分类号：TP391.41　　单位代码：10335
密　级：无　　学　　号：10421038

浙江大学

# 博 士 学 位 论 文

中文论文题目：面向增强现实的实时三维跟踪

英文论文题目：**Real-Time 3D Tracking for Augmented Reality**

申请人姓名：董子龙
指导教师：鲍虎军
合作导师：
专业名称：计算机科学与技术
研究方向：计算机视觉增强
所在学院：计算机学院

**论文提交日期：**二〇一〇年七月

# 面向增强现实的实时三维跟踪

**论文作者签名：** _____

**指导教师签名：** _____

论文评阅人 1：查红彬　教 授　北京大学

评阅人 2：吴毅红　教 授　中国科学院自动化研究所

评阅人 3：陈小武　教 授　北京航空航天大学

评阅人 4：隐名评阅

评阅人 5：隐名评阅

答辩委员会主席：汪国昭　教 授　浙江大学理学院

委员 1：陈为　教 授　浙江大学计算机学院

委员 2：刘新国　教 授　浙江大学计算机学院

委员 3：童若锋　教 授　浙江大学计算机学院

委员 4：秦绪佳　教 授　浙江工业大学计算机学院

答辩日期：二〇一〇年九月

**Real-Time 3D Tracking for Augmented Reality**

**Author's signature:** _____

**Supervisor's signature:** _____

| | | | |
|---|---|---|---|
| Thesis reviewer 1: | Hongbin Zha | Professor | Peking University |
| Thesis reviewer 2: | Yihong Wu | Professor | Chinese Academy of Sciences |
| Thesis reviewer 3: | Xiaowu Chen | Professor | Beihang University |
| Thesis reviewer 4: | Anonymous | | |
| Thesis reviewer 5: | Anonymous | | |

Committee of oral defence:

| | | | |
|---|---|---|---|
| Chair: | Guozhao Wang | Professor | Zhejiang University |
| Members: | | | |
| | Wei Chen | Professor | Zhejiang University |
| | Xinguo Liu | Professor | Zhejiang University |
| | Ruofeng Tong | Professor | Zhejiang University |
| | Xujia Qin | Professor | Zhejiang University of Technology |

Date of oral defence: September 2010

# 浙江大学研究生学位论文独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得**浙江大学**或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：　　签字日期：　　年　　月　　日

# 学位论文版权使用授权书

本学位论文作者完全了解**浙江大学**有权保留并向国家有关部门或机构送交本论文的复印件和磁盘，允许论文被查阅和借阅。本人授权**浙江大学**可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名：　　导师签名：

签字日期：　　年　　月　　日　　签字日期：　　年　　月　　日

# 摘 要

随着计算机运算能力的不断增强，计算机视觉研究得到了持续的发展，在监控、检索、识别、导航、医疗、教育等领域的应用为人的视觉提供有效的补充，甚至在某些方面很好地替代了人的视觉。虚实混合作为计算机视觉的重要应用之一，是通过特殊的设备，将计算机产生的虚拟信息与现实环境无缝融合，给人们提供额外的信息，如说明文字、视频教程、三维动画等。

本文涉及的增强现实是虚实混合技术的一种，利用相关计算机视觉技术分析现实场景中的物体和环境特征，并在指定的位置绘制计算机生成的附属信息，帮助人们更好地理解场景。一般的增强现实系统，包括视频输入、特征分析、摄像机定位、虚实融合等模块，其中特征分析和摄像机定位是最核心的模块。离线增强现实已经在电影工业，视频广告中得到广泛应用，然而实时增强现实更多地处在实验阶段。本文主要研究实时增强现实的三维跟踪技术，即实时地恢复摄像机与场景之间的相对空间方位，内容包括多线程技术框架设计、图像特征分析、大规模场景的关键帧表达、和纯旋转相机下的双层分割方法。总体来说，本文希望能促进实时三维跟踪技术在增强现实中的应用，主要贡献在以下几个方面。

- 提出统一的实时增强现实系统框架。在关键技术充分模块化，模块接口标准化的基础上，将各种现实环境下的增强现实统一在一个多线程并行框架里，用户可以便捷地在此基础上开发新的增强现实应用，而且这个框架充分利用了多核机器的计算能力，使系统在适应各种复杂环境的情况下保证高效可靠的性能。

- 提出改进的基准标志系统。在一些桌面增强现实应用中，系统不能从自然场景中提取足够的特征定位摄像机，必须辅以基准标志。本文提出的基准标志是包围在黑色方框中的汉字图像。为了在复杂的光影下也能稳定地检测出标志，系统利用边缘信息检测标志的包围框。同时，本文将汉字的结构表达为汉字轮廓到边框的距离场，增加了汉字标志的可识别度。另一方面，传统的基准标志一般是黑白图案，从视觉上看很不美观，本文于是利用自然图像作为基准标志的补充。

- 提出基于关键帧的场景表达和快速选择候选关键帧方法。在大规模自然场景中，系统利用Structure-from-Motion技术从预处理视频序列中恢复场景的稀疏三维点云。由于大规模场景的特征过于丰富，特征匹配在时间和数量上的性能都会明显下降。本文通过贪婪优化方法，从输入预处理视频序列中自动选择一些关键帧，这些关键帧将包含比较稳定的，特征明显的三维特征点。三维跟踪通过图像识别算法，为输入图像选择相似的候选关键帧，然后只跟候选关键帧进行特征匹配。为了获得更稳定的跟踪结果，本文还利用极线约束，连续帧跟踪等方法匹配更多特征点。

- 提出摄像机纯旋转运动下，快速稳定地分离前景背景物体的方法。由于场景和摄像机运动的双重复杂性，场景层次分割是处理增强现实中的虚实遮挡的重要方法，同时也是非常难解决的问题。本文尝试解决在摄像机只有旋转运动情况下前背景之间的遮挡，这事实上是一个前背景分割问题。系统首先建立背景的全景图，然后将实时输入图像与背景全景图配准，估计背景信息，并利用图割算法进行分割。针对复杂背景和背景配准误差，本文结合过分割方法对背景全景图建立局部颜色模型，同时压制背景的颜色反差信息。系统得到精确的分割结果，并实现了一系列特殊的增强现实效果。

**关键词：实时，增强现实，三维跟踪，关键帧，双层分割**

# Abstract

Along with the explosive development of computing technology, the computer vision research has achieved sustained progress in the past decades, publicly distributed in surveillance, information retrieval, object recognition, navigation, medicine, education, etc. Mixed reality is an important application of computer vision, which is trying to exhibit the seamless mixture of real and virtual worlds, providing auxiliary information, such as description text, video tutorial, 3D animation, etc.

Augmented reality (AR) is a research direction of mixed reality. It analyzes the object or scene features using related computer vision techniques, and augments the reality with the computer-generated information at the specified locations, helping people to understand the scene better. Generally speaking, AR system contains the following components, video input, feature analysis, camera pose estimation, and rendering, among which feature analysis and camera pose estimation are the cores of the system. Offline AR has been widely used in movies and TVs, but real-time AR is still in the experimental stage. The dissertation will focus on the 3D tracking technique of real-time AR, i.e. recovering the camera pose at real-time rate. The main topics include the parallel computing, the analysis of image features, the keyframe-based representation of large-scale scene, and the bi-layer segmentation with rotating camera. All in all, the dissertation aims to promote the use of real-time 3D tracking in augmented reality, and contributes as follows.

- A unified real-time AR framework is presented. The whole AR system is split into several key modules, and the interfaces of modules are standardized to unify the AR systems for different environments in a parallel computing framework. The user can design new AR applications on the basis of the framework easily. The framework fully utilizes the computing power of multi-core CPU, and makes the AR systems always efficient and robust even in complex environments.

- A refined fiducial marker system is proposed. It's inevitable to use artificial fiducial markers

in the texture-less desktop environment. A Chinese character-based fiducial marker system is proposed to assist Chinese learning. In order to detect the fiducials under varying illuminations, the system utilizes the edge detection method to extract the bounding contours of markers. Further, the particular structure of Chinese character is captured by a distance field from the character contour to the bounding box, which improves the recognition. Meanwhile, to alleviate the unfriendly black and white appearance of traditional fiducials, the natural images can be used as special markers.

- A keyframe-based scene representation with a fast candidate keyframe recognition method is proposed. In large-scale natural scene, the structure-from-motion technique is used to reconstruct the 3D point cloud of the scene from the input sequence. The performance of feature matching drops quickly because of the abundant features in the large-scale scene. With the greedy optimization, a set of keyframe is selected from the input sequence, which contains as many stable feature points as possible. During real-time 3D tracking, the online image is matched to the candidate keyframes, which are determined by fast image recognition method. To achieve better tracking result, epipolar geometry constrained feature matching and temporal information are used to get more feature matches.

- A bi-layer segmentation method with rotating camera is proposed. The scene layer segmentation method is important for handling occlusions in AR system, as well very difficult due to the complex movements of the scene and camera. A simple case of rotating camera before a static background is considered here, which is equally a bi-layer segmentation problem. The background panorama is constructed beforehand, then the online image is registered to the panorama and the corresponding background is recovered. The Graphcut is used to solve the segmentation. To handle the complex panorama and background registration error, a local color model combined with over-segmentation is proposed. With the help of the background contrast attenuation, the experiments show precise segmentation results. Some interesting augmented effects are also implemented.

**Keywords: real-time, augmented reality, 3D tracking, keyframe, bi-layer segmentation**

# 目 次

# 图 目 录

# 表 目 录

# 算 法 目 录

# 中英对译表

Appearance Vector 外观向量, 41

Augmented Reality(AR) 增强现实, 1

Augmented Virtuality(AV) 增强虚拟, 1

Background Subtraction 背景消减, 20

Bag of Words 词汇包, 41

Blob 色块, 10

Box Filter 箱式滤波器, 11

Bundle Adjustment 集束调整, 9

Central Projection 中心投影, 5

Closeloop Detection 回路检测, 19

Contour 轮廓, 26

Contrast 颜色反差, 20

Control Law 控制律, 17

Corner 角点, 10

Data Term 数据项, 20

Dynamic Programming 动态规划, 44

Epipolar Line 极线, 8

Epipolar 对极几何, 7

Extended Kalman Filter 扩展卡尔曼滤波, 18

External Parameters 外部参数, 6

Fiducial Marker 基准标志, 13

Finite Camera 有限相机, 5

Frame-to-Frame-Tracking 连续跟踪, 13

Fundamental Matrix 基础矩阵, 7

Gaussian Mixture Models 高斯混合模型, 20

Genetic Programming 遗传算法, 12

Graph Cut 图割, 20

Hashing 哈希操作, 13

Head Mounted Display 头盔显示器, 77

Homography 单应矩阵, 16

Infinite Camera 无限相机, 5

Interest Point 兴趣点, 9, 10

Internal Parameters 内部参数, 6

Kalman Filter 卡尔曼滤波, 16

Landmark 标志点, 19

Lie Algebra 李群代数, 17

Local Descriptor 局部描述量, 9

Local Feature Point 局部特征点, 9

Mixed Reality(MR) 混合现实, 1

Neural Network 神经网络, 12

Ordinal Descriptor 序数描述量, 12

Particle Filter 粒子滤波, 17

Pinhole Camera 针孔相机, 5

Point Cloud 点云, 6

Pose 方位, 4

Projection Center 投影中心, 5

Projection Kernels 投影核, 9

Projection Plane 投影平面, 5

Projective Camera 射影相机, 5

Random Tree 随机树, 13

Relocalisation 重定位, 19

Response 反应值, 11

Robust Estimation 鲁棒估计, 15

Round Robin 循环, 17

Saliency 显著度, 42

Scale 尺度, 9

Smoothment Term 平滑项, 20

State-Obervation Model 状态-观测模型, 18

Synthetic Views 合成视图, 16

Temporal Regularization 时序调整, 16

Tracking-by-Detection 匹配跟踪, 13

Two-View Geometry 双视几何, 7

Virtual Visual Servoing 虚拟视觉伺服, 17

Visual Vocabulary 视觉词汇表, 41

Vocabulary Tree 词汇树, 41

# 1 绪论

“画意能达万言”，人对世界的感知，如大小、明暗、颜色、动静，以及对机体生存具有重要意义的诸多信息，有80%以上经视觉系统获得，视觉是人最重要的感觉。然而由于知识背景等因素的差别，即使同一事物的视觉信息，不同的人也会产生千差万别的理解，比如面对名画，如果对于画家生平、创作年代、绘画技巧有一定程度的了解，自然可以发现画作的美丽细节，如果对其一无所知，那么我们只能简单地评价好看或不好看，甚至连这样的评价也很模糊。在这种情况下，如果可以通过某些手段给人们提供额外的信息，比如文字说明，视频介绍等，可以帮助人们理解画作。再比如在机械装配修理过程中，机械师看到的只是机器的外观，不同熟练程度的机械师完成任务的时间和效果都不一样，如果将以往的修理记录（文字或视频）、机器的三维模型与机器无缝地融合在一起，可以给机械师提供很多参考，帮助他们更快捷更有效地完成任务。传统的做法是利用固定的纸牌、帮助手册、显示器等提供附加说明，这种方法更新信息麻烦，而且对于不需要这些信息的人来说反而是累赘。此时，我们可以利用计算机产生的虚拟信息来替换现实的附加内容，通过计算机视觉技术的支持，这些虚拟信息可以和现实环境无缝对接。

计算机视觉技术的目的是完全模拟人的视觉，包括对视觉信息的处理和理解。随着现代计算机的运算能力不断增强，计算机视觉研究也持续发展，不仅基础理论研究在各个方向取得突破，而且在实际应用中也出现了许多成熟的系统。在监控、检索、识别、导航、医疗、教育、培训等领域，基于计算机视觉的应用为人类的视觉提供有效的补充，甚至在某些方面很好地替代了人的劳动。在计算机视觉理解现实世界的基础上，计算机可以自动地在适当的时间地点绘制虚拟信息，将现实场景和虚拟信息无缝地连接在一起，这就是**混合现实**。利用混合现实系统，用户通过特殊的设备，可以同时接受虚拟和现实信息，获得新的感知体验。

混合现实按照对虚景或实景的侧重点不同，又可分为**增强现实**和**增强虚拟**，如图1.1。现实环境是我们最熟悉的世界，即用肉眼观察到的所有信息，另一端的虚拟环境是完全虚拟的世界，对于人类来说，唯一需要提供的就是精神智能，如在《The Matrix》中

的Matrix[①]。现有的计算机模拟技术在模拟独立的、简化的、小规模的自然现象上已经有了许多真假难辨的效果，尤其是在最新的3D电影《Avatar》[②]中，那个完全虚拟的潘多拉星球和蓝色纳美种族塑造地非常真实。但是这些效果仅存在于后期电影制作中，每一帧画面的产生都要花费并行计算机集群几分钟到几个小时。显然，人们不可能为一个虚拟画面等待那么长时间，而且其中的人物动作、表情等元素仍然从现实世界捕获，这距离完全的虚拟环境还很远。退而求其次，研究者试图在虚拟与现实之间找一个平衡点（混合现实），提出了增强现实和增强虚拟。增强现实是以现实世界为主，以计算机视觉技术为基础，在现实环境中放置虚拟信息，帮助用户更好地理解环境；反之，增强虚拟是以虚拟环境为主。


图 1.1 虚拟和现实的联系。从现实环境到虚拟环境的过渡之间，是增强现实和增强虚拟。

按照运行方式的不同，混合现实又可以分为实时混合现实和后期混合现实，见表1.1。后期混合现实主要应用于影视制作、广告行业，更注重虚拟和现实信息的无缝融合，不惜时间成本代价以保证真实感。实时混合现实则牺牲一定的真实感，确保在最短反应时间内为用户提供需要的信息。

表 1.1 实时和后期混合现实。

| | 实时 | 后期 |
|---|---|---|
| 增强现实 | 实时增强现实 | 后期增强现实 |
| 增强虚拟 | 实时增强虚拟 | 后期增强虚拟 |

一般来说，实时增强现实是混合现实研究的重点[1,2]：一、在众多现实需求中，现实

[①]http://www.thematrix.com

[②]http://www.avatarmovie.com

环境还是用户视觉的主导内容；二、实时增强现实以现实为主，降低了对虚拟信息的模拟内容和真实感要求，相对容易实现。由于计算机可生成的虚拟信息十分丰富，生成、更新、维护便捷简单，实时增强现实很容易和其它计算机视觉技术结合，这给实际应用带来许多可能性③。

- 医疗领域：将各种虚拟人体模型与人体叠加，帮助医疗人员精确定位组织器官[3]。
- 军事领域：根据物体识别，位置识别的结果，获得相应的地理数据等重要军事数据，引导军事打击。
- 古迹复原：利用三维建模技术恢复古迹或者遗产的原貌模型，以增强现实的方式提供给参观者[4]。
- 工业维修：将多种辅助信息显示给工程师，可以包括设备结构、设计图纸、运转情况模拟、维修历史等。
- 网络视频通讯：利用人脸跟踪技术，在通话者的面部实时叠加虚拟物体，如帽子、眼镜等；或者将通话者与背景分离，实时替换使用各种特殊背景。
- 电视转播：在转播电视节目时，可以加入虚拟主持人，可以在舞台上生成虚拟环境，尤其在转播体育赛事时，可以通过增强现实提供战术分析，球员信息，规则图示等，帮助观众更好地欣赏比赛。
- 娱乐、游戏：增强现实游戏主要是可以让游戏者在现实环境中，与虚拟物体进行互动对抗，增加游戏的真实性和趣味性。
- 旅游、展览：人们在浏览、参观的同时，可以观看路线、展品的相关数据资料。
- 建设规划：将规划图纸以三维模型等形式叠加到真实场景中，预览规划效果。

随着计算机的普及和计算能力的提升，上述的增强现实应用已逐渐走出理论阶段，从Sensorama④到SixthSense[5]，以及iPhone手机平台应用acrossair⑤，从机械式扩展到计算

③http://baike.baidu.com/view/104668.htm

④http://en.wikipedia.org/wiki/Sensorama

⑤http://www.acrossair.com/acrossair_app_augmented_reality_nearesttube_london_for_iPhone_3GS.htm

机，直到掌上移动平台，越来越贴近人们的生活。图1.2展示了增强现实中常见的输入输出设备，如摄像头、GPS等，以及相关的计算机视觉和计算机图形学技术。

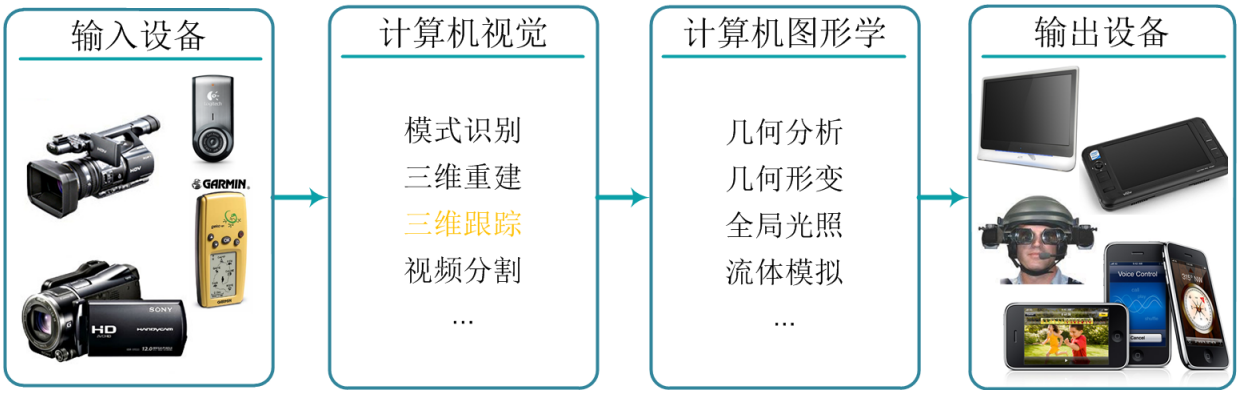



图 1.2 增强现实中常见的输入输出设备。输入主要是视频采集设备，也包括GPS、陀螺仪等硬件定位设备，输出设备主要是显示器、头盔、投影仪等。计算机视觉技术分析输入信息，计算机图形技术将分析结果作用于虚实融合。三维跟踪是增强现实的核心技术。

增强现实的目标是将虚拟信息与输入的现实场景无缝地结合在一起，使用户虚实莫辨，为了达到这个目标，需要具备四个条件：

1. 几何一致，虚拟物体在场景中摆放的位置要保持一致性，不漂移，不抖动，而且需要处理虚拟物体和现实物体之间的遮挡关系；
2. 模型真实，如果虚拟物体是三维模型，则要保证模型形状、纹理、材质的真实感；
3. 光照一致，虚拟物体的光照条件要与现实环境一致，由于现实光照的复杂性，这个要求往往很难达到；
4. 颜色一致，如果输入是视频采集设备，捕获的图像难免会受曝光、白平衡、噪声等的影响，因此绘制的虚拟物体也需要模拟采集的硬件条件。

其中，几何一致是增强现实的基本问题。当输入设备（摄像头等）在场景中运动时，我们需要计算出其相对场景的**方位**信息，以便在绘制虚拟物体时设置正确的虚拟摄像机参数，使虚拟物体在输出设备中的成像与输入设备捕获的场景（非穿透式显示设备）或人眼看到的场景（穿透式显示设备）保持一致性[6]。这个求解输入设备方位信息的过程就是增强现实的实时三维跟踪技术，也是本文的主要研究方向。本文提出了实时三维跟踪的并行

框架，并在此基础上探讨不同应用情况下的具体实现，包括大规模场景的实时三维跟踪问题，还提出利用视频分割技术，初步解决增强现实在摄像头纯旋转运动时的遮挡问题。本文主要涉及相机模型，特征分析，参数估计等计算机视觉方法，本章剩余内容将简单介绍这些基本技术，以及现有的实时三维跟踪系统。

## 1.1 摄像机模型

由于本文的目的是恢复摄像机在场景中的方位信息，首先就要为摄像机选择一个合理的参数化模型。增强现实主要模拟人眼观察世界的方式，因此本文采用常见的**射影相机**模型（图1.3）。射影相机模型用一个$3 \times 4$的矩阵表达从三维世界到二维图像的映射，这是一种**中心投影**相机，如果**投影中心**到**投影平面**的距离有限，则称为**有限相机**，反之，则是**无限相机**。有限相机的最简单的形式是**针孔相机**模型，由于现在的相机制造工艺基本上能制造出比较符合标准的成像芯片，大部分现有摄像机也基本满足这个模型，本文的系统将利用针孔相机模型来求解场景中摄像机方位参数。

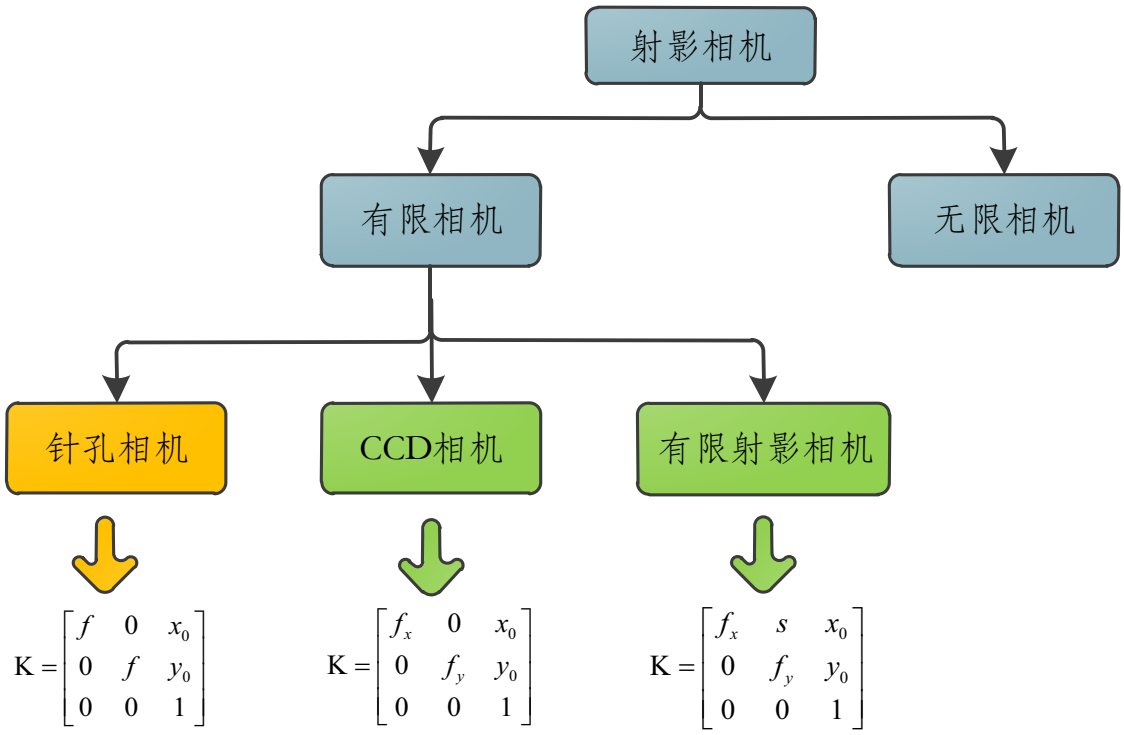



图 1.3 射影相机模型。射影相机模型用一个$3 \times 4$的矩阵表达从三维世界到二维图像的映射，按照投影平面到投影中心的距离可分为有限相机和无限相机，针孔相机模型是有限相机的一种。

针孔相机模型下的投影矩阵为：

$$x = K[I|0]X'. \tag{1.1}$$

其中，$x = [u, v, 1]^T$是图像坐标，$X' = [X, Y, Z, 1]^T$是在相机坐标系中的三维点齐次坐标，$K$即是图1.3中针孔相机对应的$K$。在针孔相机模型中，$x$轴和$y$轴方向上的焦距是相等的，$f_x = f_y = f$，即纵横比为1，而且轴向倾斜率$s = 0$。这些参数和相机在场景中所处的位置无关，因此被称为**内部参数**（简称内参），与其对应的是相机**外部参数**（简称外参），即相机在空间的方位信息。在一般的增强现实系统中，相机的内参是固定的，通过其它工具预先标定好⑥，系统只需要实时恢复相机的外参。这种做法可以简化计算，避免求解的歧义，也符合一般增强现实应用的需求。

摄像机在场景中的外参指摄像机的局部坐标系与场景的世界坐标系之间的变换关系。这里存在两个不同的坐标系统：场景对应的世界坐标系，和摄像机对应的相机坐标系，如图1.4，它们之间的变换由$R, t$表达。设摄像机的中心在世界坐标系中的位置是$C$。世界坐标系中的三维点$X = [X, Y, Z, 1]^T$，可用以下变换转换到相机坐标系：

$$X' = TX = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} X = \begin{bmatrix} R & -RC \\ 0 & 1 \end{bmatrix} X. \tag{1.2}$$

其中，$R$是欧拉空间中$3 \times 3$的旋转矩阵，也是归一化的正交矩阵，即$R^t = R^{-1}$，$det(R) = 1$，$R$可以表达成与坐标轴对应的三个欧拉旋转角或者四元数。平移向量$t$和$C$则是三维向量，一般情况下，$C$不显式表达。

