方法	预处理	法向拟合	大平面识别	剩余点云分割	物体跟踪(fps)
本文	915	455	104	63	31
对比方法 1		1031(RANSAC 大平面识别)			20
对比方法 2		2035(聚类分割时间)			

### 2.2.3 剩余点云的分割

在去除大平面之后, 我们采用经典的欧式距离聚类分割算法对剩余点云进行聚类分割, 分离得到

场景中的一个物体。在进行聚类分割之前, 我们还会对剩余点云再进行一次离群点去除操作, 目的是为了去除识别大平面时产生的一些离群点。

分割结果如图 4 所示, 左上为深度图像转换成的原始点云; 中上、右上为分割后得到墙面、桌面(为了更加直观的观察桌面的分割情况, 我们对其进行一定角度的旋转); 左下图为对大

平面抽取后的剩余点云使用聚类分割方式分割得到的不同集合, 不同颜色对应桌面与桌面物体; 中下图是从另一个角度观看的效果, 不同颜色标记分割后墙面、桌面与物体; 右下图为对分割出的物体求取其虚拟包围盒后映射到彩色图像上效果图。

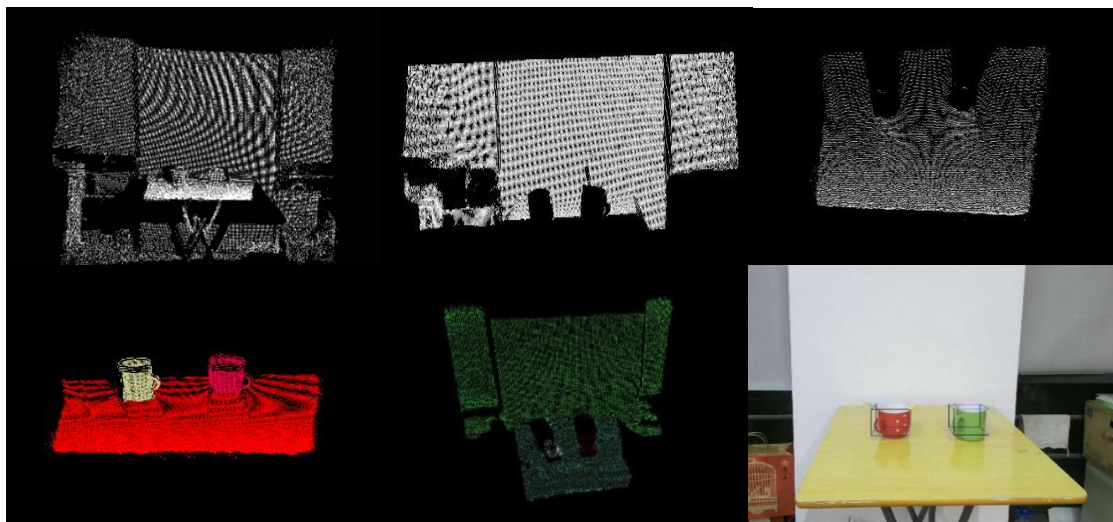


图 4 分割结果。

Fig. 4 Segmentation results

### 2.3 物体跟踪

从点云中分割出实景中的物体后, 我们还希望能够对各个物体进行实时跟踪, 即当实景中的物体发生移动时能够快速发现其新的位置。我们采用了粒子滤波的方法<sup>[7-8]</sup>对三维点云进行跟踪。其主要步骤如下。

(1) 初始化跟踪目标。除了大平面以外, 每一个分割得到的物体都被当做一个跟踪目标。计算该跟踪目标的点云分布特征。

(2) 分撒搜索粒子。在空间中分撒许多粒子去探测目标对象下一时刻可能的位置。最简单的分撒策略是在空间中均匀分布粒子, 但这样的效率不高。我们采用的策略是以当前目标物体的位置为中心, 按高斯分布进行粒子分撒。

(3) 判断目标位置。分撒出去的每个粒子将返回其所处区域的点云分布特征。依据原目标物体的

分布特征对当前的分布特征进行似然度计算, 得到当前粒子的权值。

(4) 粒子重采样。根据各个粒子返回的似然度权重, 分析得到下一时刻目标的位置, 并根据此似然度估计结果, 重新分撒粒子。

以上(2)、(3)、(4)步骤反复循环, 就可完成目标的动态跟踪。

在上述跟踪过程中, 我们依然利用了先验的大平面信息对计算进行加速。主要加速之处在于:

(1) 使用桌面和背景墙面的信息可以对分撒的粒子位置进行限制。我们限定粒子不可能被撒在桌面以下或者墙面之后, 由此可以减少分撒粒子数, 起到一定加速作用。

(2) 我们认为桌面和墙面等大平面是不会发生移动的, 因此在跟踪之前可以去掉这些大平面的点云。这样可以显著减少场景中的点云数量, 而留

下的点云都是属于物体的有效点云。当进行粒子区域的似然度估计时, 更为快速、准确。

在我们的实验场景中, 这种方式可以使跟踪效率提高 50%。具体见表 1 中的统计结果。

需要补充说明的是, 在进行多目标跟踪时, 为了加快效率, 我们采用了多线程策略, 一个线程跟踪一个目标。事实上, 这部分计算也可采用 CUDA 等并行架构在 GPU 中进行加速, 应该能够达到更高的效率。