将公式1.1和公式1.2连接在一起，我们得到投影公式1.3。可以观察到，在投影矩阵$P$的两侧，分别是点的三维坐标和图像坐标，对应环境的三维模型和输入图像。环境三维模型的表达方式可以是：三维**点云**，CAD模型，体模型等；针对不同的模型表达，三维跟踪需要在输入图像上检测相应的特征，如局部特征点，物体边缘[7–9]等。虽然与特征点相比，图像的边缘对光照变化，运动模糊等容忍度较高，但是边的特征信息较少，难以匹配，在目前的三维跟踪系统中局部特征点还是最重要的图像特征。

$$x = PX = K[I|0]TX. \tag{1.3}$$

⑥http://www.vision.caltech.edu/bouguetj/calib_doc/，如果相机成像有径向畸变，也可以预先标定。

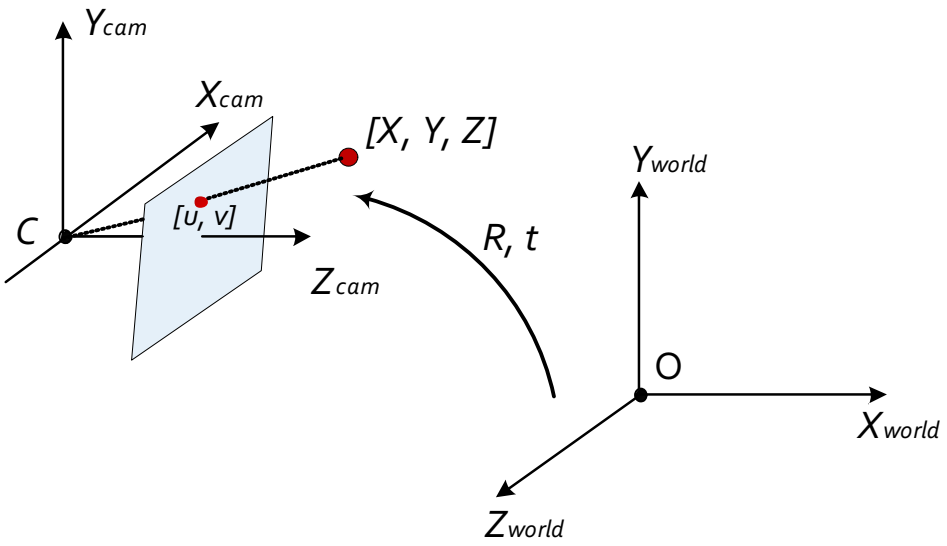



图 1.4 世界坐标系到相机坐标系的变换。变换由旋转矩阵$R$和平移向量$t$组成。

### 1.1.1 双视几何

**双视几何**涉及两个透视相机之间的几何关系，双视之间的特征点对应是双视几何的主要内容之一：**对极几何**及其代数表达**基础矩阵**。

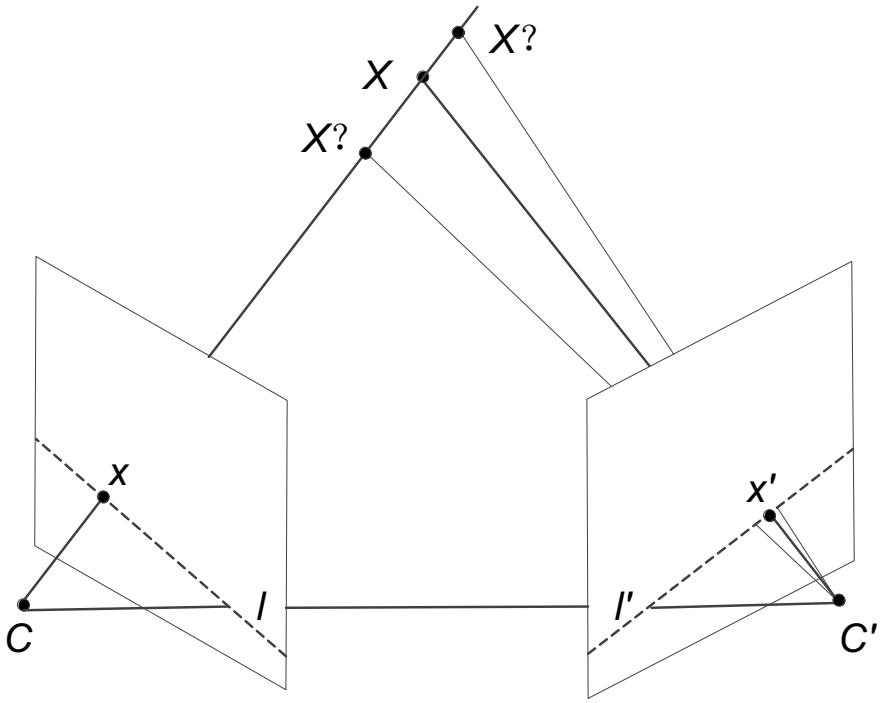



图 1.5 对极几何。

假设$X$是三维空间的一个点，在两个透视相机$C$和$C'$上的图像投影分别为$x$和$x'$。从图1.5可以看出，$x, x', C, C', X$在同一平面上。连接$x$和$X$的直线在$C'$上的投影是直线$l'$，而且$x'$在$l'$。反过来，$x$在直线投影$l$上。这就是对极几何的基本概念，$l$和$l'$都被称为**极线**。如果给定$C$中的$x$，要在$C'$中搜索对应点$x'$，那么就不必在整幅图像搜索，只需要在极线上搜索，极大地减少了计算量，也提升了搜索的准确性。

基础矩阵$F$是对极几何的代数表达形式，是一个秩为二的$3 \times 3$矩阵，满足：

$$x'^T F x = 0 \tag{1.4}$$

于是，相应的极线可以表达为：$l' = Fx$，$l = F^T x'$。由于每一对对应点可以根据公式1.4列一个关于$F$的线性方程，如果知道两副图像上的一些对应点，可以反求出基础矩阵：

$$u'uF_{11} + u'vF_{12} + u'F_{12} + v'uF_{21} + v'vF_{22} + v'F_{23} + uF_{31} + vF_{32} + F_{33} = 0. \tag{1.5}$$

其中，$x = [u, v]^T, x' = [u', v']^T$，$F_{ij}$是$F$第$i$行第$j$列的元素。$F$的尺度无法确定，可以添加一个约束令$\|f\| = 1$，$f = [F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33}]^T$。只需要最少八个对应点就可以线性地求出$F$，如果有多于八个对应点，可以通过最小二乘法得到$F$。现实应用中的对应点通常包含一些错误的对应，因此$F$的求解一般需要结合RANSAC方法[10]。

### 1.1.2 Structure from Motion技术

SfM（Structure from Motion）是计算机视觉的经典问题，指从输入的一系列图像中恢复场景的三维结构和图像对应的相机参数。经过研究者多年的努力，SfM已经有比较成熟的解决方案，现有许多商业软件⑦，还有免费的开源软件⑧，主要包括以下步骤：

1. 从输入图像中提取特征点，并在图像之间匹配特征点；
2. 自动选择在多帧图像中稳定出现的特征点，并确定关键帧；
3. 选择最佳的三个关键帧进行场景三维的初始化，并及时自定标完成从射影空间到欧拉空间的转换；

⑦Boujou, http://www.2d3.com/

⑧Bundler, http://phototour.cs.washington.edu/bundler/或ACTS, http://www.zjucvg.net/acts/acts.html

4. 求解其它关键帧，在欧氏空间渐进式地求解所有关键帧；

5. 根据欧氏重建的三维结构，求解剩余帧；

6. 用**集束调整**对整个序列的结构和相机参数进一步优化。

SfM是非常耗时的离线过程，现在主要应用于影视和广告制作、三维建模等。实时三维跟踪技术使用到的计算机视觉方法与SfM基本类似，只是降低求解的精度，换取更高的计算效率，可以看作是SfM的在线版本（第1.5节）。

## 1.2 局部特征点

**局部特征点**指场景中包含可识别纹理信息的局部图像面片；由面片的位置和**局部描述量**组成，位置表达为图像上的像素级或亚像素级坐标，局部描述量根据局部窗口内的图像信息计算得到。局部特征点的窗口可以是方形，圆形，椭圆；其**尺度**和形状有许多方法可以计算，但对同一个特征点都要保证总是包围同样的图像区域。局部描述量是一个多维向量，可以表示亮度、梯度、序数等统计信息[11,12]或**投影核**[13]。局部特征点的提取通常分成三个步骤（图1.6）：1) 检测**兴趣点**位置；2) 计算局部窗口；3) 生成局部描述量。这三个步骤有时候并不是严格独立串行，比如SIFT[14]特征点的局部窗口大小在检测阶段确定，而其形状则是由描述量计算方法决定。

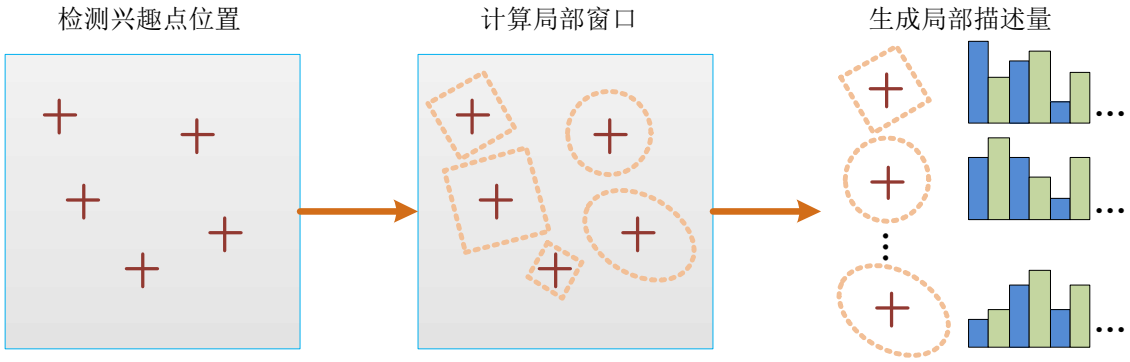



图 1.6 提取局部特征点的三个步骤。

一般的局部特征点提取方法都可以从场景检测到许多特征，其中一些特征是由图像噪声产生，还有一些是光照造成的阴影，在实际的应用中为了保持系统的效率，总是依据一定的规则选择其中比较稳定的一部分特征点，比如SIFT[14]选择DoG（Difference of

Gaussians）大于某阈值的特征点。因此，一个好的局部特征点应该有明显的特征，并在场景中持续存在，不受摄像机运动、光照变化等干扰；而好的局部特征点提取方法的任务就是把好的特征点检测出来，必须具有以下特点[12,15,16]：

- 重复性，如果一个特征点出现在多幅图像内，不论摄像机运动、光照变化如何，都应该被检测出来；
- 尺度不变，相机与特征点之间的距离变化，同一个特征点的大小也会发生变化，因此需要对局部窗口的尺度做出相应调整，保持局部窗口所涵盖的图像区域不变；
- 旋转不变，当摄像机沿着主轴旋转时，特征点局部窗口内的图像也会相应旋转，在这种情况下，特征点的描述量要保持不变；
- 仿射不变，当摄像机自由运动时，特征点的局部图像发生仿射变换，会发生扭曲，需要对局部图像进行矫正；
- 光照不变，由于光源变化、摄像机移动，图像的光照总会产生变化，特征点的描述量要保持一致；
- 描述量区分度高，如果特征点具有不变性，同一个特征点的描述量基本保持一致，那么不同特征点的描述量区分度要高，才能把特征点区分开；
- 高效率，对于实时系统来说，特征点的检测必须实时进行，甚至应该是高于实时要求。

其中，重复性和尺度选择体现在兴趣点的检测过程，而旋转、仿射不变一般是在确定特征点位置之后的一个独立过程，光照不变主要体现在局部描述量的计算方法上。

### 1.2.1 检测兴趣点位置

局部特征点的位置常常在图像的**角点**上[17,18]，也可能在**色块**的中间[14,19,20]，这些点称为**兴趣点**。现有许多检测兴趣点位置的方法，这里只简单介绍实时三维跟踪系统中常用的两类兴趣点，详细的分类参阅Rosten等的文章[12,18]。

第一种兴趣点检测是基于图像偏导，输入图像相对应的灰度图像首先进行相应的偏导运算（常见的偏导算子见表1.2），然后在图像平面或尺度空间上检测局部极值。

表 1.2 常用的偏导算子。$\mathbf{x} = (x, y)$是图像坐标，$G(\cdot)$是高斯函数，$L_x, L_y$是高斯一阶导数，$L_{xx}, L_{yy}, L_{xy}$是高斯二阶偏导。$s$表示高斯函数的方差，即特征点的尺度。$\tilde{s}$是图像高斯平滑函数的方差。

| 偏导算子 | 计算公式 |
|---|---|
| LoG（Laplacian of Gaussian）[21] | $s^2(L_{xx}(\mathbf{x}, s) + L_{yy}(\mathbf{x}, s))$ |
| Dog（Difference of Gaussian）[14] | $I(\mathbf{x}) * G(s_{n-1}) - I(\mathbf{x}) * G(s_n)$ |
| Harris函数[17,22] | $det(\mathbf{C}) - \alpha * trace^2(\mathbf{C})$, with $\mathbf{C}(\mathbf{x}, s, \tilde{s}) = s^2 G(\mathbf{x}, \tilde{s}) * \begin{bmatrix} L_x^2(\mathbf{x}, \mathbf{s}) & L_x L_y(\mathbf{x}, \mathbf{s}) \\ L_x L_y(\mathbf{x}, \mathbf{s}) & L_y^2(\mathbf{x}, \mathbf{s}) \end{bmatrix}$ |
| Hessian矩阵[20] | $\begin{bmatrix} L_{xx}(\mathbf{x}, \mathbf{s}) & L_{xy}(\mathbf{x}, \mathbf{s}) \\ L_{xy}(\mathbf{x}, \mathbf{s}) & L_{yy}(\mathbf{x}, \mathbf{s}) \end{bmatrix}$ |

Harris函数最早由Harris等在1988年提出[17]，他们在图像平面里利用局部自相关矩阵计算角点的**反应值**，然后选择反应值强的点作为角点。Shi等[22]针对KLT跟踪算法[23]，选择自相关矩阵最小的特征值作为反应值，认为反应值大的点是好的兴趣点。以上方法只在图像平面检测反应值极值点，所用高斯导数的方差是固定的，没有尺度空间的运算，因此只能确定兴趣点的位置，而不包含尺度信息。Bay等人[20]结合Hessian矩阵和尺度空间[24]运算，并用**箱式滤波器**近似高斯二阶导数，可以快速地同时检测兴趣点的位置和尺度。其它的偏导算子也可以用来计算图像的尺度表示，如Lindeberg的LoG（Laplacian of Gaussian）[21]和Lowe的DoG[14]。这些算子还可以组合使用，Mikolajczyk等提出Harris-Laplacian函数[25]，先在图像平面上检测Harris角点，然后利用LoG得到特征点的尺度。Mikolajczyk等比较了各个偏导算子的检测性能[25]：Hessian>LoG>DoG>Harris。虽然高斯二阶导数的检测性能更高，但是计算量也相应的增加，如要在实时系统中应用，需要做出一定的近似。事实上，DoG[14]就是LoG的近似，SURF[20]则是Hessian矩阵的近似。

另一种兴趣点检测方法是基于机器学习理论，先从训练样本的信息学习兴趣点分类器，然后用分类器在其它图像中检测特征。机器学习方法的优点是计算量小，效率高，但是精度不高，对实时系统来说，牺牲部分精度换取时间是可行的。

Dias等[26]和Kumar等[27]训练**神经网络**模型来识别角点，他们的训练样本为在图片上自动随机生成的边的交点。Kienzle等[28]的训练样本是跟踪用户眼睛的凝视点来获得，并用SVM（Support Vector Machine）训练。Trujillo等[29]从另一个角度考虑，将兴趣点的重复度和分布均匀度作为**遗传算法**的适应度函数，自动优化产生兴趣点检测算子。Rosten等提出的FAST[18]算法用中心像素和周围像素的亮度比较来检测特征点，虽然方法简单，但是效率很高，重复性能也很好，已经在低端计算平台上得到应用。

### 1.2.2 计算局部窗口

局部特征点的局部窗口由算法的需求确定，最简单的是方形和圆形，窗口大小由兴趣点检测过程的尺度运算确定，还需要确定旋转方向。SIFT[14]统计局部窗口内的图像梯度方向和强度的直方图，取最大的几个方向作为旋转方向，得到比较稳定的结果。这样得到的局部窗口还仅仅是尺度，旋转不变，不能容忍大的视角变化，如果要满足仿射不变，需要更复杂的方法[30]。

### 1.2.3 生成局部描述量

现在最常用的局部描述量都是基于局部梯度的统计信息，如SIFT[14,31]，GLOH(Gradient Location and Orientation Histogram)[32]，SURF[20]等。Heikkila等[33]结合SIFT和LBP（Local Binary Patterns）[34]的优点，提出CS-LBP（Center-Symmetric LBP）描述量。CS-LBP使用与SIFT类似的网格结构，但是在每一个网格里面计算CS-LBP。由于CS-LBP只涉及亮度比较操作，计算效率比SIFT高，而且性能并没有损失。Toews等[35]用**序数描述量**处理SIFT和GLOH等高维向量：对高维向量的维度进行排序，每一个维度的值用其在排序结果中的序号表示。实验结果表明，SIFT-Rank的匹配结果得到明显的改进。序号描述量只要求维度的单调性，以非参数化的方式处理图像的非线性变化，这为设计新的描述量方法提供了有用的参考。同样的，也有一些基于机器学习方法的描述量[36,37]。

### 1.2.4 特征匹配

计算特征局部描述量之后，还会涉及特征匹配，即给定一个特征点，在另一堆特征点中搜索与之距离最接近的点。显然，我们必须先定义特征点之间的距离。简单的$L_2$欧拉距离在大多数情况下都能获得很好的结果，当然也有更好的定义，如Mahalanobis距

离，EMD（Earth Mover's Distance）[38,39]。由于描述量的维度通常比较高，这其实是高维向量空间的KNN（K-Nearest Neighbor）搜索问题，如果只是利用暴力搜索，其时间复杂度太高，根本无法实际应用，但是通过构造特殊的树结构，如KD-Tree[40]，Orthogonal Tree[41]等，可以大幅度降低复杂度。如果搜索结果允许一定的误差，还可以用ANN（Approximately Nearest Neighbor）方法进一步加速[42,43]。Gionis等[44]提出用**哈希操作**来缩小搜索范围，每一次搜索都以固定的可能性返回正确的ANN匹配。Arya等[45]基于KD-Tree和Box-Decomposition-Tree实现了最优的ANN方法，并提供了源代码⑨。

还有研究者将特征匹配当作特征分类问题[46]来解决，他们的特征描述量类似LBP，但是计算方法不同。他们在特征点局部窗口内随机生成许多测试点对，然后比较点对的亮度来获取描述量，而这些测试点对也用于建立分类**随机树**[47,48]。这种基于机器学习的方法特点是描述量计算和匹配速度都非常快，很适合在低端计算环境下运行，但是需要一个离线的分类器训练过程，而且分类器通常会占用比较大的内存空间，产生的误匹配相对来说比较多，不容易保证求解的稳定性。

值得一提的是，上述的通过特征匹配来跟踪特征点信息的方法可称为**匹配跟踪**[47]，还有一种方法是**连续跟踪**。如果在两帧比较接近的图像之间进行特征匹配，或者两幅图像对应的相机参数可以估计，那么可以利用效率更高的连续跟踪方法匹配特征点，如KLT[23]，Active Matching[49]。匹配跟踪的特点是计算量或内存消耗大，但是对于快速运动不敏感；连续跟踪则反之，而且一旦跟踪失败，就无法自动恢复，如果把两者结合起来，可以得到更稳定的跟踪结果[50]。

## 1.3 实时三维跟踪

根据1.1节定义的相机模型，在增强现实中，摄像机每捕获一帧图像，系统都需要提取并匹配图像特征，然后求解当前摄像机对应的$R,t$。根据场景模型的表达方式，以及是否预先构建场景模型[51]，实时三维跟踪的分类见图1.7。基于模型的三维跟踪是在已知场景模型的情形下进行，场景模型可以是人工设定的**基准标志**，利用SfM恢复的点云，或者

⑨http://www.cs.umd.edu/~mount/ANN/

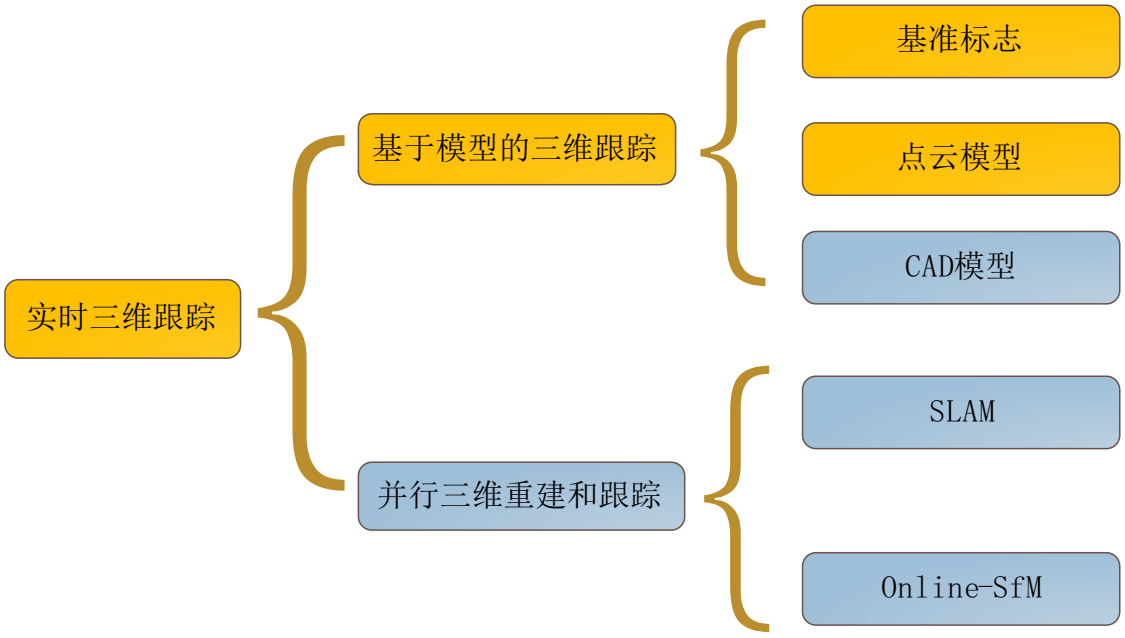



图 1.7 实时三维跟踪的分类。按照场景三维是否已知，分为基于模型的三维跟踪和并行三维重建和跟踪。

用建模软件（如Maya[10]，AutoCAD[11]等）创建的CAD模型。并行三维重建和跟踪是在跟踪过程中，同时恢复环境的三维信息，按照求解模型分为SLAM（Simultaneous Localization and Mapping ）和Online-SfM。

## 1.4 基于模型的三维跟踪

如果场景的三维模型预先建立好，并且不再改变，我们可以将图像中的二维特征与三维模型匹配，然后计算摄像机外参。这些方法通常包括两个步骤：预处理和实时三维跟踪[52]。于是，实时三维跟踪问题就变成计算机视觉中基本的二维-三维方位估计问题，即在相机内参已知的条件下，通过三维实体及其对应的图像投影信息恢复相机外参，其中最常见的是点的对应关系。

在针孔相机模型下，给定一系列点的三维坐标及其图像投影坐标，确定相机在世界坐标系中的方位，就是PnP（Perspective-n-Point）问题[53]。如果点的个数$n \leq 2$，相机参

[10] http://usa.autodesk.com/adsk/servlet/pc/index?id=13577897&siteID=123112

[11] http://usa.autodesk.com/adsk/servlet/pc/index?id=13779270&siteID=123112

数有无穷解，因此相关工作只考虑$n \geq 3$的情况。自Fischler等[10]的开创性工作以来，对于$3 \leq n \leq 5$的情形，研究者提出许多非迭代方法求解[54–59]。从理论上说，P3P可以得到一到四个可能解，至少需要加一个点才能确定唯一解。如果四个点共面，P4P可以得到唯一解，但是在现实情况中，还是可能出现歧义[60]，而P5P一般情况下都可以得到唯一解[61]。

虽然在$3 \leq n \leq 5$时，可以求解相机外参，但是这些解法很容易受到点的二维、三维坐标噪声的影响，会产生很大的误差，因此，我们在求解过程中需要考虑更多的点。当$n \geq 6$时，相机外参仍然可以通过非迭代方法[62–64]求解，如常用的DLT（Direct Linear Transform）[65]方法。非迭代方法的优点是计算量少、效率高，而且不需要设置初始外参，然而对于坐标噪声同样敏感，求解精度不如迭代方法[66,67]。在增强现实的实时三维跟踪过程中，一般可以得到几十上百个匹配点，其中包含一定数目的误匹配，为了快速稳定求解，现在通常的做法是：先用RANSAC等**鲁棒估计**[68]方法随机选取固定数目的匹配点子集，用非迭代方法计算点子集对应的外参，并剔除误匹配，最后在所有得到的正匹配上利用迭代方法优化求解结果[64]。

### 1.4.1 基于基准标志的三维跟踪

相比自然特征，基准标志更容易跟踪和识别，如果要在一些缺少特征的环境中实现三维跟踪，不得不利用一些人工设计的基准标志[69,70]。在增强现实中，基准标志一般指经过特殊设计，可直接打印在白纸上，并通过计算机视觉方法检测识别的标志，可以是圆形或者方形。Cho等[71]用不同数目、大小、颜色的圆环包围圆点，以区分不同的标志。Naimarkhe和Foxlin[72]提出更通用的方法，直接将条码编码在黑色的圆形内，可以提供更多的标志。圆形标志的问题是每个标志只能提供一个特征点，而它们所处的平面很难确定，所以单个标志不能提供稳定的跟踪结果，而方形标志可以提供四个角点[73]。

Kato等[74]的ARToolKit [12]是现今最常用的标志系统，利用方形标志，他们实现了诸如网络会议，魔法书，画图板等应用。ARToolKit检测标志的步骤包括用固定的阈值进行图像二值化，因此对光照变化的适应能力比较弱[75]，ARTag[70,76]扩展了ARToolKit，用更加鲁棒的边缘提取方法检测标志，而且将标志的编码内嵌在标志图像里，不仅避免预处理的

[12] http://www.hitl.washington.edu/artoolkit/

时间和存储，而且降低了识别的错误率。还有一些特殊的标志[77,78]，如Tenmoku等[79]根据场景的背景信息设计对应的具有相似颜色的标志，并将标志放置在物体的角落上，以降低其对应用的视觉影响。Tateno等[80]在大标志里嵌套小标志，使摄像机无论远近都能检测到适当的标志，增加了摄像机的活动范围。而Saito等[81]更是把标志设计成装饰壁纸，铺满房间的地面墙面，用户携带一个俯视的摄像机，可以在室内自由活动。

### 1.4.2 基于点云模型的三维跟踪

基准标志的可控性即是优点也是缺点，因为有些自然场景是无法干预的，在这种情况下，可以利用自然特征点。从一系列无序的图像集中恢复场景的三维点云结构是近年来十分热门的研究领域[82,83]，实时三维跟踪可以利用这些点云的特征信息[84]。Chia等[85]预先恢复两帧参考图像的摄像机参数和Harris特征点[17]，输入图像与这两帧图像匹配，再根据双视几何和三视几何恢复输入图像的参数。他们的特征点匹配总是先根据前一帧的参数求解当前图像与参考图像之间的近似**单应矩阵**，然后将输入图像上的特征点映射到参考图像上，在相邻区域内搜索匹配。可以看出他们的初始帧必须和参考帧非常接近，而且摄像机运动不能太快。他们还利用一个类似**卡尔曼滤波**的**时序调整**，使求解结果更加稳定。Skrypnyk等[86]的方法使用更多参考图像，但输入图像不是直接和参考图像匹配。在预处理阶段，他们利用SfM[65]恢复目标场景的点云结构，每个三维点不仅有三维坐标，还附加了SIFT描述量[14]；在实时三维跟踪阶段，输入图像上的SIFT特征点可以与点云特征直接匹配，得到二维-三维对应点，然后通过RANSAC和Lenvenberg-Marquardt算法优化摄像机外参，并利用前一帧求解结果和动态的权值减少参数的抖动。他们的系统提供了基于点云模型的实时三维跟踪的基本框架[87]。

Yuan等[88]的系统框架与Skrypnyk等[86]类似，但他们用KLT方法[22,23]在连续帧之间跟踪特征点，因此摄像机也不能快速运动。Mooser等[87]在预处理阶段也用KLT方法恢复点云结构，但是他们利用投影基[13]生成特征点的描述量。上述方法共同有一个缺点，如果输入图像的视角与预处理的参考图像相差较大，特征点匹配会很少，难以得到稳定的三维跟踪结果。Irschara等[89]将SIFT三维点云用于快速位置识别，同时求解摄像机外参，而且为了弥补参考图像采样不充分的缺点，他们以点云模型为基础，在摄像机参数空间创建密集的**合成视图**，并用贪婪法从中选取足够覆盖整个场景的视图集合。

### 1.4.3 基于CAD模型的三维跟踪

如果目标物体的几何不太复杂，其CAD模型比较容易得到，实时三维跟踪算法就可以充分利用模型的采样点和边界信息，实现稳定的跟踪。

**基于采样点** 以CAD模型为参考，预先恢复特征点的三维信息是最直接的想法。Vacchetti等[90]在预处理阶段利用CAD模型建立一系列目标物体的关键帧，然后结合关键帧和在线信息恢复摄像机参数。他们首先捕获目标物体的一些关键帧，并将CAD模型与目标物体图像配准，确定关键帧对应的相机参数；然后在每个关键帧上独立检测Harris特征点，反投到CAD模型上得到这些特征点的三维位置。在实时三维跟踪阶段，输入图像分别与最接近的关键帧和前一帧图像匹配特征点，得到二维-三维对应点后，再将前一帧和当前帧的相机参数放在一个目标函数进行局部集束调整[91,92]。由于他们使用的Harris特征点没有包含描述量，所以必须根据前一帧的相机参数选择关键帧，初始化的时候必须将物体放置在与关键帧相近的位置。Park等[93]也是利用关键帧和CAD模型跟踪多个物体，他们将关键帧分成几个子集，输入图像以**循环**方式与关键帧集合穷举匹配；他们使用了Ferns特征点[46]，因此避免了手动配准的初始化。Hinterstoisser等[94]观察到三个不共线的三维点可以确定一个坐标系统，将带纹理的CAD模型随机投影到图像平面，从中提取稳定出现的Harris特征点集合，称为N3M（Natural 3D Markers），并训练随机树分类器[48]来识别这些N3Ms；在运行时，根据匹配成功的N3M，计算出输入图像的相机参数。Özuysal等[47]也利用随机树分类器，在训练阶段动态更新分类器，只选取最稳定的特征点作为参考，但他们仅仅使用了简单的椭球模型来近似目标物体。

**基于边缘信息** 既然我们有CAD模型，就可以轻易得到模型各个表面之间的边界，并根据边界进行三维跟踪。Drummond等[95]不直接求解相机参数，而是将相机的外参表达成**李群代数**空间的六个矩阵基的线性组合，并优化六个线性权值。他们总是以前一帧的相机参数为基础，优化CAD模型上的可见采样边和当前图像上的边缘之间的距离。类似的方法有机械装配领域的**虚拟视觉伺服**[96]，这种方法的优化目标与Drummond等[95]相似，他们以鲁棒**控制律**为指导，通过迭代优化方法一步步收敛到最优解，跟踪结果更精确。然而，他们都是连续跟踪方法，同样有需要手工初始化，不能支持快速运动，跟踪失败之后无法恢复的缺点。**粒子滤波**[97,98]的引入在一定程度上解决了这些问题，每一个粒子代表一个相机

参数假设，可以利用边缘信息来验证这些粒子的可能性，然后选择其中最大可能的粒子作为求解结果。

**联合采样点和边缘信息** 基于采样点的方法在缺少纹理信息的情况下无法应用，而基于边缘信息的方法又很容易受到纹理的干扰，两者显然是互补的，如果集成在一起，可以获得更稳定的跟踪结果[99,100]。Vacchetti等[101]将之前的工作[90]和虚拟视觉伺服结合起来，将特征点投影误差和边缘信息融合在同一个目标函数内优化。Reitmayr等[102]用带纹理的CAD模型扩展了Drummond等[95]的边跟踪方法，为了使系统更加稳定，他们还在线搜集关键帧，提取FAST特征点[18]，如果跟踪失败，可以根据这些关键帧重新恢复跟踪。更进一步，他们利用**扩展卡尔曼滤波**[103]将陀螺仪和摄像头的输入融合在一起，提升了系统的鲁棒性，使其在户外环境得到应用。

## 1.5 并行三维重建和跟踪

基于模型的三维跟踪在增强现实中应用十分广泛，但是对环境的预处理过程比较繁琐，而且一旦环境发生改变，又得重新预处理；因此，研究者提出并行三维重建和跟踪技术，即在一个从未处理的环境中，实时地跟踪摄像机运动，同时重建其三维结构（一般是点云）。根据三维优化模型的不同，并行三维重建和跟踪可以分成SLAM和Online-SfM。SLAM将摄像机运动作为一个动态系统，以**状态-观测模型**来建模求解；而Online-SfM则是强化了SfM的性能，主要依靠多视几何[65]以及局部集束调整[92]。SLAM对计算性能的要求比较低，而Online-SfM的求解结果比较精确[104]。

### 1.5.1 SLAM

在机器人自动导航领域，SLAM指移动感应器平台在运行中即时恢复环境结构，并估计自身运动。长期以来，激光测距仪、声纳等感应器是SLAM的研究中心，而很少考虑廉价的视频摄像机。但是自从Davison等[105]的开创性工作以来，SLAM渐渐走入应用领域，并成为研究热点。SLAM的核心是一个状态-观测模型，每一个相机参数估计都分成两个步骤：1) 根据摄像机运动模型和前一帧的参数，估计当前帧的状态；2) 根据当前帧的观测值，更新状态，同时更新相应的概率分布。

假设摄像机的运动符合线性-高斯概率，那么状态-观测模型的最优解可以用扩展卡

尔曼滤波求得。Davison等[105]将场景的标志点和相机参数作为状态，并使用了一般的平滑运动模型来预测摄像机状态的变化。他们还利用摄像机运动和**标志点**三维的不确定性动态估计搜索范围，大大提升了KLT特征点跟踪的稳定性。Chekhlov等[106]进一步加强了Davison等的特征点跟踪方法，加入梯度描述量和快速尺度估计。但是扩展卡尔曼滤波的计算复杂度是$O(N^2)$，$N$是标志点的个数，一般只能处理数十个标志点的小场景。粒子滤波的计算复杂度是线性的[107–109]，可以同时处理几百个标志点，而且对摄像机运动没有任何假设。将场景分割成若干个子场景，组织成层次结构[110,111]，各个子场景之间的关系通过刚体变换表示，SLAM可以处理更大的场景。

SLAM主要基于特征点连续跟踪方法，因此**重定位**和**回路检测**是SLAM的两个重要问题。重定位指SLAM跟踪失败后，如何自动重新启动跟踪[112]；回路检测指判断SLAM在跟踪一段时间后，是否重新访问同一场景[113]。重定位针对SLAM的鲁棒性和实用性，回路检测则对于SLAM恢复的三维结构的一致性非常重要，但是两者的目标相似，都是识别之前访问过的场景，与物体识别[114,115]，位置识别[89,116]都有直接的联系。

### 1.5.2 Online-SfM

传统的SfM[82,83,117]是精确的场景三维和摄像机参数求解方法，但是通常需要很长时间的优化，不能用于实时系统。Klein等[118]提出了第一个实用的Online-SfM系统，将跟踪和重建分成两个线程，可在双核机器上运行。跟踪线程的任务是根据历史信息估计当前帧的相机参数，把场景中的三维点投影到当前帧上，然后在局部窗口内搜索三维点的FAST匹配点，并估计相机参数。重建线程的任务是恢复场景的三维点，在利用双视几何初始化场景之后，重建线程不断检测并添加新的关键帧，利用已有的关键帧和三维点初始化新关键帧的三维，并不断调用局部集束调整优化关键帧的参数和场景三维。在后续工作中，Klein等加入边特征[119]，使系统更能容忍运动模糊和光照变化。该系统成功地应用于小型桌面的增强现实，但也不能处理大场景。Mei等[120]充分利用双目相机的信息冗余，结合了FAST角点，SIFT描述量，FAB-MAP[113]等算法，可以在校园规模的环境内实时地跟踪摄像机。

## 1.6 实时视频分割技术

增强现实系统的几何一致除了虚拟物体和现实环境的位置配准，还包括虚拟物体和

现实环境之间的遮挡关系的处理，这一点对于虚实融合的真实感十分重要。虚拟物体的CAD模型当然是已知的，如果现实环境的CAD模型也可以获得，虚实遮挡就是简单的图形遮挡剔除问题[121]，但是现实环境一般很复杂，模型很难获得。Berger[122]跟踪场景中的物体边界轮廓线，首先确定这些轮廓线和虚拟物体之间的遮挡关系，然后对轮廓线聚类并计算Snake[123]包围盒，最后利用包围盒来处理遮挡；在有复杂纹理的环境中，物体的轮廓线很容易被分成相隔的若干部分，导致错误的遮挡关系。Kim等[124]利用双目相机计算场景的密集深度，可以得到精确的虚实遮挡效果，但是每幅图像的计算需要若干秒钟，很难用于实时系统。

要在实时系统中实现虚实遮挡，必须做出一些简化，将场景分割成几个层次，再处理层和虚拟物体之间的遮挡。现有许多交互的视频分割技术，如Video SnapCut[125]，LIVEcut[126]，这些方法都需要相当的人工交互，计算代价也很大。Kakuta等[127]提出在背景和摄像机都保持静止的条件下，通过视频分割获得场景中动态物体的层次，并设定地面平坦与摄像机视线平行，快速计算动态物体的层次深度；由于物体分割需要大量计算，他们的系统也不能实时运行。

事实上，如果背景已知，最高效的分割方法是**背景消减**[128]，依据颜色差异检测前景物体，运动差异分析也可以用来分割物体[129,130]。这些方法都只使用颜色或者运动信息进行分割，结果相对都比较差，不能用来处理遮挡，另有许多比较精确的基于**图割**[131]的分割方法。如果要用图割来求解分割，首先需要定义优化的目标函数，包括**数据项**和**平滑项**，而各种分割方法的差异主要就体现在这两项定义的不同。Criminisi等[132]的目标函数有四项，平滑项即是图像**颜色反差**，数据项包括：时序项，二阶的马尔科夫链，根据前两帧的分割结果估计当前帧；颜色项是**高斯混合模型**；运动项，从图像的空间梯度和时间梯度计算当前图像的运动可能性；由于综合考虑了所有信息，他们获得很好的分割结果。Sun等[133]的数据项也是基于颜色的高斯混合模型，不过为了处理更加复杂的背景，尤其降低背景上的强边对结果的影响，他们在平滑项引入一种背景颜色反差压制方法，提升了分割的稳定性。这两种方法都需要静止的背景，而且需要对高斯混合模型进行初始化。Yin等[134]的目标函数有三项，平滑项与Criminisi等[132]一样，数据项包括：运动和形状项，基于强分类器同时对像素的梯度和形状变化进行识别；颜色项，同样是高斯混合模型。虽

然通过以上方法可以得到很好的分割，但也只能完成摄像机静止状态下的简单遮挡效果，因为我们没有办法精确估计层次的深度。

## 1.7 本文工作与结构

上文总结了现有的实时三维跟踪系统，并详细例举其中涉及的关键技术，可以看出三维跟踪是一个综合性的系统问题，在系统的整体设计到每一个技术环节，都有多种可能的选择，必须根据实际应用来遴选和规划。对增强现实来说，由于虚拟信息的内容和场景密切相关，一般需要预先摆放虚拟物体和设计运行脚本，需要知道场景的结构。本文研究的实时三维跟踪针对增强现实应用，将遵循图1.7橙色方框所注的技术路线，场景的三维模型是已知的：包括基于基准标志的实时三维跟踪，和基于点云的实时三维跟踪，系统的总体框架如图1.8。

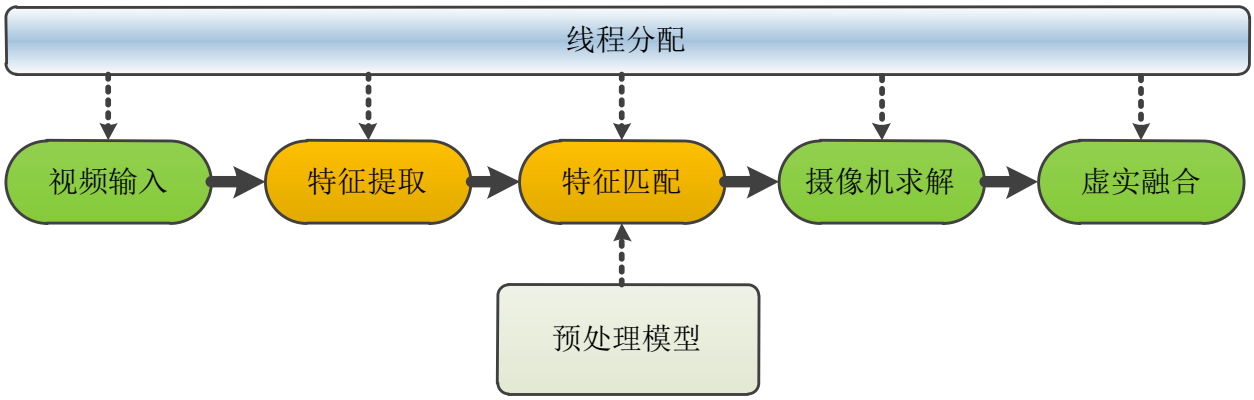



图 1.8 实时三维跟踪系统的框架。

本文的系统在实时三维跟踪的关键技术模块化，模块接口标准化的基础上，将各种现实环境下的增强现实技术统一在一个多线程并行框架里，用户可以便捷地在此基础上开发新的增强现实应用，模块通过线程安全的缓冲区相连，而且各模块可以运行在独立的线程上，充分利用了多核机器的计算能力，使系统在适应各种复杂环境的情况下保证高效可靠的性能。系统首先在预处理阶段通过一定手段获得场景的三维表达。在实时阶段，视频输入和虚实融合模块对于增强现实来说，都是一样的；然而不同的增强现实需要不同的特征，随之而来的是不同的匹配方法；匹配的结果一般是二维-三维对应信息，根据投影误

差的优化可以求解摄像机参数。本文将基于图1.8展示的框架，从以下三个方面改进现有的三维跟踪技术，并实现相应的增强现实应用。

1. 在一些桌面增强现实应用中，我们不能从自然场景中提取足够的特征，必须辅以基准标志。本文提出的基准标志是包围在黑色方框中的汉字图像，汉字内容和虚拟内容直接相关，可以完成类似看图识字的功能。为了在复杂的光影下也能稳定地检测出标志，系统利用边缘提取优化了标志检测方法；同时将汉字的结构表达为汉字轮廓到边框的距离场，增加了汉字标志的可识别度。经典的基准标志一般由黑白两色构成，视觉上不是很美观，因此本文还提出将相对美观的自然图像作为基准标志。

2. 在大规模自然场景中，场景的高复杂度和大规模导致极其丰富的特征，场景的特征点云数量很大，现有的三维跟踪方法在特征匹配的效率和性能上都无法满足实时的要求。本文利用SfM技术从预处理视频序列中恢复场景的稀疏三维点云，提出了一种基于关键帧的场景表达方法，并利用快速图像识别技术选择与输入图像相近的候选关键帧，进行特征点匹配。本文通过贪婪优化方法，从输入预处理视频序列中自动选择一些关键帧，这些关键帧将包含比较稳定的，特征明显的特征点。三维跟踪通过图像识别算法，为输入图像选择候选关键帧，然后只跟这些关键帧进行特征匹配。为了获得更稳定的跟踪结果，本文还利用极限约束，连续帧跟踪等方法匹配更多特征点。

3. 如果用户要求在增强现实中体验更真实的虚实融合，系统必须解决虚实遮挡问题，然而在现有的条件下，实时性和精确性都很难完成。场景层次分割是处理增强现实过程中的物体相互之间遮挡的重要方法，由于场景和摄像机运动的双重复杂性，这也是非常难的问题。本文基于简单的假设，尝试解决在摄像机只有旋转运动情况下前背景之间的遮挡，将其转变成前背景分割问题。系统首先建立背景的全景图，然后将实时输入图像与背景全景图像配准，估计背景信息。针对复杂背景和背景估计的误差，本文结合过分割方法对背景全景图建立局部颜色模型，并充分利用背景的颜色反差信息，得到稳定精确的分割结果。在此基础上，系统还实现了一系列特殊的增强现实效果。

本文剩余章节的结构如下：第2章介绍基于人工标志的实时三维跟踪。第3章介绍利用

场景点云结构和关键帧技术，解决从室内场景到室外场景的实时三维跟踪问题。第4章讨论实时三维跟踪在纯旋转相机和自由运动相机下的增强现实应用，包括场景的静止背景和运动前景的双层分割方法。第5章总结全文，并规划未来工作的发展方向。

# 2 基于人工标志的实时三维跟踪

本文的人工标志是指由用户精心设计，用于辅助实时三维跟踪的标志物，包括基准标志和自然标志。

基准标志在缺少特征信息的桌面增强现实应用广泛。一些增强现实希望简化特征检测，减少场景中需要布置的特征纹理，此时也大多使用基准标志。增强现实最少只需要一个基准标志就可以实现，预处理的工作量很少，实时检测的计算量小，稳定性也比较高。Kato等[74]利用ARToolKit在不同基准标志上绘制不同的用户网络会议视频，这是典型的增强现实应用。在检测标志的过程中，他们的系统要用统一的阈值将输入图像二值化，在复杂的光照变化下，往往会造成错误的标志区域。Fiala[70]提出设计可靠标志的若干标准：如高检测率、抗光照、抗透视，抗遮挡等，并根据这些标准提出Artag系统，在标志中直接内嵌了识别信息，不需要进行标志库查询。他们的优势在于可以同时提供大量的标志，但是对于单个标志的识别反而有时会出现差错，而且Artag的编码图像专为识别算法设计，不能为用户直接认知。本文将ARToolKit和Artag的优点结合在一起，实现了一个基于**汉字标志**的增强现实应用。

本文对ARToolKit的改进还体现在标志识别上。ARToolKit是将从输入图像中检测到的标志二值化后，计算与标志库中的模板标志之间的颜色差，由于二值计算缺少平滑过渡，检测过程中微小的不对齐导致误差比较大，常常会出现误匹配。本文将标志内的汉字的结构表示成平滑的距离场，提升了识别的准确性。

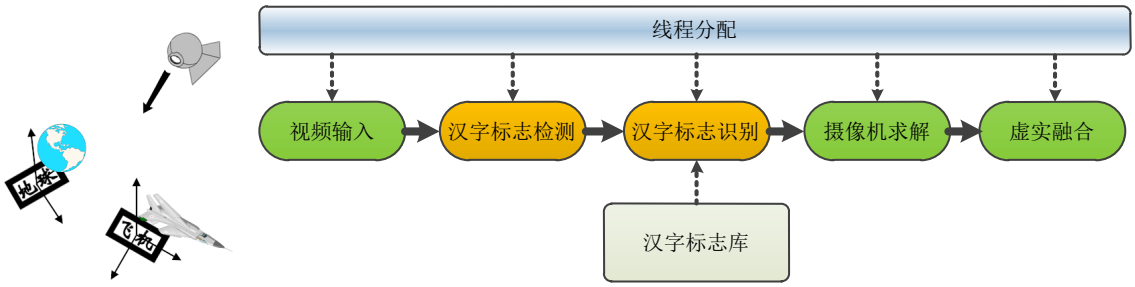
另一方面，基准标志通常都是黑白二值图，从视觉美感的角度来说，比较突兀。事实上，本文可以将普通的图像来作为基准标志，图像的复杂性会带来额外的计算复杂度，但是可以在特殊的应用中使用，这些标志被称为**自然标志**。

## 2.1 汉字标志

增强现实中的虚拟信息可以是图像，文字，三维模型；本文希望利用增强现实技术完成一个稳定的汉字学习系统，将计算机生成的多媒体信息与汉字结合起来，使学习手段更

加丰富有趣（图2.1(a)）。图2.1(b)展示了系统的模块设计，其中最重要的是汉字标志检测和识别两个模块，结合边缘检测和距离场算法，得到了稳定的结果。

系统中每一个标志的大小已知，拥有独立的摄像机参数。相机模型采用的是第1.1节介绍的针孔相机模型（本文工作中所有的相机内参都利用Calib ①预先标定好）。由于标志是一个平面，利用标志方框的四个角点就可以计算摄像机外参[60]。根据标志是一个平面，从单应矩阵出发可以得到一个满足要求的外参初值，然后通过非线性优化过程，优化角点的投影误差来进一步调整。最后，为了消除摄像机外参噪声导致的抖动，本文利用加权的历史运动信息[86]，避免当前的解与之前的外参产生不合理的偏离，以保证虚实融合的平滑性。




(a) (b)

图 2.1 基于汉字标志的增强现实。(a)系统原型，在汉字标志上绘制相关动态内容；(b)系统流程。

综上所述，本节主要介绍三个方面的工作：

1. 利用增强现实，实现了结合多媒体的汉字学习系统，丰富了学习的趣味性；
2. 提出了基于边缘检测的标志检测方法，和基于距离场的标志识别方法；
3. 结合单应矩阵，投影误差，和平滑约束，求解的摄像机参数更加精确稳定。

### 2.1.1 建立汉字标志库

在预处理阶段，系统将建立汉字标志库，汉字标志与ARToolKit的标志类似，汉字置于黑色的包围框中，而且其边框大小已知，即其三维结构已知。假设标志所在的平面是$Z$平面，如果其角点的图像坐标 $m = [u, v, 1]^T$，则$M = [X, Y, Z, 1]^T = [u, v, 0, 1]^T$，如

①http://www.vision.caltech.edu/bouguetj/calib_doc/，如果相机成像有径向畸变，也可以预先标定。

图2.2。汉字标志表示为$C^i = \{L^i, M_0^i, M_1^i, M_2^i, M_3^i, D^i\}$，其中$L^i$指汉字标志对应的名称，$M_j^i$是四个角点的三维坐标，$D^i$则是从标志的原始图像计算得到的距离场描述图像，如图2.2，可以看出距离场描述图像表现了标志的结构。距离场的计算分成两个步骤：1）将原始图像变换成固定大小的正方形图像，并进行二值化；2）计算图像中的像素到汉字边缘和内边框的距离。

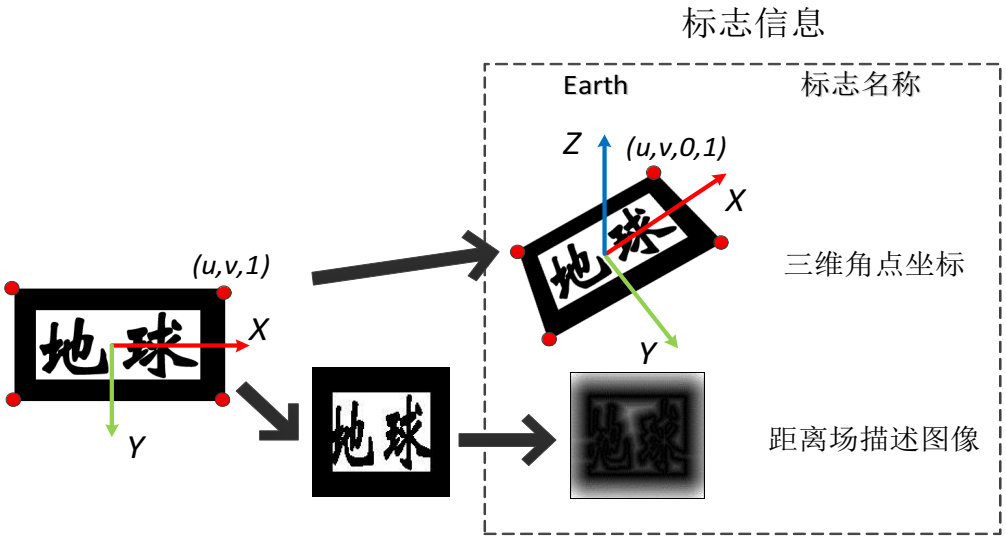



图 2.2 汉字标志的信息。从原始标志图像得到的三维结构和标志的距离场描述图像。

### 2.1.2 检测和识别汉字标志

ARToolKit将图像二值化来提取标志的**轮廓**，这种方法的问题是很难设定一个统一的全局亮度阈值提取所有的轮廓，本文考虑利用局部信息来检测标志轮廓。局部的颜色反差是比较稳定的信息，因此本文通过边缘检测来提取轮廓线。Canny[7]算法是现在最常用的边缘检测方法，系统首先利用Canny算法确定边缘像素（图2.3(b)）。标志的轮廓线总是由四条直线组成，因此要先把边缘像素连成轮廓，然后为每条轮廓线拟合组成方框的四条直线，如果拟合误差在一定阈值之内，那么轮廓所包围的图像区域就作为一个备选的标志。Hough变换是拟合直线的通用方法，但是其结果比较分散（图2.3(c)），确定直线之间的关系会很大地增加算法的复杂度；因此本文使用类似多边形简化的更直接，更高效的方法。对于每一条轮廓，具体步骤如下，

1. 确定轮廓上距离最远的两个顶点$v_0, v_2$，然后分别确定到直线$v_0v_2$最远的$v_1, v_3$；

2. 计算轮廓上所有点到直线 $v_0v_1, v_1v_2, v_2v_3, v_3v_0$的距离 $\{d_i, i = 0, 1, 2, 3\}$；

3. 将点分成四个集合$\{c_i, i = 0, 1, 2, 3\}$，轮廓上任一点$v \in c_t, t = \min_i\{d_i, i = 0, 1, 2, 3\}$（图2.3(b)）；

4. 分别对$\{c_i, i = 0, 1, 2, 3\}$中的点拟合直线，如果得到的四条直线的残差都小于阈值（实验中设为2个像素），那么就认为找到一个标志轮廓线。直线的斜距式表达方程是：$y = kx + b$；直线拟合的目标函数是点到直线的距离:

$$\epsilon = \sum_{p \in c_i} \frac{\|kp_x - p_y + b\|}{\sqrt{k^2 + 1}} / \|c_i\| \tag{2.1}$$

其中$p$是$c_i$中的点，$p_x$和$p_y$是$p$的图像坐标；$\epsilon$的优化可以通过最小二乘法完成。

图2.3(d)是检测到的两个四边形轮廓，四条直线的交点就是标志的四个角点。检测到备选标志区域之后，系统通过模板匹配确定其包含的内容。在此之前，利用标志的四个角点计算面与面之间的单应矩阵$H$，将标志图像归一化到大小一致的图像，再将其二值化，然后同样需要计算距离场描述图像。为了确定标志的旋转方向，系统需要将归一化后的图像与四个方向的标志距离场描述图像匹配，如图2.4。

在计算机视觉领域，单应矩阵$H$指平面之间的可逆映射，既可以是三维平面，也可以是图像平面。$H$是秩为3的$3 \times 3$矩阵，

$$p' = Hp = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} p, \tag{2.2}$$

其中$p$和$p'$是两个平面上的对应点，每一对点可以列两个分别关于$x$坐标或$y$坐标的方程。虽然$H$有九个参数，但可以令$h_{33} = 1$，因此只需要四个对应点就可以线性地确定$H$，这正好符合标志的四个角点的条件。

距离场描述图像之间的距离是简单的$L_2$距离，由于有些备选标志区域并不是真实的标志，需要将其剔除。本文将通过一个2NN判断规则[14]决定一个备选标志是否存在标志库中，$d_0^i/d_1^j < \sigma$，其中$d_0^i$，$d_1^j$分别是备选标志到标志$C^i$和$C^j$的距离，也是到标志库中的标志最小和次小距离；如果两者比值小于阈值$\sigma$（实验中取0.7），那么$d_0^i$对应的标志$C_i$就是备选标志的匹配。


(a) (b)

(c) (d)

图 2.3 通过边缘检测提取标志轮廓。(a) 输入图像；(b) Canny方法抽边结果；(c) Hough变换提取直线； (d) 检测到的标志方框轮廓。

### 2.1.3 求解摄像机参数

系统中每一个标志有独立的坐标系，需要计算摄像机在每一个坐标系中的参数。由于标志只有四个角点的三维信息，而且处在同一个平面上，计算很容易出现退化情况，常常不能得到稳定的解。 ARToolKit通过标志包围框的四条边之间的关系求解摄像机参数，我们发现其对图像噪声导致的直线偏移并不足够鲁棒，因此选择从单应矩阵推导出摄像机参数的初值，再优化投影误差得到最终的参数。

#### 1. 摄像机参数的初始化

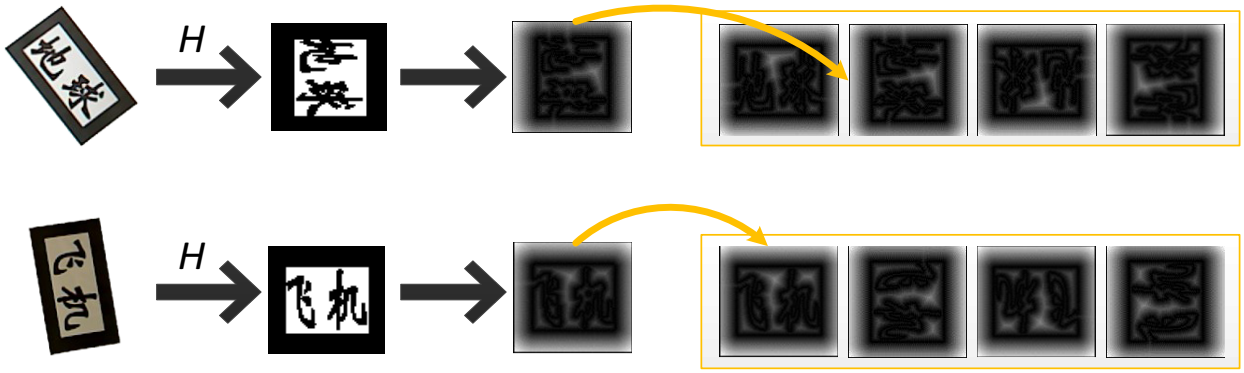



图 2.4 汉字标志的识别。输入图像中检测到的备选标志区域被归一化后，与标志库中的标志模板的四个旋转方向匹配。

设投影矩阵

$$P = K[R|t] = K[\ r_1 \quad r_2 \quad r_3 \quad t\ ],$$

根据公式1.3，$Z$平面任意一点$[X, Y, 0, 1]^T$在图像平面上的坐标

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[\ r_1 \quad r_2 \quad r_3 \quad t\ ] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K[\ r_1 \quad r_2 \quad t\ ] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \tag{2.3}$$

从公式2.3可以看出，$K[\ r_1 \quad r_2 \quad t\ ]$就是从$Z$平面到图像平面的单应矩阵，其中$K$是摄像机的内参，$r_1$和$r_2$分别是摄像机旋转矩阵$R$的第一列和第二列，$t$是摄像机的平移。由于本文假定$K$已知，只要求得单应矩阵$\lambda H = [\ h_1 \quad h_2 \quad h_3\ ] = K[\ r_1 \quad r_2 \quad t\ ]$，$\lambda$是非零的尺度，就可以通过下式计算摄像机的参数，

$$r_1 = \lambda K^{-1} h_1, \quad r_2 = \lambda K^{-1} h_2, \quad r_3 = r_1 \times r_2, \quad t = \lambda K^{-1} h_3,$$

其中$\lambda = 1/\|K^{-1}h_1\| = 1/\|K^{-1}h_2\|$。于是，系统只要求出标志的四个角点从$Z$平面到当前图像平面的单应矩阵，就可以估计出$R, t$的初值。