### 3 系统构建

上一节论述了如何完成对现实场景的分割与跟踪, 而为了构建一个完整的增强现实系统, 还需要实现虚、实景象的融合以及虚、实物体交互的计算。我们借助 Unity3D 引擎来完成虚实融合与交互, 而将场景分割、跟踪的主要算法封装在动态链接库中; 通过 Unity3D 与外部动态链接库的连接来完成整个 AR 系统的搭建。

为使场景中的现实物体能与虚拟物体进行交互, 我们为每个现实物体都构建了代理几何体。该几何体是根据分割得到的物体点云集合计算得到的长方体包围盒。当进行虚实交互时, 此

几何体就代表现实物体与虚拟物体进行遮挡、碰撞等计算。

系统的结构和流程如图 5 所示。从图 5 中可见, 本文系统由两个主要部分组成, 一个是 Unity3D 部分, 另一个是 DLL 文件(动态链接库)部分。Unity3D 一端主要负责彩色图像的获取、界面的显示、生成代理几何体、渲染输出及进行虚实交互等, 而 DLL 端主要负责对场景深度信息的获取、对场景的分割与物体跟踪, 同时负责计算所分割物体的代理几何体并将数据传送给 Unity 端。

虚、实物体间的物理碰撞效果可以借助 Unity3D 的物理引擎来实现, 只需给代理几何体添加相应的物理属性即可实现。而要实现虚、实物体间的遮挡效果则需进行一定的渲染设置。我们采用了三遍绘制的方法来完成遮挡效果的模拟: 第一遍渲染所有的虚拟物体和代表现实物体的代理几何体, 但仅输出深度缓冲; 第二遍关闭深度比较和深度写入开关, 渲染视频背景; 第三遍打开深度比较, 关闭深度写入, 渲染除了代理几何体之外的所有物体。由此即可实现虚拟物体与现实物体的遮挡效果。

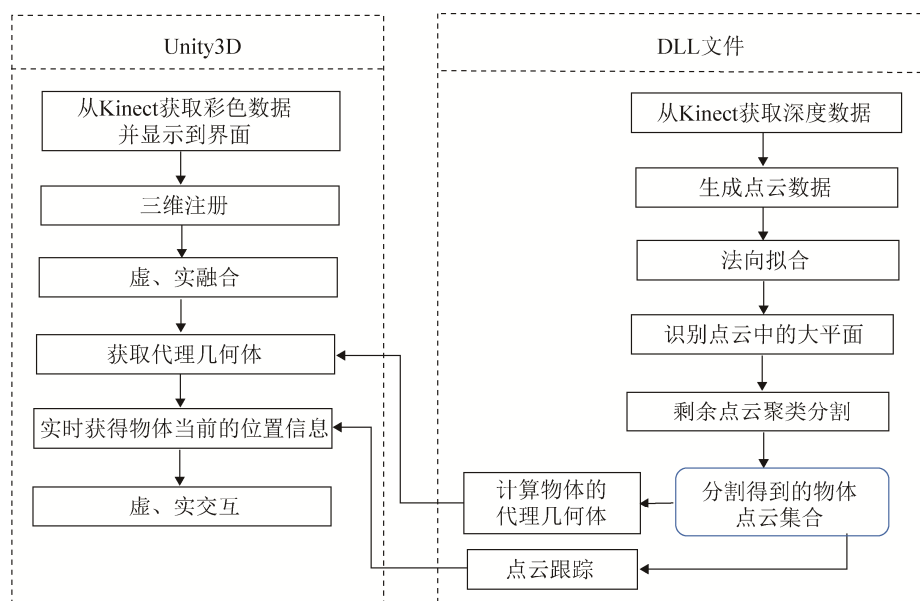


图 5 增强现实系统结构及计算流程

Fig. 5 Augmented reality system structure and calculation process



## 4 实验结果

本文工作的实验环境为: CPU i7-4790 3.6 GHZ, 8 GB 内存; GPU AMD Radeon R5 240, 1 GB 显存。深度摄像头采用了第二代 Kinect for Windows。编程环境为 Visual Studio 2012, 并借助了 PCL 点云库进行部分功能实现。

实验表明, 本文方法可以快速、准确地分割、跟踪桌面场景的物体, 实现虚实物体的交互。如图 6 显示了一个实验场景的结果, 其中左一图为 Unity 通过 Kinect 获取得到原始彩色图像; 右一、右二、左二分别显示了本文方法分割并计算得到的墙面、桌面和杯子的代理几何体(墙面和桌面的代理几何体为浅蓝色, 杯子的代理几何体为红色)。左三、右三、左四为遮挡实验结果图, 其中左三生成一个虚拟立方体图, 将立方体移动到杯子后, 立方体被遮挡(右三), 为了更清楚的观测到遮挡的情况, 我们对物体进行勾选后显示出对应的代理几何(左四); 右四为碰撞和跟踪的实验结果图, 我们向场景发射小球, 小球碰到墙面、桌面发生物理碰撞, 并做物理运动, 通过目标跟踪技术, 移动现实绿色杯子能将小球碰开。关于实验的更详细效果可见附带的两段视频。