**2. 优化投影误差与参数平滑**

通过单应矩阵得到的摄像机外参只是一个估计值，往往不够精确，且很容易受噪声的影响。系统将通过进一步优化，抑制噪声带来的影响，得到准确的外参，从而使得合成的

虚拟物体能够保持平稳而不发生漂移或抖动。首先是优化标志角点在图像平面上的投影误差，

$$\min_{R^n,t^n}\sum_i \|K[R^n|t^n]M_i - m_i^n\|^2, \tag{2.4}$$

其中$M_i, m_i^n$对应四个角点的三维坐标和二维坐标，上标$n$表示第$n$帧输入图像，角点的三维坐标保持不变，而二维坐标在不同的图像上变化。优化目标函数2.4得到的摄像机参数比较容易受图像噪声的影响（因为图像噪声使得每一帧上提取的角点位置存在一定偏差），即使当摄像机固定不动时，求解得到的摄像机参数也会有明显的抖动现象，所以系统需要进一步优化，通过平滑约束来消除图像噪声带来的影响。系统将分别利用第$n-1$帧获得的旋转和平移参数来平滑第$n$帧相应的参数，特别是当摄像机的运动很缓慢时，尽量使第$n$帧的解靠近第$n-1$帧。因为摄像机的旋转运动和平移运动变化不一致，旋转的变化通常会更剧烈，所以旋转和平移的平滑项需要分别加权，

$$\min_{R^n,t^n}(\sum_i \|K[R^n|t^n]M_i - m_i^n\|^2 + \alpha\|R^n - R^{n-1}\|^2 + \beta\|t^n - t^{n-1}\|^2), \tag{2.5}$$

其中 $\alpha$ 和 $\beta$分别用来控制旋转和平移运动的平滑度。首先根据公式2.4计算$R^n$ 和 $t^n$，这等价于公式2.5中的$\alpha = 0, \beta = 0$；然后迭代优化公式2.5平滑摄像机参数，每一次迭代，$\alpha$和$\beta$如下更新，

$$\begin{aligned}
\alpha &= \frac{e(R_k^n, t_k^n)}{\min\{e(R^{n-1}, t^{n-1}), e(R^{n-1}, t_k^n)\}} \cdot \frac{e(R_k^n, t_k^n)}{\|R_k^n - R_{k-1}^n\|^2}, \\
\beta &= \frac{e(R_k^n, t_k^n)}{\min\{e(R^{n-1}, t^{n-1}), e(R_k^n, t^{n-1})\}} \cdot \frac{e(R_k^n, t_k^n)}{\|t_k^n - t_{k-1}^n\|^2}, \\
e(R,t) &= \sum_i \|K[R|t]M_i - m_i^n\|^2.
\end{aligned}$$

根据摄像机的运动相对于前一帧主要是旋转运动变化为主还是平移运动变化为主，旋转和平移的平滑系数会做出自适应的调整：如果相对于前一帧，主要是旋转参数发生了变化，那么旋转的平滑系数就相对调低；反之，平移的平滑系数相对调低。$k$表示第$k$次迭代结果，在本文的实验中，最多$3 \sim 5$次迭代已经足够。以上所有的非线性优化都使用Levenberg-Marquardt算法。

## 2.2 自然标志

基准标志一般有易于识别的特征和良好的结构，使检测和跟踪比较稳定；但是以黑

白色为主的基准标志往往不具有美感。相比基准标志，基于自然特征[95,99,101,118]（点，边，纹理等）的增强现实更符合用户的认知习惯，也更适用于普通场景，一直是研究热点。局部特征点又因其丰富性和鲁棒性[14,20,32,46]（尺度不变，旋转不变，仿射不变等）成为最常用的自然特征。近几年，基于局部特征点的跟踪技术有了许多进展[47,86]，许多利用局部特征点的增强现实系统展示了很强的鲁棒性，能处理复杂的环境，而且具有更友好的界面。

Özuysal等[46]提出当前最稳定的单个目标跟踪方法之一，他们依赖离线密集采样得到的物体特征信息，可以很快速地匹配输入视频中的特征点。Park等[93]扩展了Özuysal等的工作，首次提出同时跟踪多个目标的系统。他们利用特征识别和关键帧技术，可以同时处理多个简单的三维物体。他们将每一个物体表示成特定的关键帧集合，所有关键帧被分成几个组，每一帧输入图像与一个关键帧组匹配，完成物体的检测和跟踪。这个系统的缺点是特征存储空间太大，限制了跟踪物体的个数，而且基于关键帧集合的检测方法存在较明显的延时。现实应用常常面对很多平面结构，或平面的组合，本文简化目标的三维结构，提出基于平面图像的增强现实。

Castle等[135]把SLAM系统[105]和基于SIFT特征点[14]的平面物体识别结合在一起，在SLAM更新摄像机参数和环境信息的同时，检测预先指定的平面物体，并把物体的三维信息加入SLAM所维护的环境信息中。本文的自然标志使用了与其类似的模型，但是他们的自然标志检测和跟踪是异步进行的，存在较明显的延时，而且物体之间的相对位置一经确定就不能变动，否则将破坏SLAM维护的环境信息。

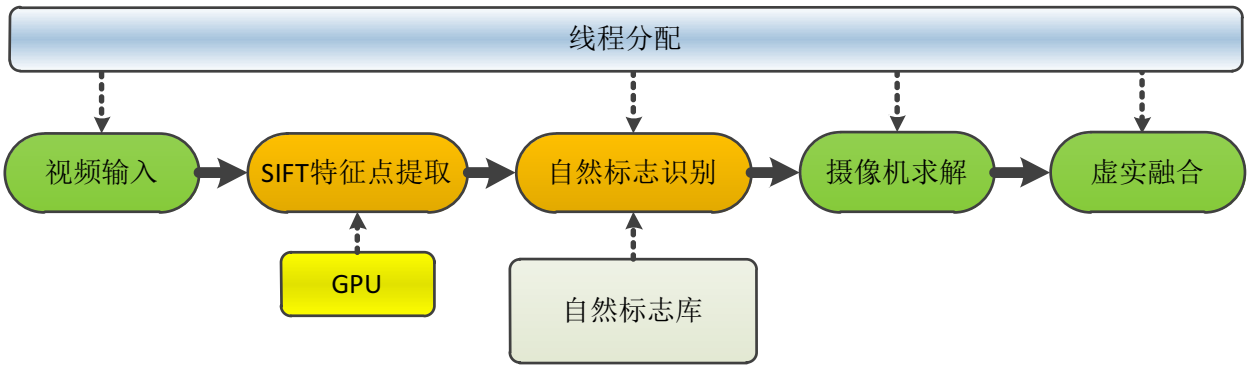



图 2.5 基于自然标志的增强现实。

本文将每一幅图像上的SIFT特征点[14,32]组合起来作为检测和求解的基本单元，称为自然标志。与汉字标志类似，自然标志拥有独立的模型和特征信息，并对应独立的摄像机参

数。为了进一步提升检测的效率，系统在处理自然标志时，将自动剔除一些相似的特征点，减少特征点的数量和特征匹配的时间，同时也增加特征点之间的区分度。图2.5展示了系统的流程，实时捕获的视频图像上的SIFT特征点需要和每一个自然标志匹配，以确定特征点所对应的标志。为了增强特征跟踪对于图像噪声的抗干扰能力（比如在噪声的影响下，很容易造成一个特征点在前一帧出现，在当前帧却没有检测到的情况），本文结合KLT方法[50]，对匹配失败的SIFT特征点继续进行追踪，这有效提高了特征匹配的稳定性。为了进一步提升运行效率，系统利用GPU的并行计算能力来提升SIFT特征提取的性能。

### 2.2.1 建立自然标志库

图2.6 展示了三幅预处理的平面图像和提取的SIFT特征点，每一平面图像的所有特征点组合成一个自然标志。于是，每一个自然标志表示为一系列特征点的集合$S^i = \{(m_0^i, M_0^i, D_0^i, V_0^i), (m_1^i, M_1^i, D_1^i, V_1^i), \ldots\}$，其中$m_j^i$是特征点的图像坐标，$M_j^i$是$m_j^i$的三维坐标，$D_j^i$是特征点的DoG值，DoG值越大，代表特征点越稳定，$V_j^i$是特征点的SIFT描述量。由于所有的特征点在同一个平面上，系统把图像平面当作$Z$平面，若$m_j^i = [u, v, 1]^t$，则$M_j^i = [X, Y, Z, 1]^t = [u, v, 0, 1]^t$。

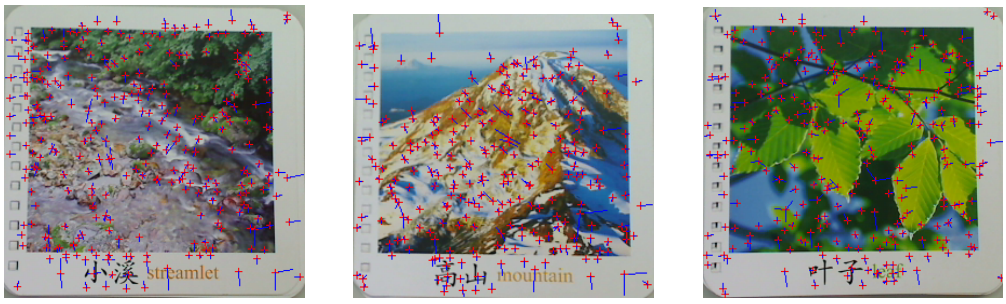



图 2.6 自然标志库的三个自然标志。红色交叉是提取的SIFT特征点的位置，蓝色线段表示其方向和尺度。

每一个自然标志通常包含几百个SIFT特征点，如果某些特征点的描述量在特征空间距离太近，会导致这些特征点都无法匹配成功，系统将自动剔除这些点，得到一个新的特征点集合$S^{i'}$。需要注意的是，在剔除特征点的过程中，算法尽量保留稳定的特征点，具体见算法2.1。其中特征相似度的判断利用了2NN判断规则[14]，即如果一个特征点与$S^{i'}$中的特

征点的最小距离和次小距离比值越小，则特征点之间越相似。$\sigma$控制算法对相似度的容忍性，$\sigma$越大，被剔除的特征越多。

**算法 2.1**: 相似特征点过滤算法

**目标：**给定一个自然标志$S^i = \{(m_0^i, M_0^i, D_0^i, V_0^i), (m_1^i, M_1^i, D_1^i, V_1^i), \ldots\}$，SIFT特征点根据其DoG值降序排列，确定新的稳定特征点集合$S^{i'}$。

1. 设$S^{i'} = \{(m_0^i, M_0^i, D_0^i, V_0^i), (m_1^i, M_1^i, D_1^i, V_1^i)\}$.
2. 对$S$中的每一个特征点$(m_j^i, M_j^i, D_j^i, V_j^i), j > 1$,
   - 计算$V_j^i$与$S^{i'}$中所有特征点描述量的距离，其中最小距离记为$d_0$，第二小距离记为$d_1$，
   - 如果 $d_0/d_1 > \sigma$，$S^{i'} = S^{i'} \cap \{(m_j^i, M_j^i, D_j^i, V_j^i)\}$。

### 2.2.2 检测和跟踪自然标志

系统实时提取的SIFT特征点要与所有自然标志进行匹配，以确定其所属的自然标志。基于SIFT的特征匹配检测已经是比较鲁棒的匹配方法，但是抗噪能力不强，特征点会在视频中闪烁丢失，这样会导致求解的摄像机参数抖动。本文将结合KLT跟踪方法[22]，在连续帧之间局部搜索补充丢失的SIFT特征点。

对于系统捕获的第$n$帧图像，先提取SIFT特征点$\{(m_0^n, V_0^n), (m_1^n, V_1^n), \ldots\}$。对于特征点$m_i^n$，设$m_i, m_j$是某个自然标志的特征点中与$m_i^n$的描述量距离最小和次小的两个点（利用ANN库②得到），如果满足2NN判断规则[14]：

$$\frac{\|V_i^n - V_i\|}{\|V_i^n - V_j\|} < \lambda,$$

就认为$m_i^n$与$m_i$是同一个特征点，$m_i^n$的三维坐标是$M_i$，实验中取$\lambda = 0.70$。如果在$n-1$帧里的匹配上的特征点$m_i^{n-1}$没有在第$n$帧里匹配上，那么很可能是因为噪声或者运动模糊等原因没有被提取出来，或者因为区分度不够。那么以$m_i^{n-1}$为初始位置，利用KLT方法，可

②http://www.cs.umd.edu/~mount/ANN/

以跟踪到其在第$n$帧的位置$m_i^n$。由此可以得到一组特征点在当前图像上的坐标和其三维坐标$\mathcal{F}=\{(m_0^n,M_0),(m_1^n,M_1),\ldots\}$，如图2.7中对应的绿点。

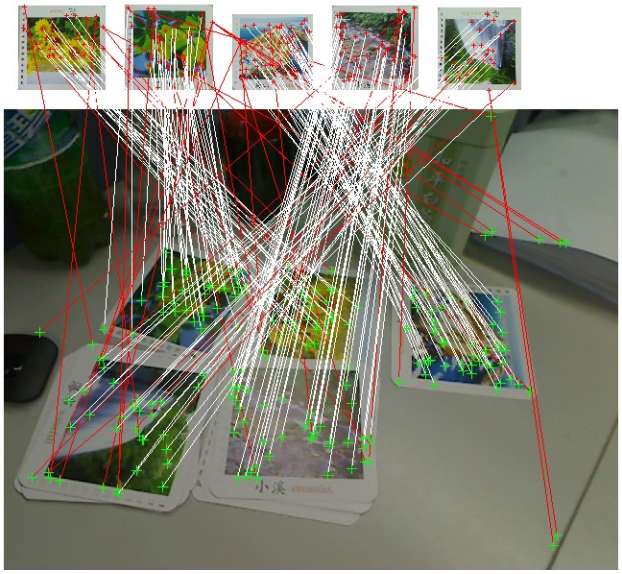


图 2.7 自然标志的检测。上方是检测到的五个自然标志，下方是输入图像。线条表示匹配到的特征点，白线代表正匹配，红线是通过单应矩阵剔除的误匹配。

由于每一个特征点都要到所有的自然标志中匹配，很可能发生误匹配，甚至有些特征点会对应到多个自然标志，如图2.7中的红线。这些误匹配会导致求解的错误，必须先行剔除。在本文的系统中，自然标志是一个平面，其三维平面结构和二维特征点之间存在单应映射，$HM_i=m_i^n$。$m_i^n=[u,v,1]^T$，$M_i=[X,Y,1]^T$表达为奇次坐标，分别是当前图像上的二维坐标和三维平面上的坐标。$H$是一个3*3的矩阵，最少可以由四个匹配点确定。由于提取的特征点位置误差和误匹配的存在，并不严格满足单应映射，但是结合RANSAC方法，系统仍然可以过滤误匹配，见算法2.2。

在本文的实验中，最大循环次数$N=30$，$\mu=5$。如果自然标志有20个以上的点被匹配上，就认为被检测到。最优的单应矩阵$H^n$将在$\mathcal{F}'$中全部匹配点重新计算。最后每个自然标志的摄像机参数计算方法与汉字标志（第2.1节）类似，不同之处在于参数的初始值可以从$H^n$得到，而且优化投影误差的特征点集合是$\mathcal{F}'$。

## 2.3 实验结果

本节的系统运行在桌面系统上，CPU是Intel(R) Core(TM)2 Quad CPU Q9550 @

**算法 2.2**: 利用RANSAC方法求取单应矩阵过滤误匹配

**目标：**给定第$n$帧图像特征点匹配集合$\mathcal{F}=\{(m_0^n,M_0),(m_1^n,M_1),\ldots\}$，确定最优正匹配集合$\mathcal{F}'$。

1. 设$\mathcal{F}'=\emptyset$，$k=0$。
2. 从$\mathcal{F}$中随机选择四个匹配点，计算单应矩阵$H_k$。
3. $\mathcal{F}_g=\{(m_i^n,M_i),\|H_kM_i-m_i^n\|<\mu\}$。
4. 如果$\|\mathcal{F}_g\|>\|\mathcal{F}'\|$，$\mathcal{F}'=\mathcal{F}_g$。
5. 如果 $k<N$，$k=k+1$，转到第2步。

2.83GHz，显卡是GeForce GTX 275。系统的输入设备为罗技高清摄像头C905，输入视频分辨率是$640\times480$，捕获帧率在$20\sim30$fps之间。虚实融合采用OpenGL API，输出设备为普通液晶显示器。

**1. 汉字标志的增强现实结果**

系统各模块具体运行时间见表2.1，系统运行帧率在25fps以上，单帧运行时间在30毫秒左右，只需要一个线程就可以达到实时性能。系统针对八个汉字标志，测试了946帧输入图像，图2.8显示部分输入图像的增强现实结果。

| 模块 | 运行时间（毫秒） |
|---|---|
| 检测汉字标志 | $\approx 10$ |
| 识别汉字标志 | $\approx 10$ |
| 求解摄像机参数 | $\approx 3$ |
| 虚实合成 | $\approx 5$ |

表 2.1 汉字标志的增强现实模块运行时间

**2. 自然标志的增强现实结果**

测试自然标志数据库里包括十个自然标志，根据不同的$\sigma$过滤相似特征点得到

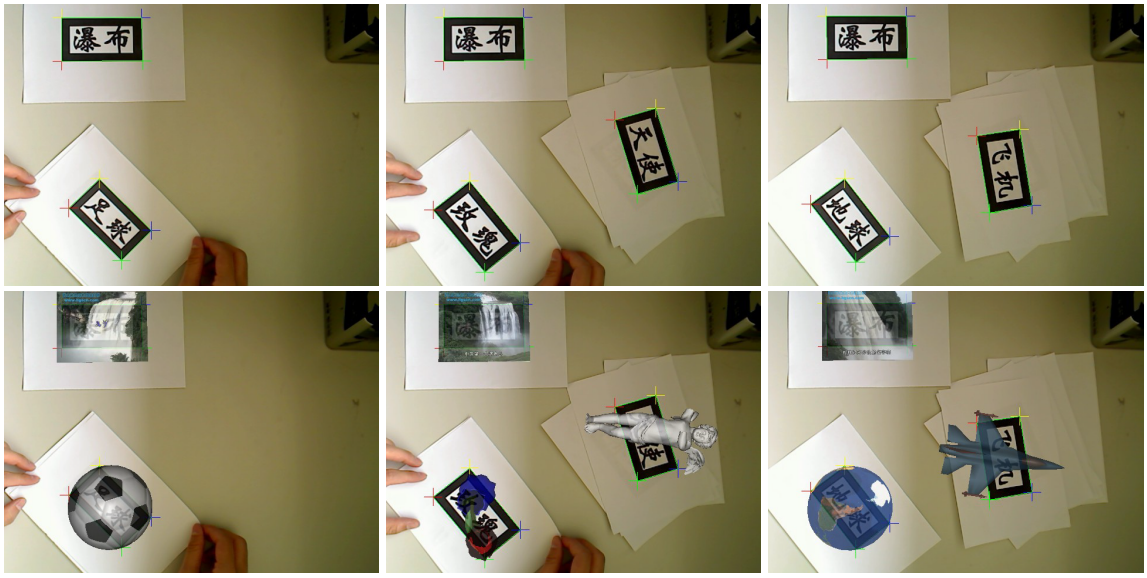



图 2.8 汉字标志的增强现实结果。第一行是输入的第100，500，700帧图像及检测到的汉字标志，第二行是增强现实之后的结果。

的SIFT特征点数见表2.2。可以看出，随着$\sigma$增大，自然标志包含的特征点越来越少，而匹配成功的点会增加；然而太大的$\sigma$会把好的特征点也过滤掉，因此匹配点又会减少。本节实验选择$\sigma = 0.5$，特征点减少约10%。

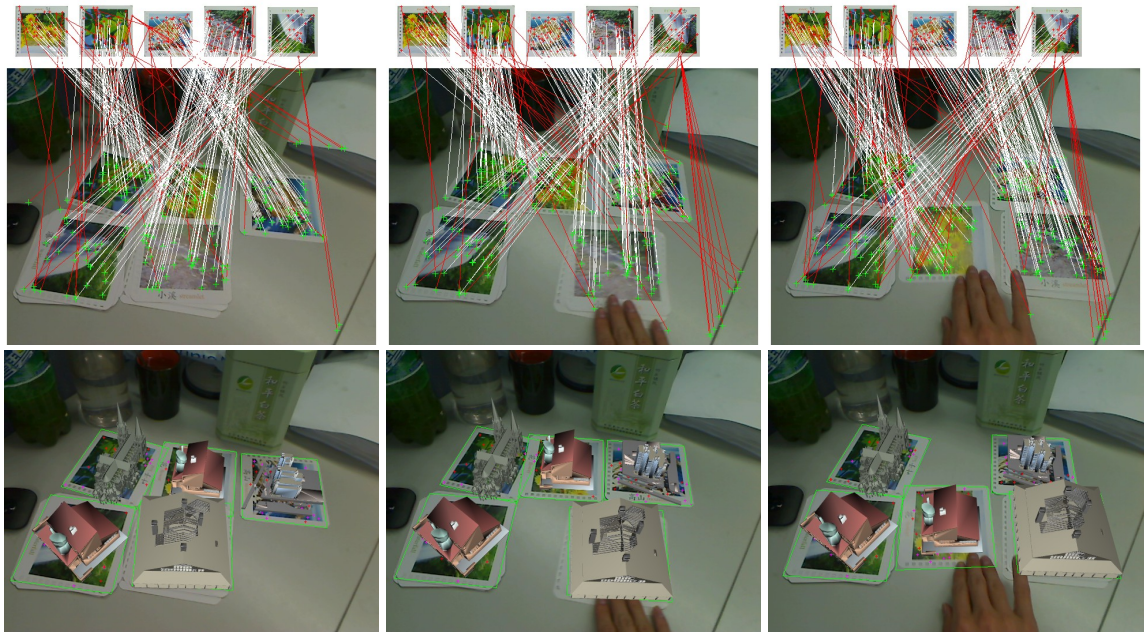


图 2.9 自然标志的增强现实结果。第一行是输入的第26，206，306帧自然标志检测结果(白色线条表示正匹配，红色线条表示误匹配)，第二行是增强现实结果。

| $\sigma$ | 特征点数 | 平均每帧匹配的特征点数 |
|---|---|---|
| 1.0 | 22 | 匹配失败 |
| 0.75 | 3792 | 174 |
| 0.5 | 4093 | 206 |
| 0.25 | 4333 | 184 |
| 0.0 | 4548 | 160 |

表 2.2 过滤后特征点数和匹配成功特征点数

| 模块 | 运行时间（毫秒） |
|---|---|
| 基于SiftGPU [③] 检测标志 | ≈ 55 |
| KLT跟踪特征点 | ≈ 5 |
| 求解摄像机参数 | ≈ 5 |
| 虚实合成 | ≈ 5 |

表 2.3 自然标志的增强现实模块运行时间

系统各模块处理一帧图像的时间见表2.3，运行帧率大约12fps，单帧运行时间在70毫秒左右，需要两个计算线程才能达到实时性能。图2.9显示了一部分增强现实结果，系统在自然标志上绘制了相应的建筑模型，控制其布局。

## 2.4 小结

本章在实时三维跟踪系统框架下实现了基于汉字标志和自然标志的实时三维跟踪，并实现了增强现实应用。

基于汉字标志的增强现实系统将汉字学习与计算机生成的多媒体可视化信息结合起来，丰富了汉字学习的手段和乐趣。本文的边缘检测相比于传统的基于阀值的检测方法，对光照变化具有较好的容忍性，能够适应实际应用中复杂的光影变化。在摄像机参数的求解上，系统利用单应矩阵可以快速求解出好的初值，从而有效避免了直接的非线性优化极易陷入局部最优解的问题。而且，本文的摄像机参数求解加入了平滑约束，并根据旋转和

平移运动的异步性，提出了一套自适应的权重系数调整，不仅有效抑制了图像噪声对求解的影响，而且改善了求解的精度和稳定性。

基于自然标志增强现实系统利用相对美观的图像来作为人工标志，通过SIFT特征点，将增强现实和特征识别结合起来，丰富了静态平面图像的信息，在现实生活（报刊，广告，画展等）中有广泛的应用前景。与基于汉字标志的增强现实系统相比，自然标志直接利用自然图像，应用更加方便，更加通用。

# 3 基于关键帧的实时三维跟踪

第2章探讨了人工标志在增强现实中的应用，但是在许多自然环境中不方便摆放人工标志。虽然自然标志已经使用了SIFT特征点识别用户提供的标志图像，但还是不能避免许多人工干预，本文希望直接利用场景中自有的特征信息实现三维跟踪。现有的基于自然特征的三维跟踪主要运用的是特征点信息。

基于特征点的并行三维重建和跟踪在处理小规模场景上具有很大的优势，因为系统不需要预处理的过程，具有很强的适应性和随意性。但是并行三维重建和跟踪的技术方法很难处理大规模场景，不论是集束调整[118]，还是状态滤波器[105]都难以达到实时性能；而且，每次系统运行的初始阶段，其实也需要一个交互收敛的初始化过程，每次重建的三维结构不能保证一致，如果应用于增强现实，那么虚拟信息的位置每次都要重新调整。

基于点云模型的三维跟踪方法由于具有固定已知的三维结构，可以达到更高的性能。对于特征点的跟踪，主要有两种方法：连续跟踪和匹配跟踪（第1.2节）。连续跟踪是通过一定的方法[23,49]在连续帧之间限定特征搜索范围，以达到快速跟踪的效果，但是误差容易累积、鲁棒性较弱，一旦受到快速运动、遮挡等干扰跟踪失败之后，就需要其它方法来重新启动跟踪。并行三维重建和跟踪主要使用的也是连续跟踪，具有同样的问题。匹配跟踪总是有一个全局的参考，比如关键帧[112]、点云[86]等，输入图像上的特征点随时可以与这些参考信息匹配，恢复失去的方位。可以看出，连续跟踪和匹配跟踪是两种互补的跟踪方法，Lee等[50]已经将两者结合起来应用于桌面增强现实，本文的工作将进一步将两者的结合应用于更大规模的环境。

当然，本文主要还是研究如何提升基于匹配跟踪的三维跟踪系统的性能，在最大程度保证系统的鲁棒性的同时，保证实时性能，系统流程见图3.1。该系统是基于点云模型的实时三维跟踪系统，特征点可以是任意的局部特征点（第1.2节），这里使用的是流行的SIFT特征点[14,32]。系统在预处理阶段通过SfM恢复场景的三维点云，并依此选择关键帧，然后在三维跟踪阶段基于关键帧匹配得到二维-三维对应点；其中最重要的是最优关键帧集合的自动选择，和快速识别与实时输入图像相似的候选关键帧。

关键帧技术在多媒体领域被广泛地用来概括视频内容，减少数据冗余，用于视频压

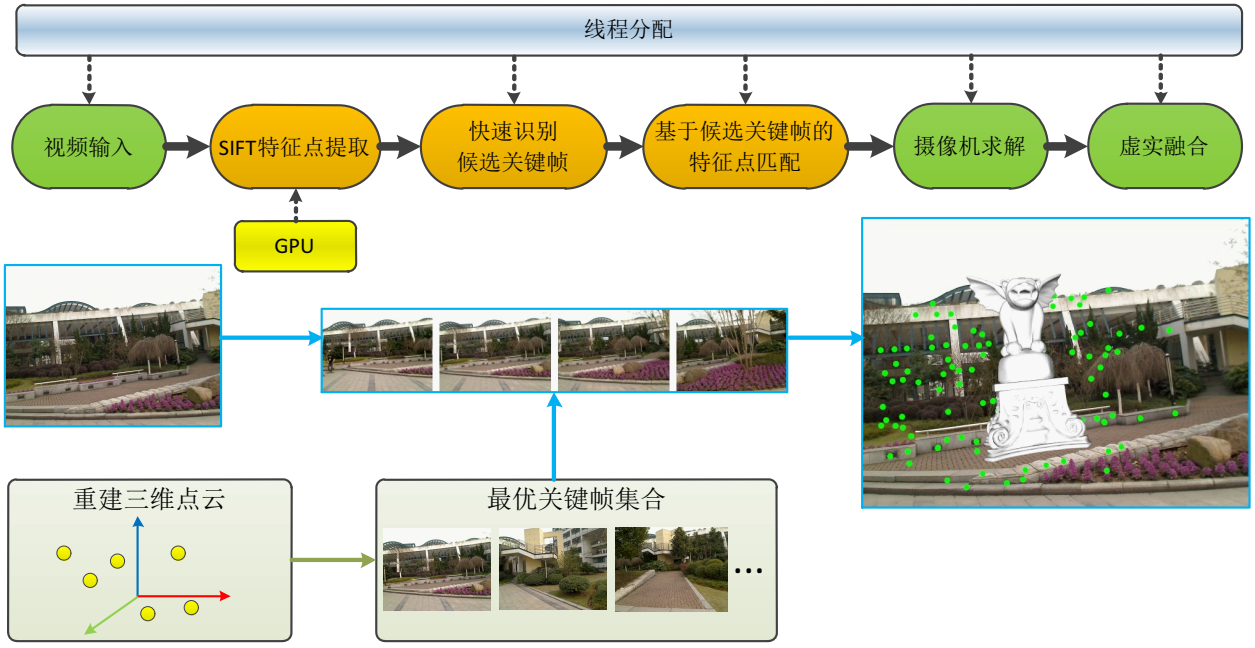



图 3.1 基于关键帧的三维跟踪。系统利用的是场景的三维点云信息，结合场景的关键帧表达和基于关键帧的匹配方法，完成自然场景的三维跟踪。

缩、预览、传输等[136–138]。他们一般通过直接比较视频图像之间的颜色差来估计关键帧与原始视频之间的误差，仅仅针对图像内容进行优化，而不会考虑三维信息，对于三维跟踪要求的关键帧来说不是最优的。在实时三维跟踪领域，关键帧主要为匹配跟踪提供参考信息，完成定位[102]。Klein等[118,119]在线自动选择关键帧，用于集束调整，优化场景的三维结构；为了解决重定位问题，还为每一个关键帧计算了简单的图像描述量。Vacchetti等[101]在离线阶段手工选择关键帧，实时输入图像会与相同可见区域最大的关键帧匹配。Park等[93]则是为多个物体选择关键帧，同时跟踪多个物体。在以上方法中，关键帧或是手工，或是通过简单的时间差、方位差选择，都没有针对三维跟踪的目标进行特殊的优化。

本文的工作首先体现在关键帧的选择方法上，该方法从SfM恢复的场景三维点云出发，针对实时三维跟踪的特点，自动而快速地从预处理视频序列中选择关键帧。本文提出以下三点关键帧需要满足的条件：

1. 关键帧包含的三维信息必须逼近原始序列，即包含尽可能多三维点；

2. 关键帧之间的重复特征要尽可能少，以减少冗余信息；

3. 三维特征点要在关键帧之间均匀分布，保证输入图像匹配到的特征点能覆盖较大的空间，保证摄像机求解的精度。

在实时三维跟踪阶段，确定相似的候选关键帧可转化为信息检索领域的图像识别问题。本文提出一种高效的关键帧识别算法，可以从几百帧关键帧中很快地确定最相似的候选关键帧。因此，输入图像上的特征点只需要与少量的候选关键帧匹配，不用与全局点云匹配，即节省匹配时间，又提高匹配的成功率。在输入图像和候选关键帧匹配时，本文还提出两遍匹配方法，得到尽可能多并分布均匀的特征点。

现有许多方法利用局部特征点进行物体、位置识别[113,114,116,139]。Video Google[114]将文本检索的框架和概念应用于图像检索，提供了大规模图像检索的基本框架：1) 训练**视觉词汇表**；2) 计算图像**外观向量**；3) 比较图像外观向量。为了高效处理大规模图像数据库，Nister等[115]利用**词汇树**来组织和搜索百万级特征点，由于没有对图像数据库进行冗余数据消除，不能直接应用于实时跟踪。Cummins等[113]考虑特征之间的相关性，在概率框架应用**词汇包**提升位置识别的精度，但计算量太大，不能用于实时系统。

最近，Eade等[140]在SLAM系统中基于关键帧统一处理重定位和回路检测。这个方法在实时跟踪时渐进式地建立视觉词汇表，输入图像根据这个动态的词汇表和保存的位置图像进行匹配，以识别访问过的位置。然而对于重定位，输入图像需要和所有位置进行匹配，如果存储位置过多，匹配时间会线性增加，而本文的识别方法的计算时间几乎是恒定的。

Irschara等[89]基于SfM点云的快速位置识别和本文的系统十分相似，同样使用了关键帧技术。为了压缩图像数据库的大小，他们先从输入图像中自动生成密集的合成视图，然后从所有视图中自动选择最优的关键帧集合覆盖所有视角。与他们不同，本文的目标函数是以三维点云为标准，同时考虑完备性和冗余性，而且提出两遍匹配方法进一步增加二维-三维对应点，这对于鲁棒的摄像机方位求解非常重要。

## 3.1 最优关键帧选择

在预处理阶段，系统从参考序列中恢复场景的三维点云结构，为了能跟踪在大规模场景中运动的摄像机，必须增加特征检测的效率，减少特征匹配的歧义，本文的出发点是选

择参考图像的子集表达整个场景，即关键帧。这些关键帧应该包含尽可能多的显著特征点，但要避免特征点重复，而且这些特征点应该均匀分布在环境中。

我们先给出最优关键帧选择问题的模式化定义：给定$n$帧输入参考序列 $\hat{I} = \{I_i | i = 1, 2, ..., n\}$，尝试从中计算出一组最优子集（即关键帧集合） $F = \{I_k | k = i_1, i_2, ...i_K\}$，使其相关的目标能量函数$E(F; \hat{I})$最小。关键帧的数目$K$根据实际需要可以自由调整。$E(F; \hat{I})$包括两项：完备项$E_c(F)$和冗余项 $E_r(F)$；分别表示关键帧包含的三维特征点与原始场景之间的相似度，和关键帧中特征信息的冗余度。

$$E(F; \hat{I}) = E_c(F) + \lambda E_r(F), \tag{3.1}$$

其中$\lambda$是一个权值，平衡完备性和冗余性之间的关系，实际上控制了关键帧的数目$K$。

### 3.1.1 完备项

完备项的作用是约束所选择的关键帧包含尽可能多的稳定SIFT特征点。在离线SfM阶段，系统都希望特征点可以在更多的帧成功匹配，因为匹配帧数越多，恢复的三维越准确，有利于实时三维跟踪的稳定性。所有相互匹配的SIFT特征点都有相似的描述量[14]，为了压缩存储空间，本文将对应同一个三维点的SIFT特征点进行聚类，成为一个**SIFT跟踪点**，用$\mathcal{X}$表示。$\mathcal{X}$包含多个分布在不同帧的SIFT特征点，$\mathcal{X} = \{\mathbf{x}_i | i \in f(\mathcal{X})\}$，$f(\mathcal{X})$表示$\mathcal{X}$出现的图像集合。$|f(\mathcal{X})|$肯定大于1，因为跟踪点只在一帧出现的话，三维就不能恢复。如果 $|f(\mathcal{X}) \geq l|$，$\mathcal{X}$就被称为**优先跟踪点**，所有的优先跟踪点表示为集合$V(\hat{I})$。在本文的实验中，$l$取$5 \sim 21$之间，只有长度大于$l$的优先跟踪点的三维坐标才有意义。

关键帧应该包括特征显著的特征点，因为这样的特征点更有可能被重复检测出来。跟踪点的**显著度**由两个因素决定：跟踪点的长度和SIFT相应的DoG反应值：

$$s(\mathcal{X}) = D(\mathcal{X}) \cdot \min(|f(\mathcal{X})|, T), \tag{3.2}$$

其中，$T$是一个截断阈值，防止某些过长的跟踪点压制了其它跟踪点的贡献（实验中设为30）。$|f(\mathcal{X})|$越高，表示这个特征的匹配成功率越高，越稳定。$D(\mathcal{X})$表示为：

$$D(\mathcal{X}) = \frac{1}{|f(\mathcal{X})|} \sum_{i \in f(\mathcal{X})} D_i(\mathbf{x}_i),$$

其中，$D_i$是图像的DoG反应值。$D(\mathcal{X})$的值等于$\mathcal{X}$在所有$f(\mathcal{X})$中的DoG的平均值。$D(\mathcal{X})$ 越大，SIFT特征越显著。

除了上述两个衡量标准，还要考虑另外一个对三维跟踪很重要的因素：特征点在空间分布的均匀性。为了求得更精确的摄像机参数，匹配到的对应三维点应该在空间中分布均匀，否则参数结果可能有错误的倾向。但是直接在三维空间控制特征点的分布是很困难的事情，本文选择在图像平面上完成这个工作，先按算法3.1计算图像$i$中像素$\mathbf{y}$的特征密度$d(\mathbf{y}_i)$。所有图像的特征密度计算完毕之后，就可以定义跟踪点的特征密度为

$$d(\mathcal{X}) = \frac{1}{|f(\mathcal{X})|} \sum_{i \in f(\mathcal{X})} d(\mathbf{x}_i),$$

其中 $d(\mathbf{x}_i)$表示图像$i$中像素$\mathbf{x}_i$的特征密度。

**算法 3.1**: 图像的特征密度计算

1. 将图像$i$的特征密度图初始化为0。
2. 对 $j = 1, ..., m$, $\%$ $m$ 是图像$i$上的特征数目
   对每个像素 $\mathbf{y}_i \in W(\mathbf{x}_j)$,
   $\%W$ 是$31 \times 31$ 的局部窗口
   $\%\mathbf{x}_j$ 是特征$j$在图像$i$中的坐标
   $d(\mathbf{y}_i)$ += 1.

综上所述，完备项的完整定义为：

$$E_c(F) = 1 - (\sum_{\mathcal{X} \in V(F)} \frac{s(\mathcal{X})}{\eta + d(\mathcal{X})}) / (\sum_{\mathcal{X} \in V(\hat{I})} \frac{s(\mathcal{X})}{\eta + d(\mathcal{X})}), \tag{3.3}$$

其中$\eta$ 控制特征密度的敏感度（试验中设为3），$V(F)$ 表示关键帧集合$F$所包含的优先跟踪点。

### 3.1.2 冗余项

为了减少三维信息的冗余度，简化特征匹配，关键帧之间的重复跟踪点必须尽可能少，最优的情形是跟踪点在且仅在一个关键帧中出现。既然已知跟踪点所出现的图像集合$f(\mathcal{X})$，那么冗余度就可以表示成$f(\mathcal{X})$与关键帧集合之间的交集，冗余度少，则交集小。

$$E_r(F) = \frac{1}{|V(\hat{I})|} \sum_{\mathcal{X} \in V(F)} (|f(\mathcal{X}) \cap F| - 1), \tag{3.4}$$

其中$1/|V(\hat{I})|$是归一化，$|f(\mathcal{X})\cap F|$计算跟踪点$\mathcal{X}$在关键帧集合中出现的次数。如果$|f(\mathcal{X})\cap F|=1$，则表示没有任何冗余。

### 3.1.3 关键帧选择贪婪法

穷举$\hat{I}$所有可能的子集，计算出最小的$E$（公式3.1），当然可以求出最优的关键帧集合，但是这样要处理$2^n$个子集，运算代价太高。在Liu等[136]的视频内容摘要自动提取方法中，他们固定了关键帧的数目，利用**动态规划**搜索最优的关键帧。这个方法在这里并不适用，因为关键帧数目不能预先指定，而且他们只考虑相邻帧之间的重叠区域。于是，本文利用贪婪法来得到一个近似的解，虽然得到的关键帧集合只是一个局部最优解，但对三维跟踪的结果几乎没有影响。

贪婪法的关键帧选择方式类似于数值优化中的最速下降法，见算法3.2。具体流程如下：首先设关键帧集合$F$为空集；在迭代优化的每一个循环，从输入序列中选择使目标函数$E$下降最多的图像加入关键帧集合；直到$E$不能再下降或输入序列为空为止。这个算法的计算复杂度是$O(n^2)$，理论上还是比较消耗时间，不过实验发现，从成百上千帧输入图像中选择关键帧也只需要几秒钟。

**算法 3.2**: 关键帧选择贪婪法

1. $F=\emptyset$.
2. 如果 $\forall I_i\in\{\hat{I}\setminus F\}, E(F\cup\{I_i\})\geq E(F)$, 退出。
3. 否则 $I'=\arg\min_{I_i\in\{\hat{I}\setminus F\}}E(F\cup\{I_i\})$, $F=F\cup\{I'\}$, 跳转到第2步.

## 3.2 快速关键帧识别和匹配

在场景的关键帧表达基础上，系统可以基于关键帧完成特征匹配。然而当关键帧的数目较大的时候，直接将输入图像与所有关键帧进行匹配还是很消耗时间。我们观察到每一帧输入图像都仅仅覆盖三维空间的一小部分区域，没有必要和整个关键集合匹配，只需要选择与之接近的、重叠区域较多的部分关键帧，就可以得到足够多的匹配点。因此，本文

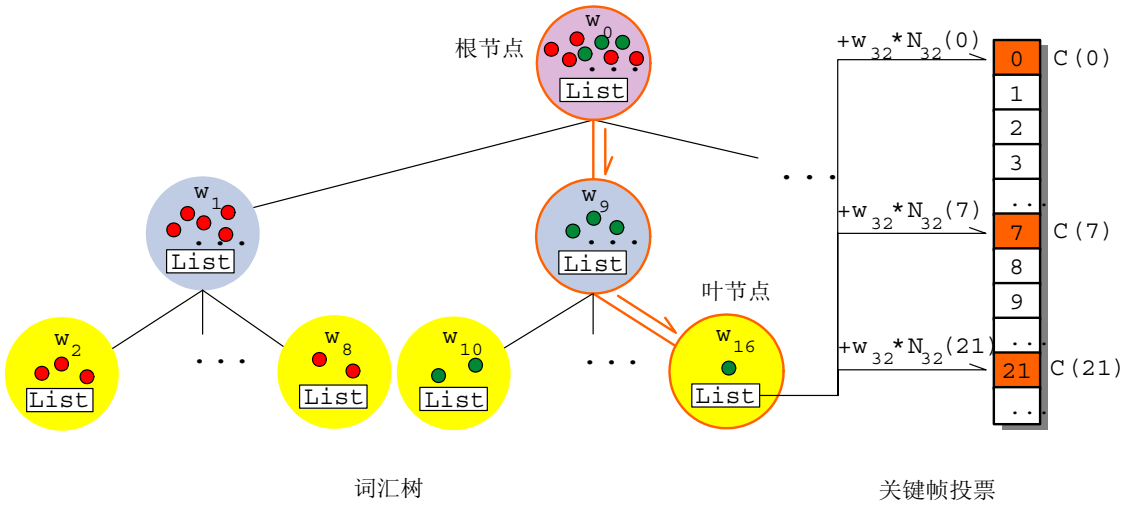



图 3.2 词汇树结构。关键帧包含的所有跟踪点描述量起初都在根节点，之后K-Means方法对其进行层次性划分；每一个节点都带有一个权值表示其区分度。

采用词汇树方法，选择与输入图像相似的关键帧成为候选关键帧，输入图像只和候选关键帧进行匹配。

### 3.2.1 词汇树构建

给定一个关键帧集合，系统在其包含的所有优先跟踪点上建立视觉词汇树，用以识别关键帧。词汇树的构建方式与之前的工作类似[115,116]，词汇表$V$组织成$l$层$b$叉树，根节点包含所有的优先跟踪点，然后通过层次K-Means聚类方法生成整棵树。每一次K-Means聚类方法都对当前层的树节点进行划分，生成的子类作为新的子节点，并继续对子节点进行划分；这个过程一直进行到预先指定的$l$层，或者节点中的跟踪点太少、误差太小。图3.2给出词汇树的大致结构。最终的词汇树有$|V|$个节点，每一个节点$i$包含一个SIFT描述量，是其中所有的优先跟踪点描述量的平均值。$i$还记录了其中的跟踪点所出现的所有关键帧集合$L_i$，并计算关键帧$k$在节点$i$中跟踪点数目$N_i(k)$。这些信息将会用来对关键帧的相似度进行投票。

节点$i$还带有一个权值$w_i$，表示其区分度，即该节点对于识别关键帧的可依赖度。一般来说，如果$i$中包含越多关键帧，表明$i$中的特征越普遍，其所能提供的区分信息就越少。$w_i$定义如下：

$$w_i = \log \frac{K}{|L_i|}, \tag{3.5}$$

**算法 3.3**: 候选关键帧识别

1. 初始化每一个关键帧$k$的匹配值 $C(k) = 0$ 。
2. 对每一帧实时输入图像，检测出$m$个SIFT特征点与词汇树的节点进行比较： 在树的每一层，对于每一个距离最接近的节点$i$，权值$w_i > \tau$， 对每一个 $k \in L_i$, $C(k) += N_i(k) \cdot w_i$.
3. 选择$C$最大的 $\mathcal{K}$ 个关键帧作为候选关键帧。

其中$K$是关键帧的数目。节点数$|V|$大致上由词汇树的分叉数 $b$和深度$l$决定。本文的实验通常选择20 $\sim$ 80个关键帧，每一个关键帧包含500 $\sim$ 1000优先跟踪点，如果重复的跟踪点不计数，总体包含6000 $\sim$ 60000 个独立三维点。词汇树一般设定$b = 10, l = 5$。

### 3.2.2 候选关键帧识别

在之前的工作中[115,116]，每幅图像都根据词汇树生成外观向量，向量的每一个元素对应树的一个节点。构建图像的外观向量，要将图像上的每一个特征点在词汇树中搜索最近的节点，从根节点出发，每一层都要与$b$ 个节点比较SIFT描述量，距离最小的节点的权值叠加到外观向量中相应的元素上，再继续和其子节点比较；输入图像上所有$m$个特征点都搜索之后，才生成图像外观向量。

基于图像外观向量，两幅图像之间的相似性可以直接比较，即外观向量之间的距离。如果在候选关键帧识别中直接利用这个方法，即使考虑到外观向量的稀疏性（只比较非零元素），计算复杂度也是$O(m \cdot K)$，随关键帧数目线性增长。

本文介绍了一种更高效的关键帧识别算法，见算法3.3。图3.2给出了关键帧投票方法的示意图，类似的投票方法也在Angeli等[139]的系统使用到。算法复杂度是$O(m \cdot \tilde{L})$，其中$\tilde{L}$是特征点在遍历词汇树过程中，访问的节点包含的平均关键帧数。值得一提的是，本文定义了截断阈值$\tau$来摒弃包含太多关键帧而识别能力太弱的节点。一般来说，越接近树根的节点，包含的跟踪点越多，关键帧也越多，因此更有可能被排除到计算之外。相反，叶节点通常对结果的影响更大。这种加权方法使得关键帧投票算法的运行时间几乎是恒定

的，不管有多少关键帧；而大部分的运算时间花在SIFT描述量的比较上，这个时间也是相对稳定的，因为输入图像上的特征点数 $m$不会有太大的变化。

### 3.2.3 基于关键帧的两遍匹配方法

确定输入图像相关的候选关键帧之后，就要进行特征匹配。要鲁棒而高效地求解摄像机参数，匹配的二维-三维对应点需要满足三个条件：1) 足够多；2) 最大数目应该被控制，因为太多的对应点不必要且增加计算量；3) 在三维空间均匀分布。之前的方法通常忽略后两个条件，本文的两遍匹配方法会同时考虑三个条件，保证摄像机参数求解结果的稳定和精确。

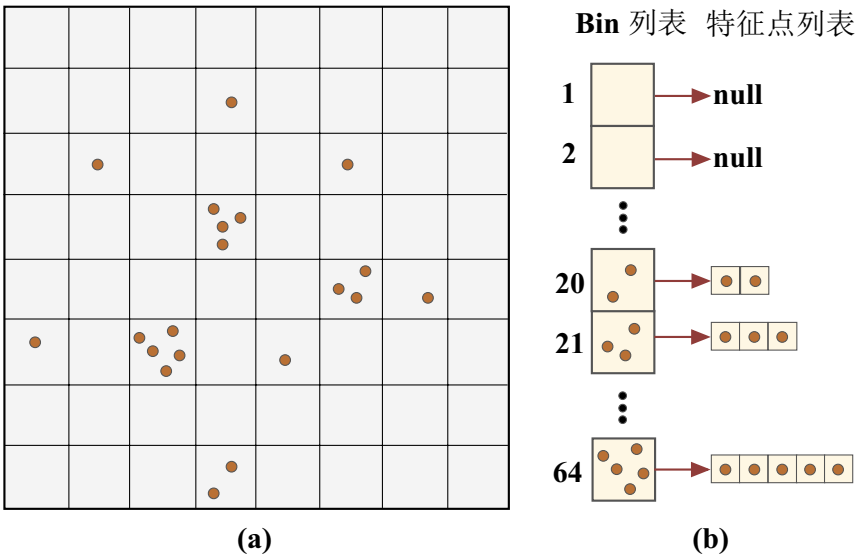



图 3.3 匹配Bin。(a) 图像被均匀划分成64个Bin；(b) 排序之后的Bin列表，每个Bin包含一个特征点列表，特征点按DoG值降序排列。

基于关键帧的特征匹配目标是找出输入图像$\tilde{I}_t$和候选关键帧之间的公共SIFT特征点。为使距离计算更加高效，系统使用64维度的SIFT描述量。虽然64维描述量的区分度比标准的128维小，但是实验表明性能已经足够，因为关键帧表达减少了相似结构的出现，同时每一个关键帧包含的特征点也比较少，一般在1000左右。

为了使匹配的特征点分布更均匀，系统将图像划分成64个Bin，表示为$\{B_i|i = 1, 2, ..., 64\}$，见图3.3。每一个Bin包含一个特征点列表，即图像坐标在这个Bin内的特征点，按照DoG值降序排列。理想情况下，如果每一个Bin贡献一个匹配特征，正好可以得

(a) (b) (c)

图 3.4 基于关键帧的两遍匹配结果。(a) 选择的候选关键帧；(b) 第一遍匹配只得到23个匹配跟踪点；(c) 第二遍匹配之后，共有43个匹配跟踪点；新增加的匹配点用红色表示。

到64个特征点匹配，足以求解摄像机参数。现实应用中，常有一些Bin不能提供特征匹配。因此，本文采用两遍匹配算法（算法3.4），尽可能找到足够多的特征点匹配，同时保证这些特征点均匀分布。

在系统运行的初始化时，每一个关键帧包含的跟踪点都组织成KD-Tree。输入图像上的64个Bin组成Bin列表，按照Bin中的已经匹配成功的特征点数目升序排列，特征点匹配越多的Bin，匹配优先级越低。在匹配的初始阶段，所有的Bin都不含有特征点匹配，因此按照在图像中的出现顺序，从上到下，从左到右进行匹配。对$B_i$中每一个SIFT特征点$\mathbf{x}_j$，系统同样用ANN库从候选关键帧$k$中搜索十个最接近的跟踪点，表示为$\{\mathcal{N}_s^k(\mathbf{x}_j)|s=1,...,10\}$；然后用2NN判断规则[14]确定匹配的可信度：

$$c=\frac{||\mathbf{p}(\mathcal{N}_1^k(\mathbf{x}_j))-\mathbf{p}(\mathbf{x}_j)||}{||\mathbf{p}(\mathcal{N}_2^k(\mathbf{x}_j))-\mathbf{p}(\mathbf{x}_j)||}. \tag{3.6}$$

$\mathbf{p}$表示特征点的描述量，表示$c$衡量了特征点的区分度。如果$c<\varepsilon$（实验中$\varepsilon$设为0.7），$\mathcal{N}_1^k(\mathbf{x}_j)$ 就被当作候选正匹配，并停止$B_i$的匹配。否则，继续对$B_i$中剩余的特征点进行匹配，直到得到一个候选正匹配，或者全部特征点都匹配完毕。这个匹配过程是第一遍匹配，即算法3.4的第2步。第一遍匹配得到的特征点匹配往往含有一些误匹配，通过RANSAC方法估计$\tilde{I}_t$和候选关键帧之间的基础矩阵，可以将误匹配剔除，见算法3.5。

第一遍匹配之后，如果特征点匹配的数目还小于一定的阈值$N$（实验中$N$设为50 ∼ 70），系统要进行第二遍匹配。此时，Bin列表可以按照其中特征点匹配的个数升序排列。候选关键帧中可能存在相似结构，许多特征点不能满足2NN判断规则，系统将结合对极几何约束，重新匹配这些特征点。

**算法 3.4**: 基于关键帧的两遍匹配方法

1. 对每一个提取的SIFT特征点$\mathbf{x}_j$，令 $C^1(\mathbf{x}_j)=0$， $C^2(\mathbf{x}_j)=0$。

2. 对 $i=1,...,64$: // 第一遍匹配

   对每个特征 $\mathbf{x}_j \in B_i$ & $C^1(\mathbf{x}_j)=0$,

   对 $k=1,...,\mathcal{K}$:

   从候选关键帧$k$找到最近的十个特征点 $\{\mathcal{N}_s^k(\mathbf{x}_j)|s=1,...,10\}$，并令$C^1(\mathbf{x}_j)=1$。如果$\mathcal{N}_1^k(\mathbf{x}_j)$满足2NN判断规则，停止当前Bin的匹配。否则，继续匹配直到所有$C^1(\mathbf{x}_j)=1$。

3. 利用得到的特征匹配点估计输入图像和候选关键帧之间的基础矩阵，同时剔除误匹配。如果已经得到$N$个正匹配，停止特征匹配，否则继续。

4. 对 $i=1,...,64$: // 第二遍匹配

   对未匹配成功的特征点 $\mathbf{x}_j \in B_i$ & $C^2(\mathbf{x}_j)=0$,

   对 $k=1,...,\mathcal{K}$:

   利用对极几何得到过滤的最近特征点$\{\tilde{\mathcal{N}}_s^k(\mathbf{x}_t)|s=1,...\}$，并令$C^2(\mathbf{x}_j)=1$。如果 $\tilde{\mathcal{N}}_1^k(\mathbf{x}_j)$ 满足2NN判断规则，并且 $||\mathbf{p}(\mathbf{x}_t)-\mathbf{p}(\tilde{\mathcal{N}}_1^k(\mathbf{x}_t))|| < \varsigma$，停止当前Bin的匹配。否则，继续特征匹配，直到所有的$C^2(\mathbf{x}_j)=1$.

5. 如果已经得到$N$个正匹配，停止特征匹配，否则继续。

6. 重复步骤2-5，直到得到$N$个匹配点，或者所有的SIFT特征点都遍历完毕。

在第一遍匹配中，系统估计了输入图像$\tilde{I}_t$和候选关键帧$k$之间的基础矩阵。对于特征点$\mathbf{x}_j$，系统通过对极几何过滤其候选匹配集合$\{\mathcal{N}_s^k(\mathbf{x}_j)|s=1,...,10\}$，仅保留到$\mathbf{x}_j$对应的极线距离小于2个像素的匹配点，表示为$\{\tilde{\mathcal{N}}_s^k(\mathbf{x}_j)|s=1,...\}$。如果$\mathbf{x}_j$确实存在一个匹配，则很可能在新的候选匹配集合中。因此，如果$\tilde{\mathcal{N}}_1^k(\mathbf{x}_j)$ 满足一下2NN判断规则：

$$\frac{||\mathbf{p}(\tilde{\mathcal{N}}_1^k(\mathbf{x}_j))-\mathbf{p}(\mathbf{x}_j)||}{||\mathbf{p}(\tilde{\mathcal{N}}_2^k(\mathbf{x}_j))-\mathbf{p}(\mathbf{x}_j)||} < \varepsilon, \tag{3.7}$$

**算法 3.5**: 利用RANSAC方法求取基础矩阵过滤误匹配

**目标：** 给定输入图像$\tilde{I}_t$和关键帧$k$的特征点匹配集合$\mathcal{F}=\{(x_0,x_0^k),(x_1,x_1^k),\ldots\}$，确定最优正匹配集合$\mathcal{F}'$。

1. 设$\mathcal{F}'=\emptyset$，$k=0$。
2. 从$\mathcal{F}$中随机选择八个匹配点，计算基础矩阵$F_k$（第1.1.1节）。
3. $\mathcal{F}_g=\{(x_i,x_i^k),\|x_i^T F_k x_i^k\|<\mu\}$（实验中$\mu=2.0$）。
4. 如果$\|\mathcal{F}_g\|>\|\mathcal{F}'\|$，$\mathcal{F}'=\mathcal{F}_g$。
5. 如果 $k<T$（T是最大循环次数，实验设为30），$k=k+1$，转到第2步。

而且$||\mathbf{p}(\mathbf{x}_j)-\mathbf{p}(\tilde{\mathcal{N}}_1^k(\mathbf{x}_j))||<\varsigma$，$\varsigma$控制描述量之间的绝对距离（实验中设为500，只是一个比较宽松的约束），则认为$\tilde{\mathcal{N}}_1^k(\mathbf{x}_j)$是$\mathbf{x}_j$的特征点匹配。这个策略可以有效地增加特征点匹配数目，如图3.4。

以上步骤一直重复，直到获得$N$个特征点匹配，或$\tilde{I}_t$中所有的特征点都匹配完毕。最后，根据得到的二维-三维特征点匹配，系统还是利用第2.1节的优化方法求解摄像机参数。由于场景的三维特征点不在同一个平面上，不能再依据单应矩阵来估计摄像机参数的处置，而是直接估计变换矩阵$T$（公式1.3），然后从中分解出$R,t$的初始值。

## 3.3 系统实现和实验结果

### 3.3.1 并行框架

由于本章的实时三维跟踪系统比第2章中的基于标志的方法更加耗时，为了达到实时性能，对硬件和软件的具体实现都需要多考虑一些细节。

帧率和延时是实时系统两个重要的衡量标准。帧率通常指的是系统在一秒钟内所能处理的图像数目，延时指系统捕获一帧输入图像到最终虚实融合之间经过的时间。好的实时系统应该具有高帧率和低延时。表3.1展示了Campus实例中各个模块的运行时间，同样地，系统用SiftGPU提取SIFT特征点。由于各个模块是独立运行的，只要CPU和GPU具有足够

| 模块 | 运行时间（毫秒） |
| --- | --- |
| SiftGPU提取SIFT特征点 | $\approx 45$ |
| 识别候选关键帧 | $\approx 2$ |
| 基于关键帧的特征匹配 | $\approx 4 \times \mathcal{K}$ |
| 求解摄像机参数 | $\approx 5$ |
| 虚实合成 | $\approx 5$ |

表 3.1 基于关键帧的实时三维跟踪运行时间，$\mathcal{K}$是候选关键帧的数目。

的计算能力，系统的帧率由最耗时的模块决定，可以看出提取SIFT特征点的时间最长，因此系统的帧率在20fps左右。而系统的延时是各模块的运行时间之和，在80毫秒左右（实验中$\mathcal{K} = 4$），虽然明显比基于标志的实时三维跟踪延时长，但在实际应用中用户对100毫秒以内的延时都不会有明显感觉。