图 6 实验结果。

Fig. 6 Experimental results

图 7 左图为物体动态跟踪的实验场景, 当用户移动桌面上物体时, 算法可以实时跟踪物体。图 7 右是另一个场景, 场景中摆放了更多的物体, 不断发射的虚拟小球与墙面、桌面及桌面上的积木物体发生碰撞、遮挡等效果。



图 7 另外两个实验场景

Fig. 7 Two other experimental scenarios

本文方法的运行效率见表 1 所示。其中是对图 7-右所示场景进行的效率统计。表 1 还将我们的方法与传统方法进行了对比分析。在对比方法 1 中, 其大平面识别没有采用法向拟合策略, 而是直接调用 RANSAC 方法进行识别, 需要 1000 次迭代; 而本文是在法向拟合之后进行引导性大平面识别, 仅需 10 次迭代, 其整体的识别速度提升了大约 40%。此外, 对比方法 1 中进行动态目标跟踪时没有利用先验的大平面信息进行加速, 我们较之效率提高了 50%。在对比方法 2 中, 没有进行大平面的识别和去除, 直接采用了欧式聚类分割的方法进行场景分割, 我们的方法较这种方式大约提高了 70%。效率提升按如下公式计算: (对比方法时间—本文方法时间)÷对比方法时间)。

## 5 结论

本文围绕增强现实系统中虚、实物体的交互问题展开研究, 提出了一种基于深度图像对场景进行分割及物体跟踪的解决方法, 并基于此方法成功实现了增强现实系统中虚、实物体间遮挡、碰撞等复杂交互效果。在场景分割及目标物体跟踪算法中, 本文提出了基于先验性场景知识, 首先识别并去除大平面区域, 再对剩余点云进行处理的策略, 显著提高了算法效率。本文方法可有效增强当前 AR 系统中虚、实物体的交互效果, 为开发更有创意的 AR 应用提供了可能, 具有实用价值。

本文方法在以下几个方面有待更进一步的研究和改进:

(1) 在目标跟踪算法中, 如果场景中的目标物体被遮挡一段时间, 会导致物体跟踪丢失。算法鲁棒性还有待提高。

(2) 当前我们的代理几何体只采用了简单的长方体包围盒, 在处理虚、实物体交互时可能会由于包围盒精度不够而发生问题, 例如, 图 6-右三图中虚拟物体长方体被杯子遮挡时在杯子边缘出现了问题。可以考虑对物体的点云集合进行模型重建, 构建更为精细的代理几何体。

(3) 本文的深度摄像头是固定不动的, 如何处理摄像头实时移动情况下的虚实融合问题, 将是我们下一步的工作。

## 参考文献:

- [1] Klein G, Murray D. Parallel Tracking and Mapping for Small AR Workspaces[C]//IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE, 2008: 1-10.
- [2] Tan W, Liu H, Dong Z, et al. Robust monocular SLAM in dynamic environments. In: Proceedings of IEEE International Symposium on Mixed and Augmented Reality, Adelaide, 2013: 209-218.
- [3] Raul Mur-Artal, Juan D. Tardos, ORB-SLAM2: an Open-Source SLAM System for onocular, Stereo and RGB-D Cameras[A]. Computer Vision and Pattern Recognition, 2016.
- [4] Schnabel R, Wahl R, Klein R. Efficient RANSAC for point-cloud shape detection[J]. Computer Graphics Forum, 2007, 26(2): 214-226.
- [5] Hoppe H, DeRose T, Duchamp T, et al. Surface Reconstruction from Unorganized Points[J]. Siggraph Computer Graphics, 26(2): 71-78.
- [6] 王醒策, 蔡建平, 武仲科, 等. 局部表面拟合的点云模型法向估计及重定向算法[J]. 计算机辅助设计与图形学学报, 2015(4): 614-620.  
Wang Xingce, Cai Jianping, Wu Zhongke, et al. Normal Estimation and Normal Orientation for Point Cloud Model Based on Improved Local Surface Fitting[J]. Journal of Computer-Aided Design & Computer Graphics, 2015(4): 614-620.
- [7] 张波. 基于粒子滤波的图像跟踪算法研究[D]. 上海, 上海交通大学, 2005.  
Zhang Bo. Research on Visual Tracking Algorithm Based on Particle Filter [D]. ShangHai JiaoTong University, 2005.
- [8] Brown A, Capson D W. A Framework for 3D Model Based Visual Tracking Using a GPU-Accelerated Particle Filter[J]. IEEE Transactions on Visualization and computer Graphics, 2011, 18(1): 68-80.