充分地利用并行计算能力，系统的延时还可以进一步减少。系统框架包含两个平行层次，帧内并行和帧间并行。所谓帧间并行即上述的模块间的并行，每个模块可以同时处理不同的输入图像，主要作用是提升系统的运行帧率，但是不能缩短延时。帧内并行则是对同一帧输入图像的并行处理，在本文的框架里，主要体现在基于关键帧的特征匹配上，因为候选关键帧之间是独立，输入图像和候选关键帧之间的匹配可以并行，只要CPU计算能力足够，就可以忽略候选关键帧数$\mathcal{K}$对延时的影响。

### 3.3.2 利用时序信息

可以看出，以上所述的基于关键帧的实时三维跟踪属于匹配跟踪（第1.2节），即使只提供系统单帧图像，也可以快速识别候选关键帧，完成特征匹配，并估计摄像机参数。不过，增强现实系统的输入主要是实时捕获的视频，完全可以利用时序上的信息，使跟踪更加稳定可靠。

针对候选关键帧识别，系统在确定$\tilde{I}_t$对应的候选关键帧时，利用前一帧$\tilde{I}_{t-1}$的信息。如果关键帧$I_k$与$\tilde{I}_{t-1}$有最多的公共特征点，那么$I_k$和$\tilde{I}_t$也很可能是相似的，所以首先将$I_k$作为$\tilde{I}_t$的候选关键帧，然后选择摄像机参数和$I_k$最接近的其余3个候选关键帧。当$\tilde{I}_t$和候选关键帧完成两遍匹配之后，与$\tilde{I}_t$具有最多共同特征点的关键帧又继续引导$\tilde{I}_{t+1}$的匹配。如果

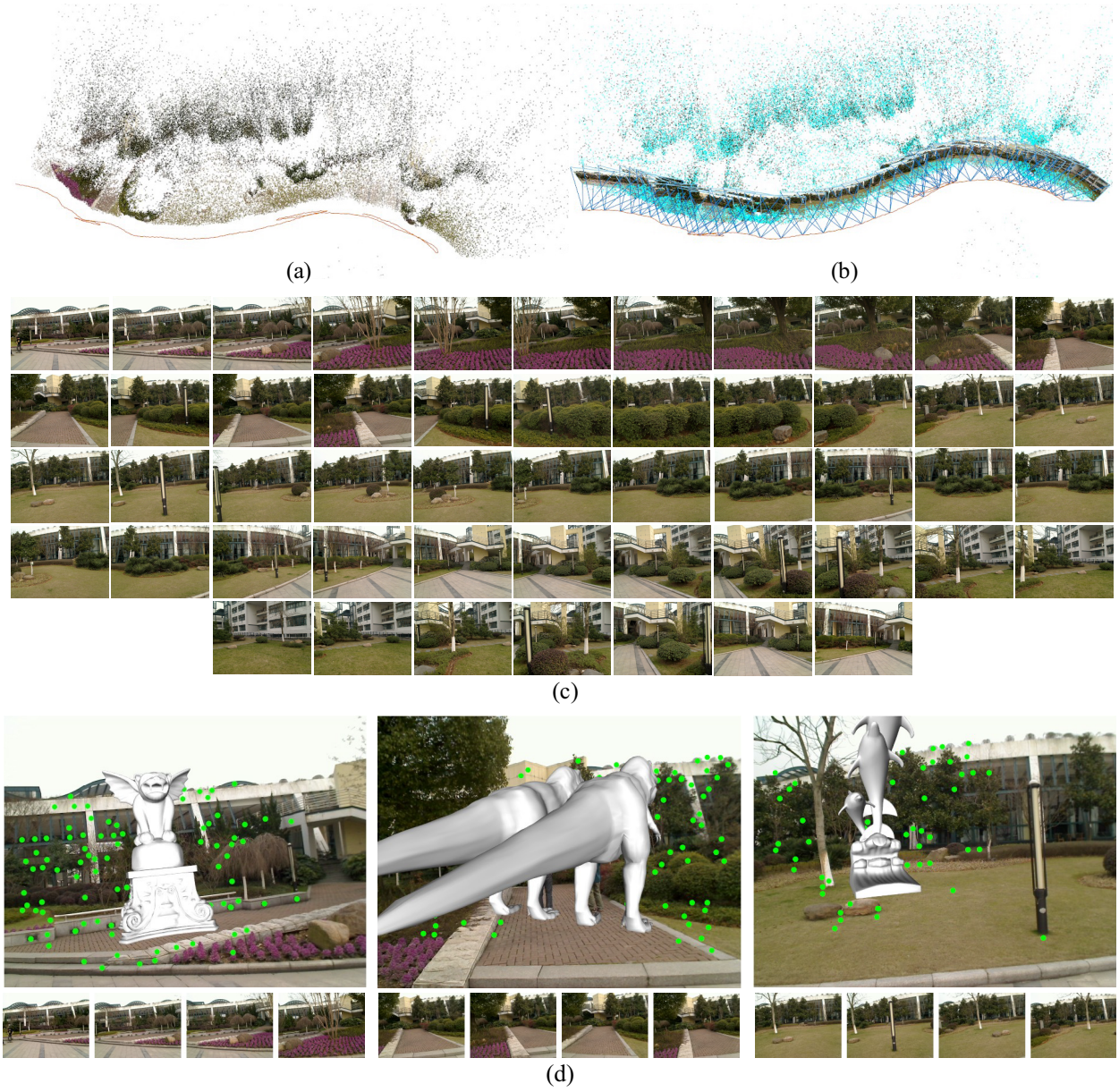



图 3.5 Campus实例。(a) 预处理重建的三维点云结构。(b) 在三维环境中观察关键帧，青色标注的点是关键帧所包含的三维点；(c) 关键帧图像；(d) 增强现实结果及其候选关键帧，绿色的点是过滤误匹配之后的特征点匹配。

摄像机运动过快，前一帧的信息事实上不能提供有用的参考，时序信息会导致跟踪失败，此时系统又用第3.2节的方法进行特征匹配。

另一方面，既然在$\tilde{I}_{t-1}$匹配成功的特征点已经获得相应的三维信息，自然也可以用来帮助当前帧的求解，如第2.2节，本文结合KLT方法为$\tilde{I}_t$补充时序的跟踪点。如果一个跟

踪点在$\tilde{I}_{t-1}$被匹配上，$\tilde{I}_t$却没有，那么就从跟踪点在$\tilde{I}_{t-1}$上的匹配特征点的位置出发，利用KLT方法跟踪其在$\tilde{I}_t$上的位置。

### 3.3.3 实验结果

本节的系统运行在桌面系统上，CPU是Intel(R) Core(TM)2 Quad CPU Q9550 @ 2.83GHz，显卡是GeForce GTX 275。系统的输入设备为罗技高清摄像头C905，输入视频分辨率是$640 \times 480$，捕获帧率在$20 \sim 30$fps之间。虚实融合采用OpenGL API，输出设备为普通液晶显示器。

**1. 大规模户外Campus实例**

图3.5展示在大规模的户外Campus场景中进行实时三维跟踪。这个实例对实时跟踪来说非常困难，因为场景的规模很大，而且具有许多重复的相似结构。图3.5(a)展示了重建的场景三维点云，共有$72,616$个三维点，预处理序列包含$1,913$帧图像。图3.5(c)是自动选择的关键帧图像，它们基本上覆盖了整个场景，如图3.5(b)。虽然彼此之间还有不少重复的三维点，但是三维数据的冗余度已经大大减少。

表3.2展示权值$\lambda$对关键帧选择结果的影响。可以看出，如果选择123个关键帧，可以包含95%以上优先跟踪点（三维点），但只要33个关键帧就可以保留42.16%以上优先跟踪点。本文的实验设$\lambda = 2.0$，可包含足够多的优先跟踪点。本文的关键帧选择算法实现十分高效，这个例子只需要11秒钟运算时间。

图3.6展示候选关键帧识别的执行性能，运行时间在2毫秒左右。与基于外观向量[115]的方法相比，本文的方法运行速度快许多，而且几乎与关键帧的数目无关。

为了进一步说明系统的有效性，本文还将基于关键帧的匹配方法与全局匹配方法进行比较。Skrypnyk等[86]为三维重建的全部点云建立KD-Tree，输入图像上的特征点直接和全局的KD-Tree比较，这种方法叫做全局匹配方法。全局匹配十分依赖特征点的全局区分度，然而随着场景变化，全局特征点的数量增加，彼此之间区分度不可避免地下降，会影响全局匹配的性能。图3.7比较了两种方法得到的特征点正匹配数目。关键帧方法产生的正匹配数目明显多于全局匹配，因此求解结果也更加稳定。

利用KD-Tree的全局匹配方法的算法复杂度是$O(\log M)$，$M$是全局特征点的数目。关键帧方法的复杂度是$\mathcal{K} \cdot O(\log m)$，$m$是关键帧上平均特征点数目，$\mathcal{K}$是关键帧数目。全局

| $\lambda$ | 关键帧数 | $E_c$ | $E_r$ | 优先跟踪点比例 |
|---|---|---|---|---|
| 0.1 | 123 | 0.020207 | 0.584265 | 95.7998% |
| 1.0 | 65 | 0.194271 | 0.142999 | 70.2325% |
| 2.0 | 51 | 0.289593 | 0.071128 | 58.4981% |
| 5.0 | 33 | 0.443603 | 0.022406 | 42.1656% |
| 10 | 24 | 0.558724 | 0.006362 | 31.1694% |
| 100 | 12 | 0.739066 | 0.000220 | 16.4674% |

表 3.2 关键帧选择算法的参数分析。在不同$\lambda$下，完备项$E_c$和冗余项$E_r$的值，以及关键帧包含的优先跟踪点占全部跟踪点的比例。

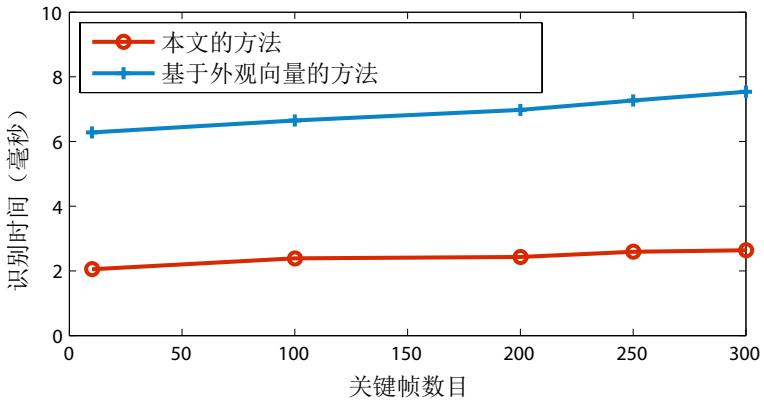



图 3.6 候选关键帧识别时间。基于外观向量的方法的计算时间与关键帧的数目成线性关系，而本文的方法几乎是恒定的。

方法的的计算时间随$M$增长，而关键帧方法的时间相对稳定，因为候选关键帧的数目固定，关键帧上的特征点数也相对稳定。在实验中，每一帧输入图像大致提取300个特征点，全局匹配的时间大约是45毫秒，关键帧方法的时间大约是16毫秒（$\mathcal{K}=4$）。

**2. 更多室内外跟踪结果**

图3.8展示室内的Cubicle实例。图3.8(a)是重建的$10,691$三维点，相应的19个关键帧展示在图3.8(b)和图3.8(c)中，包含57.3%的三维特征点。摄像头输入视频的噪声比较强，而且摄像头移动很快，有一定的运动模糊，这些因素给实时三维跟踪带来许多困难。对大多

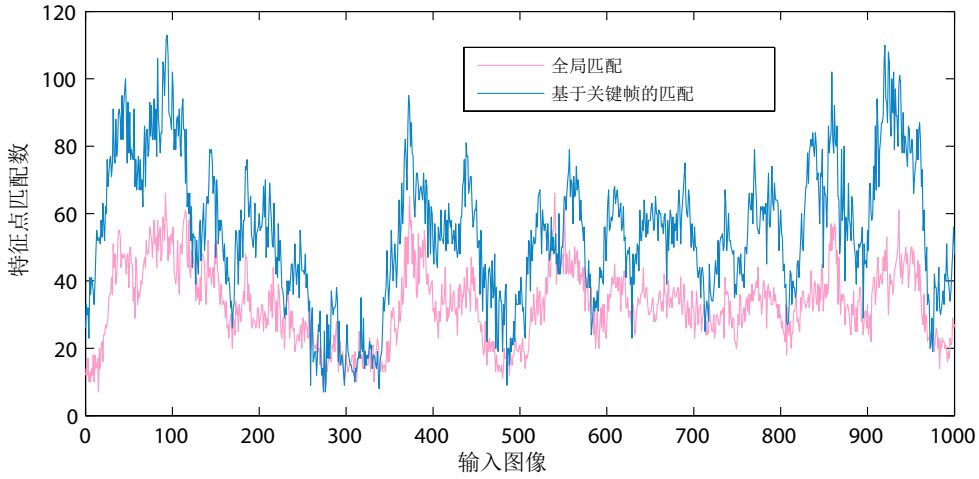



图 3.7 比较全局匹配和基于关键帧的匹配方法。即使没有利用时序信息补充匹配点，基于关键帧的方法也比全局匹配得到更多的特征点匹配。

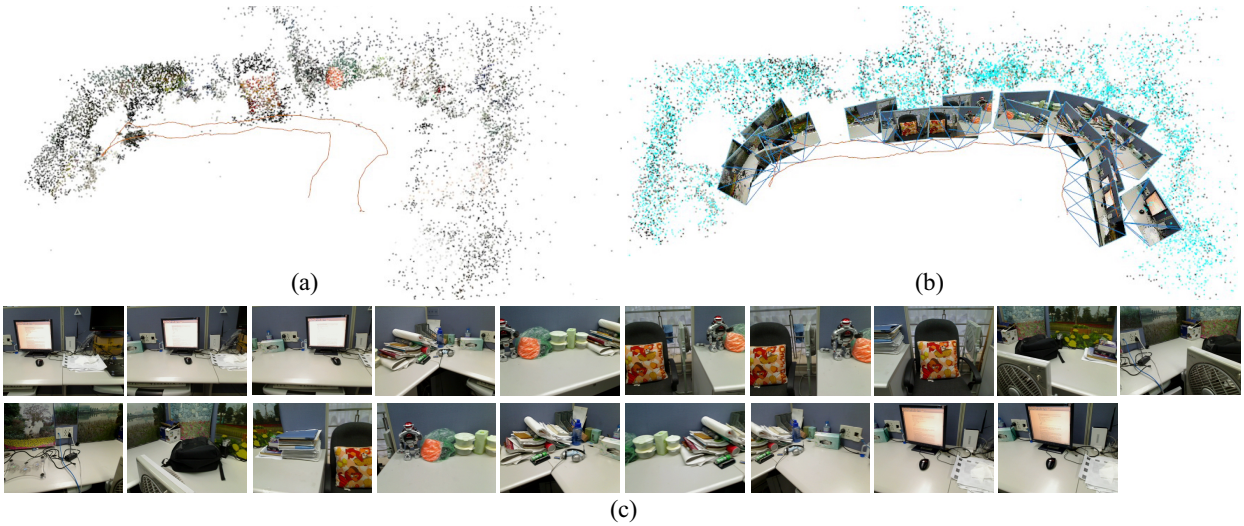



图 3.8 Cubicle实例的三维点云和关键帧。(a) 预处理阶段重建的三维点云；(b) 在三维环境中观察关键帧，青色标注的点是关键帧所包含的三维点；(c) 关键帧图像。

数输入图像来说，系统可以精确地选择相应的关键帧，求解摄像机参数，只有一些与预处理视频差距太大或模糊太厉害的图像跟踪失败。为了衡量跟踪结果的精度，系统在场景中加入增强现实的虚拟物体，如图3.9。从结果可以看出，虚拟物体的位置十分稳定，说明摄像机参数的精确。

图3.10展示另一个室外的Street实例。预处理序列包含3,385帧图像，摄像机沿着一


图 3.9 Cubicle实例的实时三维跟踪结果。第一行展示了输入图像的实时增强现实结果，第二行是相应的四个候选关键帧。绿色的点是过滤误匹配之后的特征点匹配。

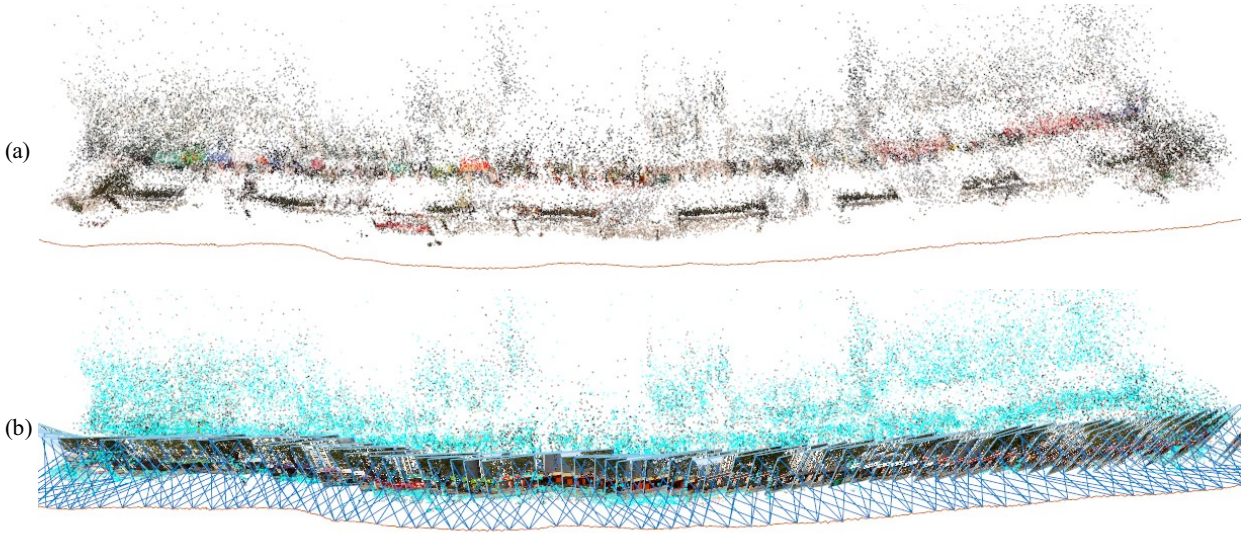
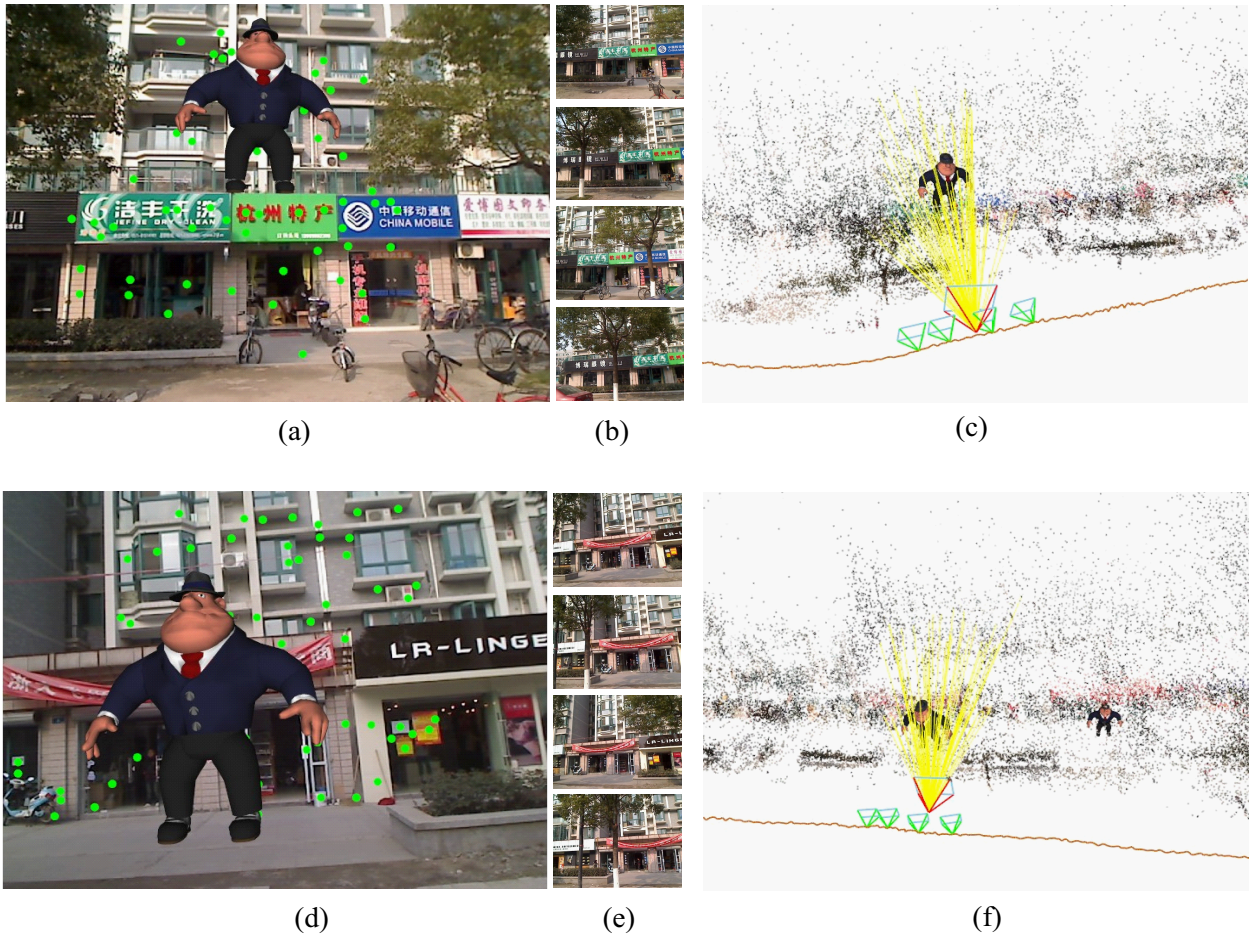



图 3.10 Street实例的三维点云和关键帧。(a) 预处理阶段重建的三维点云；(b) 在三维环境中观察关键帧，青色标注的点是关键帧所包含的三维点。

条道路移动，捕获了一系列建筑，总的移动距离大概是200米。图3.10(a)展示了重建的72,004个三维点，图3.10(b)是选择的76个关键帧。实时三维跟踪的结果在图3.11，系统同样加入增强现实的虚拟物体。

### 3. 与PTAM的比较

Klein等[118,119]结合在线集束调整技术和并行计算，避免了预处理的三维重建阶段，可




(a) (b) (c)

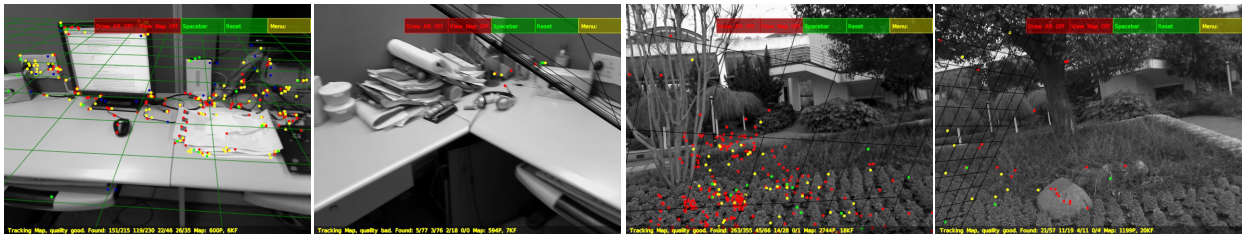
(d) (e) (f)

图 3.11 Street实例的实时三维跟踪结果。(a) 输入图像的增强现实结果；(b) 图像(a)的候选关键帧；(c) 在三维视图中查看跟踪结果；(d-f) 另一帧输入图像的增强现实结果和三维视图。

以直接在场景中展开三维跟踪。这个策略对于小型的桌面场景来说非常有效，但是不适用于大规模的场景，因为集束调整同样需要很长时间才可能重建出场景的三维，否则跟踪很不稳定。本文系统和开源软件PTAM[①]做了比较。为了给PTAM充分的集束调整优化时间，输入的视频帧率调整为5fps；即便如此，PTAM仅在Cubicle实例的前半段成功跟踪，而两个室外场景都失败了。图3.12展示了PTAM跟踪Cubicle和Campus实例的结果。

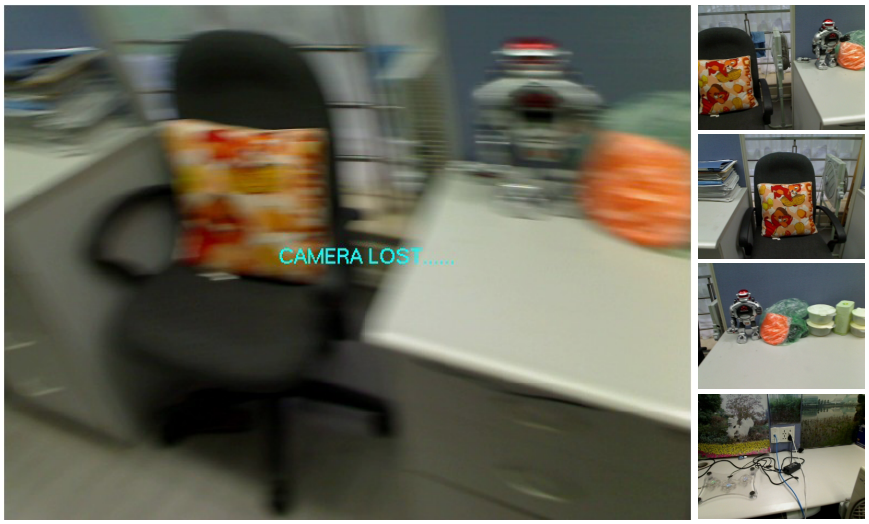
当然，本文的系统也会出现跟踪失败的情形，一种情况是摄像机的视角与预处理序列相差太大，无法找到相似的候选关键帧；还有一种情况是场景光照变化太大或摄像机运动

[①]http://www.robots.ox.ac.uk/gk/PTAM/



(a) (b) (c) (d)

图 3.12 PTAM跟踪Cubicle和Campus实例的结果。(a-b) Cubicle的两帧PTAM跟踪结果；(c-d) Campus的两帧PTAM跟踪结果。由于场景中特征点的丰富和摄像机的快速移动，PTAM的跟踪结果很不稳定。




(a) (b)

图 3.13 三维跟踪失败的情形。(a) 输入图像；(b) 识别的候选关键帧。因为输入图像的外观由于运动模糊发生了很大的变化，无法正常匹配特征点，也就不能求解摄像机参数。

导致的模糊，使输入图像的外观发生很大的变化，匹配的特征点数太少，无法求解摄像机，如图3.13。

## 3.4 小结

本章在实时三维跟踪系统框架下实现了基于关键帧的实时三维跟踪，并在大规模的室外场景中实现了增强现实。

为了在更大的自然场景中进行实时三维跟踪，本文提出利用关键帧来表达三维场景，并基于关键帧进行特征点匹配和三维跟踪。以往方法的关键帧通常都是手工选择，本文将

关键帧选择作为一个优化问题求解，提出的目标能量函数充分考虑三维跟踪的具体要求，并通过贪婪法快速地优化目标函数，自动得到关键帧。其次，本文结合图像识别方法，通过关键帧投票来确定与输入图像相似的候选关键帧，大大降低了输入图像与关键帧匹配特征点的计算代价。整个系统的通用性和稳定性在实验结果中得到验证。

# 4 基于实时三维跟踪的增强现实系统

前两章讨论的是实时三维跟踪技术，试图解决增强现实中几何一致性的虚实物体配准问题，本章将在此基础上，讨论基于实时三维跟踪的增强现实系统。首先介绍纯旋转相机下的增强现实系统，由于在这种情况下，场景的层次结构一般比较明显，除了利用三维跟踪保证虚实物体的几何配准，还可以通过层次分割技术处理虚实物体的遮挡关系。其次，本章还将以虚拟装机为例展示自由运动相机下的增强现实系统。

## 4.1 纯旋转相机下的增强现实系统

纯旋转相机的三维跟踪显然相对简单，但是在现实中的确有许多情况满足应用条件，如虚拟演播室、监控摄像头、视频聊天等。此时，由于场景一般具有明显的层次结构，虚实遮挡问题可以转化成双层分割问题，只要将前景和背景内容分割开，按照背景、虚拟物体、前景的顺序绘制场景，就可以得到正确的遮挡效果。

双层分割问题在计算机视觉领域是广泛研究的课题。交互双层分割方法有Video SnapCut[125]、LIVEcut[126]等，实时双层分割的方法有背景消减[128]、运动分割[129,130]、背景分割[133]、机器学习分割[134]等。交互分割方法侧重于分割结果的精确性，需要用户的大量交互，由于分割算法充分利用人工约束和图像信息，计算效率一般不能满足实时要求。而实时自动分割方法侧重运行效率，分割结果相对比较差，而且摄像机需要保持静止，背景也不能有太大变化。本系统需要一种新的实时双层分割方法。

解决虚实遮挡问题，最可靠的方法是利用现实场景的精确三维模型。三维重建虽然已经取得很大的进展，但是对于绝大多数应用场景来说，现有的技术还无法自动完成这个任务。当然可以采用大量人工手段去重建三维，但是代价太高，限制了增强现实的应用。事实上，我们可以对场景做出一定假设，简化遮挡关系的处理，在保证增强现实效果的前提下，提升虚实融合的真实感。本节工作面对的增强现实必须满足以下三个条件：

1. 摄像机的位置保持不变，只有旋转运动；

2. 场景可以分成前景和背景两个层次，增强现实内容主要展示在单个层次上；

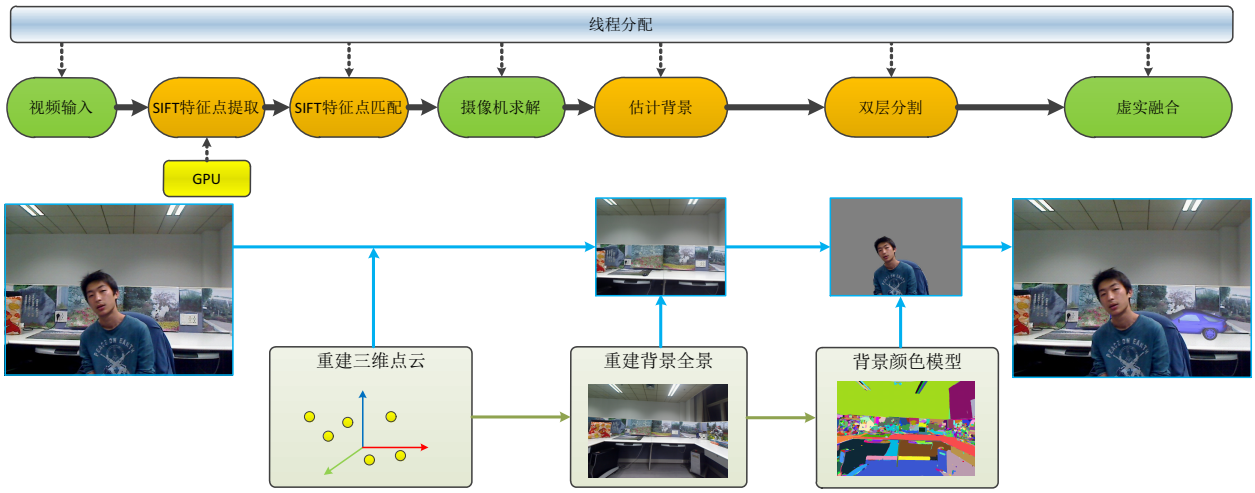



图 4.1 纯旋转相机下的增强现实系统。系统基于实时三维跟踪框架，主要的模块是估计背景和双层分割。

3. 场景的背景具有丰富的特征信息，且基本保持静止。

图4.1展示了整个系统的流程，同样分为预处理和实时两个阶段。在预处理阶段，系统首先要恢复背景模型，包括背景的全景图和其中的SIFT特征点。预处理的视频序列包括摄像头运动可能的旋转角度，因此在实时阶段，当摄像头在场景中转动的时候，输入图像的SIFT特征点和背景全景图自动配准，就可以估计输入图像对应的背景，完成双层分割。虽然系统可以得到输入图像的背景内容，但是难免会有配准误差，在之前的方法中，不对齐的背景信息对分割结果有非常大的影响。考虑到这个因素，本文提出对背景颜色计算一个局部的颜色高斯模型。总的来说，本文的实时双层分割方法主要有以下三个方面的改进：

1. 利用实时三维跟踪的摄像机参数结果，估计输入图像的背景信息；

2. 实时分割算法引入背景的局部高斯颜色模型，并压制背景颜色反差；

3. 利用多分辨率实现加速分割算法，并实现多种增强现实效果。

### 4.1.1 背景全景图创建

系统的预处理阶段与基于关键帧的三维跟踪方法类似，不同在于本章的系统还要重建背景的全景图。纯旋转摄像机的全景拼图是一个成熟的研究领域，Brown等的方法[141]可

(a) (b) (c)

图 4.2 背景全景图创建和实时配准。(a) 背景全景图；(b) 输入图像；(c) 配准估计的背景图像。背景全景图上的黄色四边形所包围的区域是输入图像对应的背景，需要经过单应矩阵映射成当前视角的背景图像。

以从散乱的图像序列中自动提取SIFT特征点，检测图像之间的重叠区域，完成全景图的拼接。摄像机模型仍旧是标准的针孔相机模型，但在纯旋转运动下，摄像机的位置固定，变换矩阵是：

$$X' = TX = \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} X. \tag{4.1}$$

对于输入的图像序列$\hat{I} = \{I_i | i = 1, 2, ..., n\}$，系统采用Hartley的自定标方法[142]和Triggs的集束调整优化方法[91]解得图像对应的旋转矩阵$R_i$，同时得到一系列三维跟踪点$\mathcal{X}$。因为摄像机的位置是固定的，所以系统无法求解三维跟踪点的深度，$\mathcal{X}$的$Z$坐标全部默认为1.0。假设一个三维点$X$投影到图像$I_i, I_j$的二维坐标是$x_i, x_j$，那么

$$x_i = KR_iX, \tag{4.2}$$

$$x_j = KR_jX. \tag{4.3}$$

其中$R_i, R_j$即是$I_i, I_j$的旋转矩阵。将公式4.3代入公式4.2，可以得到

$$x_i = KR_iR_j^{-1}K^{-1}x_j. \tag{4.4}$$

可以看出，$KR_iR_j^{-1}K^{-1}$就是从$I_j$到$I_i$的单应矩阵。

系统将选择正对场景中心的视角$I_r$作为参考视角，将其它视角上的图像映射到参考视角上。$I_i$到$I_r$的单应矩阵$H_{ir}$可以根据公式4.4轻易得到，即$H_{ir} = KR_rR_i^{-1}K^{-1}$。$I_i$中的像素坐标经过$H_{ir}$变换后得到的新坐标通常不是整数，像素的颜色根据线性插值方法分摊到$I_r$中相邻的四个像素上，映射到$I_r$中同一位置的像素颜色直接混合在一起。最终，系统可以得到背景的全景图，如图4.2(a)。通过上述的计算，系统得到基本的背景模型：

- 静止背景的全景图；
- 参考三维跟踪点$\mathcal{X}$的三维位置和SIFT描述量；
- 中心视角的旋转矩阵$R_r$。

### 4.1.2 实时背景配准估计

在实时阶段，首先要做的是估计输入图像对应的背景图像。输入图像上的SIFT特征点与参考三维跟踪点进行匹配，根据2NN判断规则，系统可以得到一系列二维-三维对应点$(x_i, X_i)$，并优化投影误差求解出当前帧的旋转矩阵$R_t$：

$$R_t = \arg\min_R \sum_i ||KRX_i - \mathbf{x}_i||^2 \tag{4.5}$$

然后，可以计算从参考视角到当前视角的单应矩阵：$H_{rt} = KR_tR_r^{-1}K^{-1}$。根据$H_{rt}$就可以把全景图映射到当前视角，估计出背景图像。图4.2(b)和图4.2(c)显示了一帧输入图像及其估计的背景。

### 4.1.3 实时双层分割

设$I$是当前的输入图像，$I^B$表示$I$对应的背景图像。对于$I$中的每一个像素$i$，像素颜色值用$I_i$表示。本文使用的是RGB颜色空间，每个通道的颜色取值范围是$[0, 255]$。双层分割的目标是为每个像素$i$估计二元值$\alpha_i$，如果$i$是前景像素则$\alpha_i = 1$，反之$\alpha_i = 0$，因此双层分

割其实是二元标记问题，标记值$\alpha = \{\alpha_i\}$可以通过最小化Gibbs能量函数$E(\alpha)$获得：

$$E(\alpha) = \sum_{i \in V} E_d(\alpha_i) + \lambda \sum_{(i,j) \in \mathcal{E}} E_s(\alpha_i, \alpha_j), \tag{4.6}$$

其中$V$是$I$的像素节点集合，$\mathcal{E}$是$I$中的相邻像素的边集合。$E_d$是数据项，代表$i$的标记取$\alpha_i$的可能性；$E_s$是像素边$(i,j)$之间的平滑项，当$i,j$的标记$\alpha_i$和$\alpha_j$取不同的值时，$E_s$会施加一定的惩罚值。对于二元分割问题来说，最重要的就是数据项和平滑项的定义，而最终的优化都可以通过图割算法[131]完成。

#### 4.1.3.1 数据项定义

根据第4.1节的方法，可以估计出$I$对应的背景图像$I^B$，数据项的定义依赖$I^B$的准确估计。由于有了背景信息，数据项可以是$I$与$I^B$之间的颜色差，即背景消减方法。考虑到前景物体和摄像机运动对场景光照的影响，本文利用图像上的特征点和所匹配的三维跟踪点之间的亮度比例的平均值，将$I^B$的亮度做一个全局调整。由于$I$与$I^B$之间不可避免的配准误差，比较像素颜色的时候，系统总是在$I^B$的局部窗口内搜索与$I$中颜色最接近的像素：

$$\mathcal{S}_i = \min_{j \in W(i)} ||I_i - I_j^B||, \tag{4.7}$$

其中$W(i)$就是以$i$为中心的方形局部窗口，实验中设为$5 \times 5$。虑及效率，RGB图像先转成灰度图。从亮度差$\mathcal{S}_i$，可以初步定义$I$中像素属于前景或背景的概率$L_S$：

$$\begin{aligned} L_S(I_i|\alpha_i = 0) &= \frac{\mathcal{S}_i^2}{\mathcal{S}_i^2 + \delta^2}, \\ L_S(I_i|\alpha_i = 1) &= \frac{\delta^2}{\mathcal{S}_i^2 + \delta^2}, \end{aligned} \tag{4.8}$$

其中$\delta$其实是一个软阈值，决定像素$I_i$属于前景还是背景。如果$S_i > \delta$，那么$L_S(I_i|\alpha_i = 0) > 0.5 > L_S(I_i|\alpha_i = 1)$，即像素$i$更可能是前景。这种背景消减方法得到的结果往往很不精确，尤其是前景和背景颜色相近时，系统需要更加鲁棒的颜色模型来建立数据项。

双层分割算法常常使用高斯混合模型[133,143]。高斯混合模型的经典做法是分别为前景背景颜色建模，背景颜色的训练样本就是背景图像，而前景颜色可由用户交互指定或自动检测。于是，背景的颜色模型$p(I_i|\alpha_i = 0)$可以定义如下：

$$p_g(I_i|\alpha_i = 0) = \sum_{k=1}^{K_B} w_k^B N(I_i|\mu_k^B, \Sigma_k^B), \tag{4.9}$$

图 4.3 高斯混合模型的分割结果。因为背景的颜色分布非常复杂，而且含有与前景相近的颜色，经典的高斯混合模型并不适用。

其中$w_k^B$是背景高斯混合模型的第$k$个高斯模型的权值，$\mu_k^B$和$\Sigma_k^B$是其对应的颜色均值和协方差矩阵。同样地，前景高斯混合模型定义为：

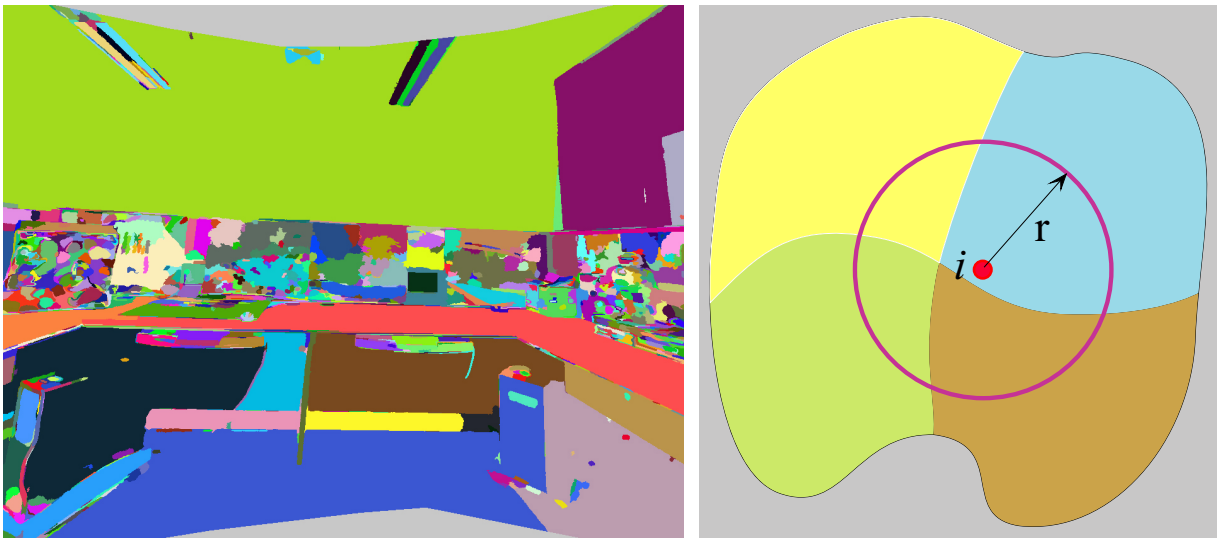
$$p_g(I_i|\alpha_i = 1) = \sum_{k=1}^{K_F} w_k^F N(I_i|\mu_k^F, \Sigma_k^F), \tag{4.10}$$

其中$w_k^F$是前景高斯混合模型的第$k$个高斯模型的权值，$\mu_k^F$和$\Sigma_k^F$是其对应的颜色均值和协方差矩阵。

如果前景和背景颜色分布比较简单，而且彼此之间差异大，那么上述的高斯混合模型可以达到很好的效果。但是我们发现现实应用中，前景和背景的颜色总有相近的组成，高斯混合模型无法将其区分开。还有，在经典的高斯混合模型中，前景和背景的高斯模型个数应该保持相近，否则它们计算出来的概率值数量级都不一样，然而现实环境中背景的颜色往往比前景复杂，尤其是户外环境。如图4.3中的背景有丰富的纹理信息，高斯混合模型很难处理这种复杂的环境。

经典的高斯混合模型是一种全局性的颜色模型，在整幅图像上估计颜色分布，为了处理更复杂的情况，本文考虑使用局部方法。Zhong等[144]提出基于颜色区域的高斯颜色模型，使用K-Means方法对图像颜色进行聚类，每一个类别估计一个高斯模型，类别中的像素都对应这个高斯模型。Bai等[125]也利用了局部的颜色模型改进视频分割结果，出于相同的考虑，本文也提出一种局部的背景颜色模型。

在预处理阶段，系统用Mean Shift[145]方法分割背景全景图，如图4.4(a)。全景图上的




(a) (b)

图 4.4 背景颜色的局部高斯模型。(a) 背景全景图被Mean Shift分割成多个区域，每一个分割区域用不同颜色标识；(b) 中心像素$i$的局部窗口内有四个独立的高斯分布。

每一个分割区域$S_k$，都可以估计一个高斯分布$\{N(\mu_1^b, \Sigma_1^b), N(\mu_2^b, \Sigma_2^b), \ldots N(\mu_i^b, \Sigma_i^b)\}$。背景全景图上的每一个像素$i$所属的分割区域用$m_i$表示，即$i \in S_{m_i}$。为了使背景颜色模型在边缘的像素上更加鲁棒，每个像素的颜色模型包括相邻的窗口内的所有高斯模型，这个局部窗口的大小固定为半径$r = 3$的圆形区域，如图4.4(b)，在中心像素的周围有四个高斯模型。这个分割、建模、采样的过程完全可以在预处理阶段完成。假设局部区域内有$l$个高斯模型，那么本文的背景颜色模型定义为:

$$L_G(I_i|\alpha_i = 0) = 1 - \max_{j=1}^{l} N(I_i|\mu_{m_j}^b, \Sigma_{m_j}^b). \tag{4.11}$$

由于前景物体通常是运动变化的，系统很难确定每个像素对应的局部区域，只能用全局的高斯混合模型。前景的训练样本颜色在第一帧由用户手工标注，本文利用EM方法（详见算法4.1）估计高斯混合模型：$\{N(\mu_1^f, \Sigma_1^f), N(\mu_2^f, \Sigma_2^f), \ldots N(\mu_{K_f}^f, \Sigma_{K_f}^f)\}$。前景颜色模型定义为：

$$L_G(I_i|\alpha_i = 1) = 1 - \sum_{k=1}^{K_f} w_k^f N(I_i|\mu_k^f, \Sigma_k^f), \tag{4.12}$$

**算法 4.1**: 估计前景高斯混合模型的EM方法

**输入**：$P$个像素颜色样本$\{I_1, I_2, \ldots, I_P\}$，每个颜色是一个三维RGB向量；

**输出**：$M$个高斯分布$\{N_1(\mu_1, \Sigma_1), N_2(\mu_2, \Sigma_2), \ldots, N_M(\mu_M, \Sigma_M)\}$，权重$\{w_1, w_2, \ldots, w_M\}$，和相对应的像素集合$\{C_1, C_2, \ldots, C_M\}$；

1. **初始化**：利用K-Means算法将像素样本聚成$M$类，$\{C_1, C_2, \ldots, C_M\}$，每一个类估计相应的高斯分布$\{N_1, N_2, \ldots, N_M\}$，且$w_i = \frac{\|C_i\|}{P}$，$s' = 0$；

2. **E步骤**：

$$s = \sum_{1}^{P} log(\sum_{1}^{M} w_m * N_m(I_p));$$

3. **退出条件**：如果$\|\frac{s}{P} - s'\| < \xi$（实验中$\xi$取0.0001），退出；否则，$s' = \frac{s}{P}$；

4. **M步骤**：对于$i = 1, \ldots, M$,

$$e_i = \sum_{1}^{P} \frac{w_i * N_i(I_p)}{\sum_{1}^{M} w_m * N_m(I_p)},$$

$$w_i = \frac{e_i}{P},$$

$$\mu_i = \frac{\sum_{1}^{P} I_p * \frac{w_i * N_i(I_p)}{\sum_{1}^{M} w_m * N_m(I_p)}}{e_i},$$

$$\Sigma_i = \frac{\sum_{1}^{P} (I_p - \mu_i)(I_p - \mu_i)^T * \frac{w_i * N_i(I_p)}{\sum_{1}^{M} w_m * N_m(I_p)}}{e_i}$$

5. **退出条件**：如果循环次数太多，退出；否则回到第2步。

实验中的前景物体颜色一般相对简单，因此实验中$K_f$设为5。

最后将背景消减和局部颜色模型结合起来，数据项定义如下：

$$E_d(I_i|\alpha_i) = \begin{cases} L_S(I_i|\alpha_i) & if \ \ c_i > 0 \\ 0.5 & otherwise, \end{cases} \tag{4.13}$$

其中$c_i$判断背景消减和局部颜色模型的判断是否一致，定义为；

$$c_i = (L_S(\alpha_i = 0) - L_S(\alpha_i = 1)) \cdot (L_G(\alpha_i = 0) - L_G(\alpha_i = 1)).$$

(a) (b)

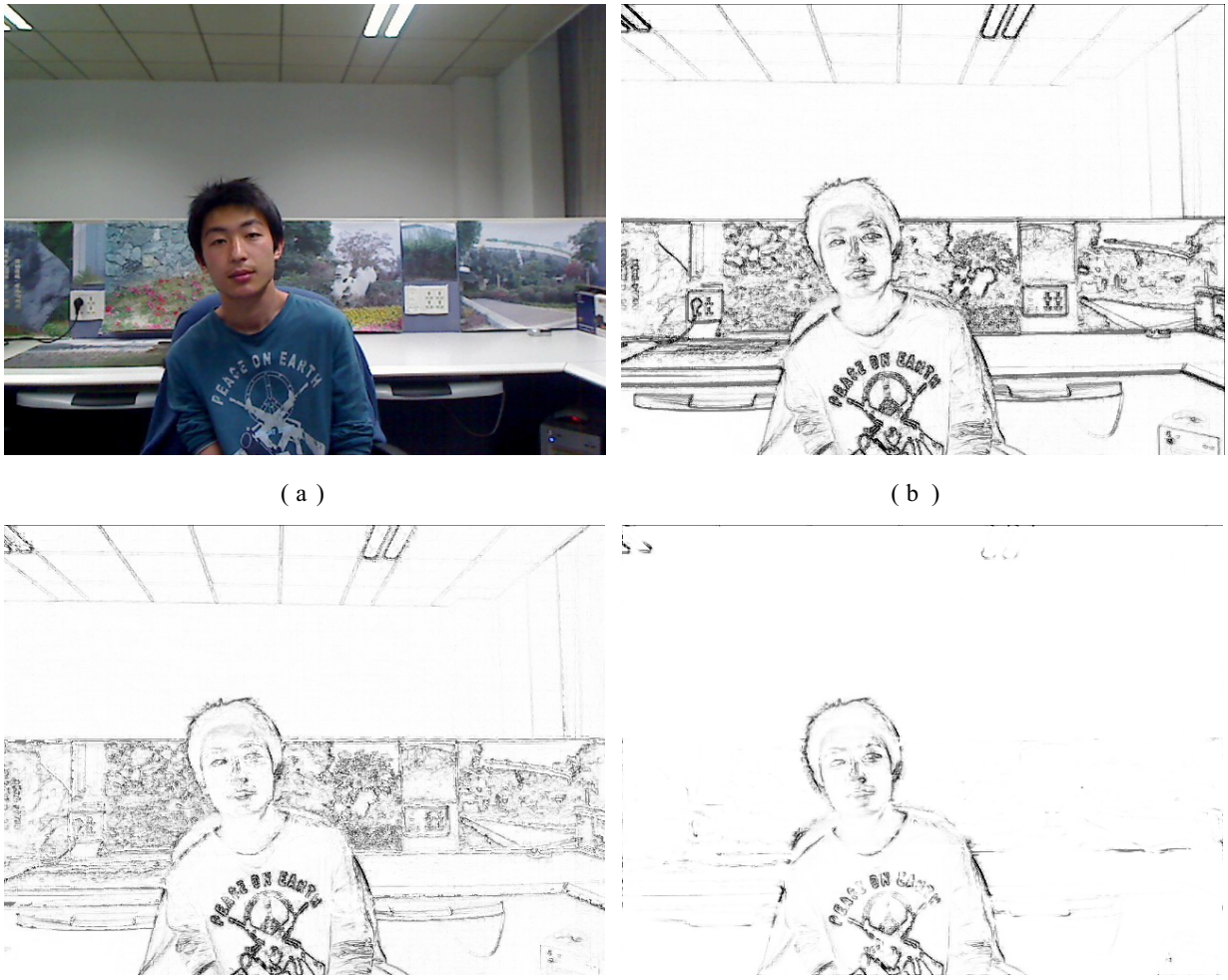
图 4.5 背景和前景数据项。(a) 背景数据项，像素颜色越亮，属于背景的概率越高；(b) 前景数据项，像素颜色越亮，属于前景的概率越高。

如果背景消减和局部颜色模型的判断一致，说明数据项的概率判断是可信的，系统直接利用背景消减的概率；否则，数据项存在不确定性，不能判断像素的标注，只能让平滑项决定。这种数据项的设置方法非常保守，因为我们观察到数据项其实包含很多噪声而不精确，如果设置错误会给结果带来很大的影响，综合考虑两种数据项的投票结果可以很大程度上保证数据项的正确。在估计输入图像背景的时候，系统已经计算出背景全景图到当前图像的单应矩阵$H_{rt}$，使用同样的映射可以得到输入图像每一个像素在背景颜色模型上对应的局部颜色模型，如图4.5。可以看出，前景的人物形状已经基本呈现出来，而且背景的局部颜色模型比前景的高斯混合模型更加精确。

#### 4.1.3.2 空域平滑项

时域平滑项虽然能对分割结果的稳定提供重要的作用，但是计算代价太高，不适合实时系统，因此本文主要考虑的是空域平滑项。Sun等指出[133]背景图像上的强烈颜色反差（强边）对分割结果影响很大，并提出一种背景颜色反差压制方法。然而，我们发现直接利用这种方法不能有效地消除背景上的边，如图4.6(c)。背景边不能消除的原因是估计的背景和真实背景之间存在配准误差，而不对齐的边无法直接消除。

考虑了输入图像和背景图像之间的配准误差，本文提出一种新的背景颜色反差压制方法，主要的想法是：把简单的颜色反差算法替换成更复杂的平滑的梯度算子，如高斯一阶



( a ) ( b )

( c ) ( d )

图 4.6 背景颜色反差压制比较。(a) 原始输入图像；(b) 输入图像的颜色反差；(c) Sun等[133]的背景颜色反差压制结果；(d) 本文的背景颜色反差压制结果。

梯度：

$$\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}) = (I * \frac{\partial G}{\partial x}, I * \frac{\partial G}{\partial y}),$$

其中$G$是一个高斯函数，定义为：

$$G(x,y) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{x^2+y^2}{2\sigma^2}),$$

其中，$\sigma$是高斯标准差，$\sigma$越大，可以处理的配准误差也越大，实验中设为3.0，梯度计算结果如图4.8(b)所示。

至此，系统可以计算前景和背景梯度图$\nabla I$和$\nabla I^B$，并用$\nabla I^B$压制$\nabla I$上属于背景的梯


(a) (b) (c)

图 4.7 不同配准误差下的背景颜色反差压制结果。(a) 原始颜色反差图像。(b)(c) 配准误差分别为3像素、5像素的背景颜色反差压制结果。

度。本文对$\nabla I^B$每一个像素施加不同的缩放因子，使其尽可能接近$\nabla I$的梯度，然后计算缩放之后的梯度差。缩放因子必须是正数，而且不能超过设定的阈值。于是，压制之后的梯度强度可以计算：

$$\overline{g} = (1 - \exp(-\frac{(1 - \min\{\gamma, 1/\gamma\})^2}{\sigma_\gamma^2})) \cdot \min_\gamma |\nabla I - \gamma \cdot \nabla I^B|, \tag{4.14}$$

$$\gamma = \min\{\max\{\frac{\nabla I \cdot \nabla I^B}{||\nabla I^B||^2}, 0\}, \tau\},$$

其中，$\gamma$就是梯度的缩放因子，$\tau$是$\gamma$的上限. 理想情况下，如果输入图像上的像素属于背景，$\gamma$应该等于1 。然而考虑到光照变化和颜色混合，$\gamma$会在$[0, \tau]$之间取一个压制效果最佳的值。实验中，$\tau$设为10。

受到图像噪声的影响，梯度的方向会发生一定的改变，$\min_\gamma |\nabla I - \gamma \cdot \nabla I^B|$对背景梯度的压制并不彻底。因此，本文在公式4.14中添加了一个衰减因子，在$\gamma$接近1的时候，这个衰减因子会进一步压制梯度。$\sigma_\gamma$的作用是调节$\gamma$的衰减影响，在实验中设为0.5。

上述的背景梯度压制方法对于光照变化和配准误差都有较大的容忍性，因为均一的全局光照变化不会影响梯度的方向。如果像素确实属于背景，系统一般都能找到最优的$\gamma$使$|\nabla I - \gamma \nabla I^B|$近于0。再者，高斯一阶梯度算子使得系统对背景配准误差也不敏感。图4.8(c)展示了一个对$\nabla I$进行梯度压制的结果。

在$\overline{g}$中，背景的梯度已经被压制到很弱的强度，空域平滑项可定义为：

$$E_s(\alpha_i, \alpha_j) = |\alpha_i - \alpha_j| \cdot \exp(-\beta d_{ij}), \tag{4.15}$$

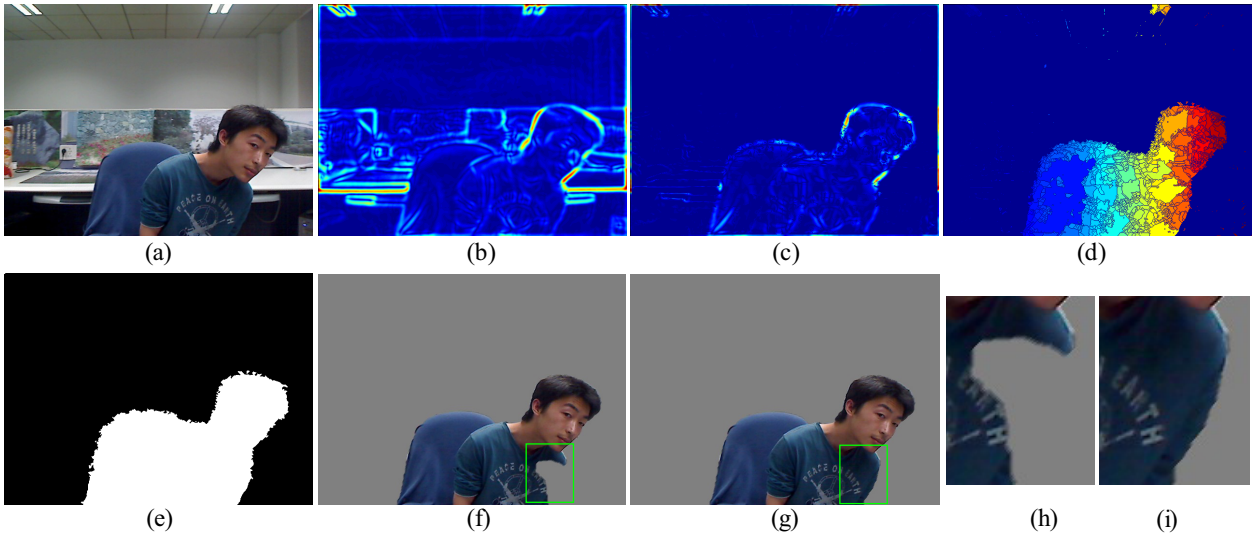


图 4.8 背景梯度压制和前景漏洞补全。(a) 原始输入图像；(b) 图像梯度$\nabla I$的梯度强度；(c)梯度压制结果$\bar{g}$；(d) $\bar{g}$的Watershed分割结果；(e) 候选的前景分割区域；(f) 没有进行漏洞补全的前景分割结果；(g) 漏洞补全之后的前景分割结果；(h) 图(f)的放大视图；(i) 图(g)的放大视图。

其中，$i,j$是输入图像上的相邻像素，$\beta$是一个根据图像内容自适应的鲁棒参数，$\beta = (2 < ||I_i - I_j||^2 >)^{-1}$（与Sun等的方法一致[133]）。$d_{ij}$是衰减因子：

$$d_{ij} = \|I_i - I_j\|^2 \cdot \frac{1}{1 + \frac{K^2}{\max\{\|\bar{g}_i\|^2, \|\bar{g}_j\|^2\}}}, \tag{4.16}$$

其中，$K$是一个常量，控制衰减强度。$\bar{g}_i$表示像素$i$被压制之后的梯度，见图4.6(d).

为了进一步证明本文的背景颜色反差压制算法在配准误差情形下的有效性，我们将估计的背景图像平移一定像素，人为地创造配准误差，测试算法在不同误差下的表现。如图4.7，即使有5个像素的对其误差，本文的算法也可以有效的压制背景颜色反差，这样的容忍度对纯旋转摄像机来说已经足够。

#### 4.1.3.3 前景漏洞补全

公式4.6中能量函数$E(\alpha)$的优化一般采用图割算法[①]，求解的$\alpha$就是双层分割结果。虽然现在的分割模型通常可以给出比较好的结果，但是在一些前景物体和背景颜色比较接近的输入图像中，还是会产生一些漏洞，如图4.8(f)。增加平滑项的权值可以将一些小的漏

[①]http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/software.html

洞补全，但是无法处理大漏洞，而且过强的平滑项会引入一些琐碎的结构，结果反而更差。为了有效地解决这些分割错误，本文引入一种基于Watershed的漏洞补全方法。

依据已有的梯度图$\bar{g}$，系统首先用Watershed方法[146]对其进行分割。Watershed方法的边缘阈值设为0.6，避免过分割的出现，分割结果见图4.8(d)。因为背景的梯度已经被压制过，大部分的背景像素会被分成一个大块；相反，前景的像素则被分成许多个小区域。因此，系统只需要考虑那些小的分割区域，它们更有可能是前景。对于这些分割区域，系统根据第4.1节的数据项，计算其中前景像素的比率：

$$\varepsilon = \frac{1}{|S_k|}\sum_{i \in S_k} h(I_i), \tag{4.17}$$

$$h(I_i) = \begin{cases} 1 & E_d(I_i|\alpha_i = 1) < E_d(I_i|\alpha_i = 0), \\ 0 & E_d(I_i|\alpha_i = 1) \geq E_d(I_i|\alpha_i = 0). \end{cases}$$

对于每一个分割区域$S_k$，如果$|S_k|/M < 0.03$（$M$图像的总像素），$\varepsilon > 0.2$，$S_k$就很可能是前景区域。图4.8(e)展示了这些候选的前景区域。当然可以直接将这些区域全部标注成前景物体，但是我们知道一旦背景消除和背景局部颜色误差的判断一致，数据项是十分可信的，因此系统仅仅将那些不可确定的像素标注为前景，即$E_d(I_i|\alpha_i = 1) = E_d(I_i|\alpha_i = 0) = 0.5$。最后，还是求解公式4.6的$E(\alpha)$，图4.8(g)的结果表明这个方法可以有效地填补前景物体中的漏洞。

### 4.1.4 系统实现和实验结果

本节的系统运行在桌面系统上，CPU是Intel(R) Core(TM)2 Quad CPU Q9550 @ 2.83GHz，显卡是GeForce GTX 275。系统的输入设备为罗技高清摄像头C905，输入视频分辨率是$640 \times 480$，捕获帧率在$20 \sim 30$fps之间。虚实融合采用OpenGL API，输出设备为普通液晶显示器。

#### 4.1.4.1 多分辨率实现

输入图像的分辨率是$640 \times 480$，直接进行双层分割的计算量太大，不能达到实时性能，系统采用三层多分辨率实现进行加速。各层的分辨率分别是：$160 \times 120$、$320 \times 240$、$640 \times 480$。SIFT特征点在$640 \times 480$分辨率上提取，输入图像的梯度也在原分辨率上计算。

第二、三层的双层分割都是在前一层的分割结果边界的邻域内完成，即沿着前景背景分割边界的20像素宽窄带，大大减少了计算量。

在计算公式4.7的背景消减时，前景像素不必要与局部窗口$W(i)$内的所有像素进行比较，只要找到一个像素$j$，满足$||I_i - I_j^B|| < 0.8\delta$，就可以结束搜索，再次减少了计算量。

系统中可调节的参数很多，但大多数是固定的。几个特殊的参数设置如：$\lambda = 0.5$, $K = 5, \tau = 10, \sigma_\gamma = 0.5$。对于室外场景，$\delta = 4 \sim 10$；对于室内场景，由于前景物体运动的影响，背景光照变化较大，相应的$\delta$也比较大，实验中设为22。

#### 4.1.4.2 实验结果

表4.1展示了系统各个模块大致的运行时间，双层分割还是比较耗时的，如果在单线程机器上运行，帧率只有4fps左右。采用与第3.3节类似的并行框架，系统在四核机器上运行，帧率可以达到12fps，基本满足实时要求。系统得到双层分割的结果之后，在前景的边缘上进行3个像素的羽化操作，使合成结果更加平滑。

| 模块 | 运行时间（毫秒） |
|---|---|
| SiftGPU提取SIFT特征点和匹配 | $\approx 50$ |
| 求解摄像机参数 | $\approx 5$ |
| 估计背景图像 | $\approx 20$ |
| 双层分割 | $\approx 100$ |
| 虚实合成 | $\approx 5$ |

表 4.1 实时双层分割运行时间。

图4.9展示了一个室内实例，分割出来的前景物体轮廓清晰精确，说明算法的鲁棒性，见图4.9(b)。图4.9(c)在背景增加了两个虚拟物体，由于已知前景背景之间的层次关系，系统可以精确地处理遮挡，增强真实感。将背景全景图替换成视频或者图像，再根据输入图像和背景全景图之间的映射关系，可以自然地替换输入视频的背景，如图4.9(d)。基于类似的操作，系统可以直接对背景全景图进行处理，比如模糊操作，然后替换背景，完成诸如景深控制等效果，如图4.9(e)。对背景的处理都是在预处理阶段就可以完成，实时阶段只需要简单的映射操作，十分高效。


图 4.9 Cubicle实例的增强现实结果。(a) 两帧输入图像；(b) 分割的前景物体；(c) 增强现实加入虚拟物体的遮挡效果；(d) 动态背景替换；(e) 模糊背景模拟景深效果；(f) 前景隐形。

以上操作主要针对背景，前景物体当然也可以进行相应的处理。本文实现了隐形效果，利用估计的背景信息$I^B$估计新的前景物体，然后和输入图像混合。设前景物体中心为$p_c$，对于前景物体上的某一点$p$，系统需要计算其在背景上对应像素$p'$，具体步骤如下：

1. 计算$p$到$p_c$的距离$d$；

2. 放大：$p' = p_c + (p - p_c) * (d/D + 0.4)$；

3. 扭曲：$p' = p' + sin((p - p_c)/P) * M$；

4. 扰动：$p' = p' + rand()/S$。

其中，$D$是$p$到$p_c$的最大距离，$P$和$M$分别是正弦函数的周期和幅度，$S$是扰动强度，$rand()$是随机函数。图4.9(f)展示最终的前景隐形合成结果。

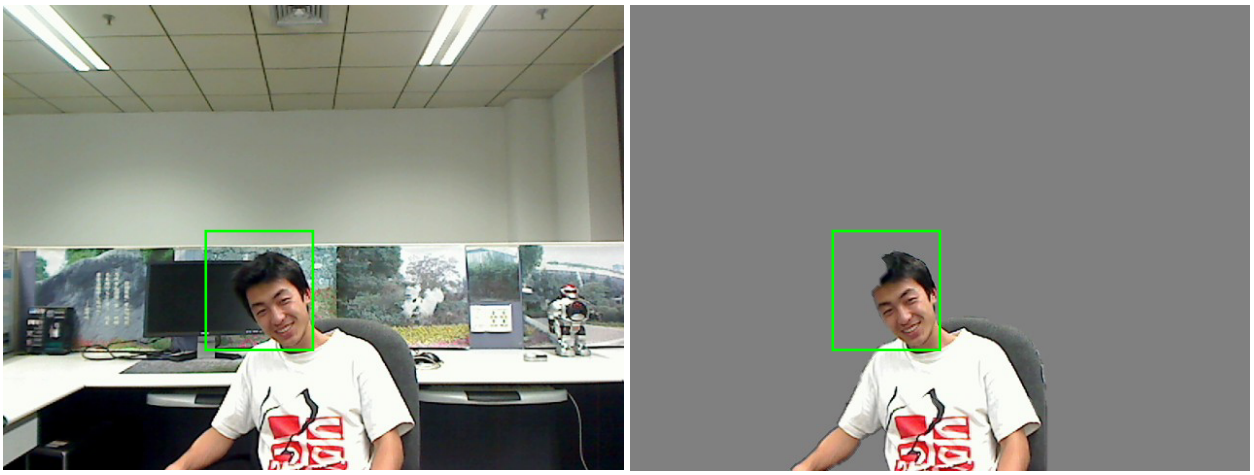
图4.10展示了一个室外Garden实例，背景全景图分别用发光和晶格化处理，然后映射到输入图像上。发光和晶格化都是非常耗时的图像处理操作，但在系统中只要预处理一遍，没有给实时阶段引入任何额外的操作。图4.11展示了另一个室外场景实例。可以看出，这些实例的背景都很复杂，尤其是室外场景的背景，而且摄像机的转动幅度很大，但是本文的系统表现都十分稳定。

图 4.10 Garden实例的增强现实结果。第一列是输入图像，第二列是分割结果，第三列上图发光效果，第三列下图是晶格化效果。

然而，当前的系统还存在一些内在的缺陷。一、背景上必须有丰富的特征，否则实时的摄像机参数恢复和背景估计就不可能完成；二、前景和背景颜色分布还是不能太接近，由于没有利用交互或时序信息[125]，此时的分割结果还是容易出错，如图4.12，头发的黑

图 4.11 Campus实例的增强现实结果。

色和显示器的黑色混杂在一起，算法无法区分二者；三、场景背景不能出现大的变动，比如物体的移动，光照的变化等，否则要重复预处理过程，比较繁琐。



(a) (b)

图 4.12 双层分割错误。(a) 输入图像；(b) 前景物体。如绿色方框包围的区域，由于头发的颜色和显示器的颜色都是黑色，头发的边缘信息又很弱，本文的算法无法区分二者的颜色，导致分割错误。

## 4.2 自由运动相机下的增强现实系统

上一节讨论了摄像机只有旋转运动情况下的增强现实系统，然而实际应用中更多摄像机可以自由运动。由于运动的复杂性，此时不能再以简单的层次关系表达场景的结构，我们不考虑虚实物体之间的遮挡关系。本节以虚拟装机为例，侧重于自由运动相机下的增强现实系统设计和虚拟内容提供。

由于摄像机可以自由运动，用户的运动范围也比较大，配套的硬件需要更高的敏捷性。虚实融合显示终端采用**头盔显示器** eMagin Z800 3DVisor（图4.13(a)）。Z800只配备双目视频显示器，本身不带有摄像输入设备，还需要附加一个移动摄像头，我们采用罗技高清摄像头C905（图4.13(b)）。理想情况下，Z800和C905通过USB数据线连接便携笔记本上，放置在一个背包内，用户可以背着书包自由走动，如图4.13(c)。但是从第3章可以看出，实时三维跟踪对计算机的性能要求比较高，流畅的系统运行需要四核机器和高端显卡，现有的便携笔记本性能通常不能满足要求。如果把Z800和C905连接到计算服务器上，用户的活动又会被数据连接线影响，因此本系统采用服务器/客户端网络框架。服务器完成特征点提取、匹配，和摄像机求解等计算量大的工作，而客户端只负责捕获视频输入和虚实融合。

（a）


（b）

（c）

图 4.13 移动摄像头和头盔显示器。(a) 头盔显示器 eMagin Z800 3DVisor；(b) 罗技高清摄像头C905；(c) 用户携带头盔显示器、摄像头、和笔记本。

### 4.2.1 系统设计

如上所述，本系统采用服务器/客户端框架，服务器为常见的配备四核处理器和高端显卡的台式机；客户端是普通的便携笔记本，因为客户端也要执行多线程任务，为了避免线程之间的资源竞争，保证系统运行的流畅，笔记本的运算CPU最好是双核以上。捕获摄像头C905和头盔显示器Z800连接在客户端上，负责输入和输出，服务器负责中间运算，服务器和客户端之间通过WI-FI无线网络连接，系统流程见图4.14。

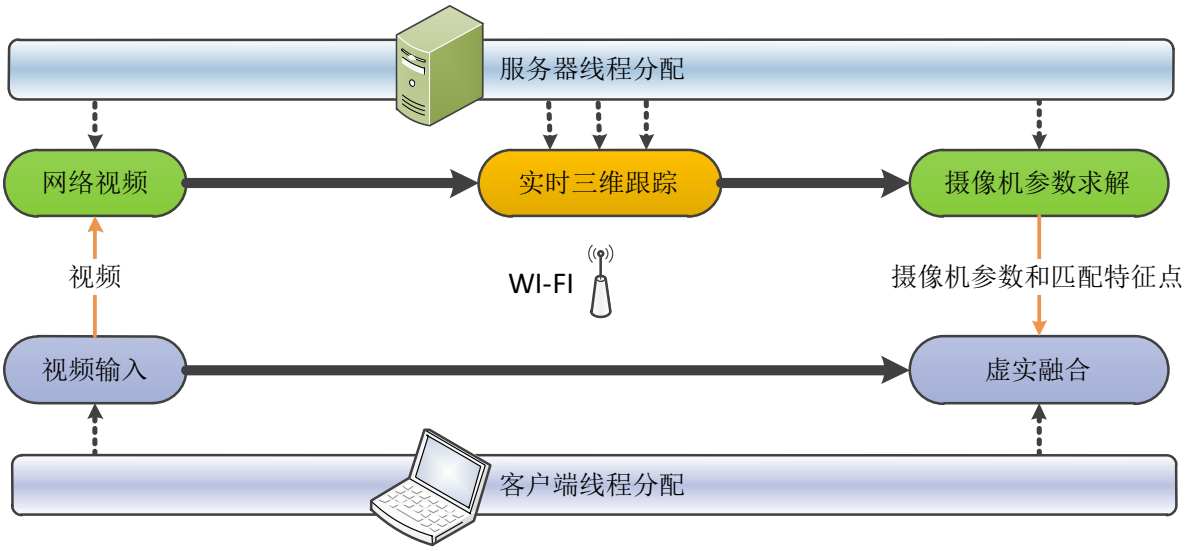



图 4.14 自由运动相机下的增强现实系统。系统基于实时三维跟踪框架，为了使摄像机的运动更加方便，采用移动终端显示虚实融合结果，并通过WI-FI无线网络连接计算服务器。

本系统采用的线程库是Boost Thread[②]。由于输入视频的数据非常大，如果直接传输图像，按照每秒钟20帧640 × 480图像计算，需要的网络带宽大约是17.6MB/s，WI-FI无线网络不具备这样的传输能力。系统选择将图像分辨率降到320 × 240，并利用JPEG2000[③]算法进行图像压缩，最后的传输数据为800KB/s左右。网络传输的协议有TCP和UDP，TCP传输比较稳定，但是速度慢，UDP传输充分利用网络带宽，但是容易丢包，两者都不符合系

[②] http://www.boost.org

[③] http://www.jpeg.org/jpeg2000/

统要求。本系统利用开源网络协议UDT（UDP-based Data Transfer）④完成网络传输，服务器和客户端的具体运行方式见表4.2。

| 客户端的运行方式 | 服务器的运行方式 |
| --- | --- |
| 1. 摄像头C905捕获一帧场景图像；<br>2. 利用JPEG2000算法压缩图像；<br>3. 向服务器发送压缩图像；<br>4. 等待服务器送回摄像机参数；<br>5. 完成虚实融合绘制。 | 1. 从客户端接收压缩图像；<br>2. 利用JPEG2000算法解压图像；<br>3. 提取、匹配SIFT特征点，求解摄像机参数；<br>4. 向客户端发送摄像机参数和匹配特征点。 |

表 4.2 服务器和客户端的运行方式。

### 4.2.2 虚实融合

本节以计算机主板装配为例演示虚实融合技术，其中图形绘制采用OpenGL API。在预处理阶段，系统已经重建了场景的点云模型，我们可以添加虚拟物体，并以点云模型为依据将虚拟物体摆放到合适的位置，见图4.16。系统暂时支持以下虚拟信息：

- OBJ三维模型和动画：系统在实物上叠加虚拟物体的三维模型，展示用户需要装配的部件，并以三维动画的形式演示装配过程；
- 视频：系统预先录制装配操作的视频，以更符合现实情况的方式展示装配流程；
- 文字：利用文字信息，系统可以按照用户的提示，一步步提供装配的指示文字。

用户在系统中摆放好虚拟信息后，就可以运行实时程序，系统根据三维跟踪的结果，自动将虚拟信息和现实场景配准，图4.15展示了一部分结果。由于系统加入了网络传输模块，虚实融合的延时难免要加入传输的时间，现在的系统延时大致增加了70ms，在150ms左右。

④http://udt.sourceforge.net/index.html

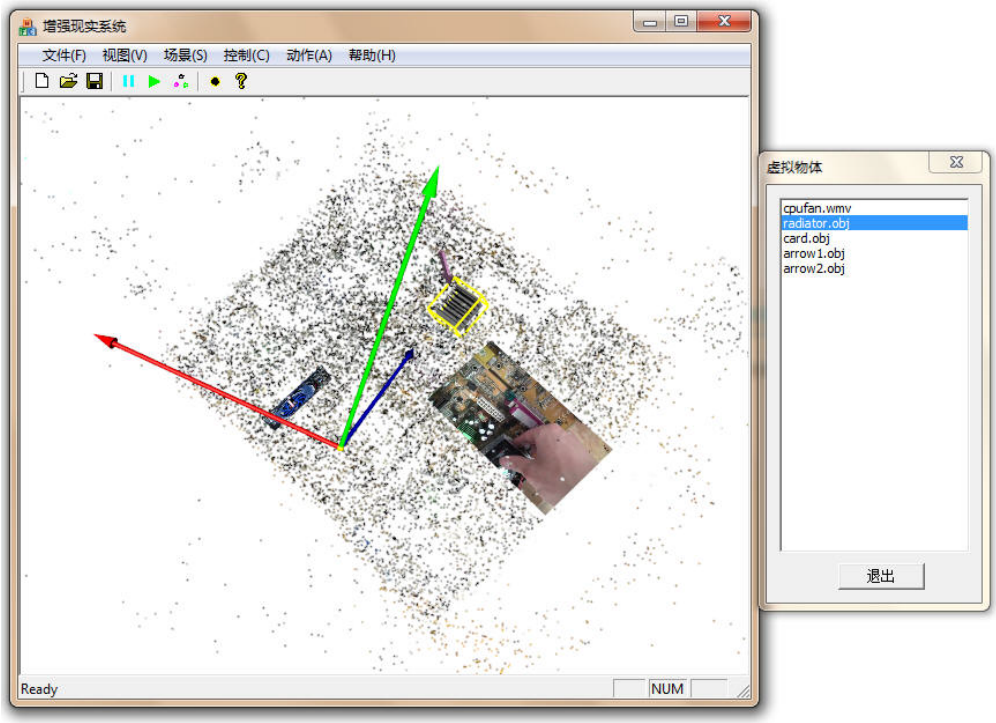



图 4.15 增强现实系统的界面。在预处理阶段，系统以场景的三维点云为基础，布置虚拟物体，如三维模型，视频等。左边是程序的交互主窗口，右边是场景中的虚拟物体列表。

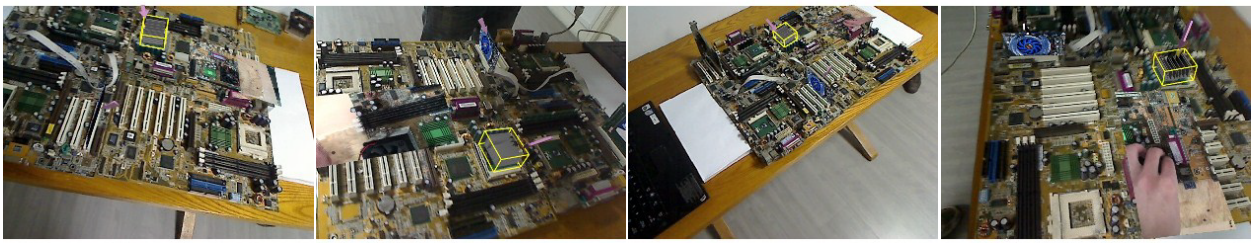


图 4.16 增强现实系统的运行结果。虽然用户的视角变化很大，但是三维跟踪都能成功求解摄像机方位。

## 4.3 小结

本章实现了两类基于实时三维跟踪的增强现实系统，分为纯旋转相机和自由运动相机两种情况。

在纯旋转相机情况下，本文提出一种基于三维跟踪框架的实时双层分割方法，可以处理简单的增强现实虚实遮挡，并实现了一系列新颖的增强现实效果。本文的方法将场景的背景表示成全景图，以便摄像机旋转运动时，能恢复输入图像的背景信息。实时三维跟踪

算法提供输入图像的摄像机旋转参数，以及与背景全景图之间的映射关系。复杂场景中的双层分割是很难稳定求解的问题，而且摄像机的运动，虽然只限于旋转运动，也引入配准误差等额外的负担。本文结合背景的局部颜色模型和鲁棒的背景颜色反差压制方法，有效地改进了基于已知背景的实时双层分割方法。在此基础上，系统还实现了一系列特殊的增强现实效果。

本文还以三维跟踪技术为基础，实现了自由运动相机下的增强现实系统。系统利用网络传输，将三维跟踪和虚实融合分别运行在服务器和客户端，增加用户活动的灵活性。系统还加入对三维模型，视频、文字等虚拟信息的支持，演示了增强现实的虚拟装机应用。

# 5 总结和展望

计算机视觉和计算机图形学是计算机科学研究的两个重要领域，体现了人们对现实世界和虚拟世界的理解，虚实融合是两者的交叉，具有很广泛的应用。增强现实作为虚实融合的主要分支，已经在影视制作、广告设计、交互游戏中商用，但这些都还是比较耗时的离线过程，实时增强现实也已经产生一些原型系统，而且还在持续的研究开发当中。

实时增强现实首要解决的是三维跟踪问题，一般情况下场景的三维在跟踪之前是已知的，实时三维跟踪的任务是实时恢复输入摄像机的方位，使虚拟物体与场景三维配准。对于实时三维跟踪来说，最重要的是系统的实时性能，各个系统模块在不影响稳定性的前提下，要尽可能的缩减运行时间。总而言之，实时性和稳定性是实时三维跟踪的主要研究内容。

## 5.1 本文总结

本文考虑实时性和稳定性要求，提出了实时增强现实系统的基本框架，并在此基础上针对不同的应用需求，探讨了多种实现方案。

- 人工设计的基准标志在增强现实中具有重要意义，具有求解稳定，计算量小，运行平台要求低等特点，主要在缺少纹理颜色特征的环境中使用。本文继承了基准标志的已有工作，在应用上做出两个方面改进。一是将汉字内容与标志检测结合起来，利用标志边缘信息更稳定地检测标志，利用汉字结构更精确地识别标志，实现了基于汉字标志的学习系统。二是扩展了基准标志的内容，把现实中现有的平面图像引入标志系统，弥补经典基准标志的颜色单调缺陷，改善了用户的视觉体验。

- 在特征丰富的场景中，我们可以直接利用场景自有的特征信息进行三维跟踪，实现更自然的增强现实。利用SfM技术可以自动精确地恢复场景的三维点云结构，再以三维点云为基础进行三维跟踪。现有的方法直接将输入图像上的特征和三维点云进行全局匹配，可以处理较小的场景，但应用到大规模场景，会出现诸如匹配点不足，求解不稳定等问题。本文通过从输入图像选择关键帧，将三维场景分割成每个关键

帧包含的子场景，特征点匹配从全局转移到局部，实时三维跟踪的性能和稳定性得到很大的提升。系统包括优化目标函数自动选择关键帧的方法，和快速识别与输入图像相似的候选关键帧并与其匹配的方法。目标函数充分考虑关键帧的性质，综合数据完备性和冗余性因素，并且可通过贪婪法快速优化。识别与输入图像相似的候选关键帧则采用图像识别的框架，本文改进关键帧投票方法，很大地压缩了识别时间。室内室外环境的实验结果证明系统的实用性和稳定性。

- 虚实遮挡关系也是增强现实的重要问题，如果解决得当，可以大大增加虚实融合的真实感。然而，要完全解决虚实遮挡是很困难的工作，需要现实场景的完整三维模型，很多时候没办法得到。本文工作希望先在简单的情形下解决虚实遮挡，为更通用的遮挡处理方法提供参考。系统的基本设定是纯旋转运动的摄像机、静止背景和运动前景，并假设虚拟信息在背景和前景之间、或在前景之前，因此只需要把背景和前景物体分离开，就可以完成虚实遮挡。这是一个实时双层分割的问题，本文采用了现有的图割框架，根据实际情况重新定义数据项和平滑项，并通过一系列增强现实结果证明系统的有效性。

## 5.2 未来工作

本文的工作虽然取得一定的成果，但还是存在一些问题，比如依赖场景的预处理结果，对计算平台的性能要求比较高，移动性弱等，未来工作可从以下方向着手。

- **并行三维重建和跟踪**

  SLAM和Online-SfM是并行三维重建和跟踪领域的研究热点，包括与其直接相关的重定位、回路检测、子场景组织等问题。并行三维重建和跟踪不需要对场景的预处理过程，每次实时重建的三维都不一致，不适合增强现实应用，但与本文的工作正好形成互补。我们考虑将两种技术路线结合起来，一方面简化对现实场景的预处理建模，另一方面在实时三维跟踪阶段利用并行三维重建和跟踪技术补充预处理阶段未能恢复的三维信息。这种组合可以保证系统始终面对一致的场景三维，又可以根据场景的变化自动做出调整，不断延伸场景的三维信息，必能推动实时三维跟踪技术的应用发展。

- **手机平台**

  近几年来，随着手机的迅猛发展和移动应用程序商店的出现，手机上的应用程序也形成爆炸式的增长，已经出现一些增强现实系统[147,148]。手机的优点是普遍性和移动性，但是计算能力受到其体积的限制，无法实时地运行复杂系统。然而，计算能力的不足可以通过硬件来弥补，最新的IPhone 4代等手机搭载了GPS，重力感应器，陀螺仪等①，如果充分利用这些硬件，复杂的增强现实在手机上也可以现实[149]。借助手机的硬件和移动性能，城市规模的实时三维跟踪也渐渐成为可能。

- **人机交互**

  在人机交互中，增强现实的作用主要是在现实场景中，利用投影仪、触摸屏等设备提供交互界面，并保证界面的一致性、稳定性、友好性。最近的SixthSense[5]和Microsot Kinect②深入挖掘增强现实和动作识别结合之后的应用可能性，将人机交互推到一个全新的高度。本文的工作过去集中于增强现实，将来必须与人机交互结合，发挥其应有的价值!

①http://www.apple.com/iphone/

②http://www.xbox.com/kinect/

# 参考文献

[1] AZUMA R, BAILLOT Y, BEHRINGER R, et al. Recent advances in augmented reality[J]. IEEE Computer Graphics and Applications, 2001, 21(6):34–47.

[2] 朱淼良, 姚远, 蒋云良. 增强现实综述[J]. 中国图象图形学报A辑, 2004, 9(7):767–774.

[3] FISCHER J, EICHLER M, BARTZ D, et al. Model-based Hybrid Tracking for Medical Augmented Reality[C]. Eurographics Symposium on Virtual Environments (EGVE). 2006: 71–80.

[4] 黄天智, 刘越, 王涌天, et al. 用于圆明园数字重建的定点式户外增强现实系统[C]. 第四届全国虚拟现实与可视化学术会议. 大连: 2004.

[5] MISTRY P, MAES P, CHANG L. WUW - wear Ur world: a wearable gestural interface[C]. CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems. New York, NY, USA: ACM, 2009: 4111–4116.

[6] 明德烈, 柳健, 田金文. 增强现实中的虚实注册技术研究[J]. 中国图象图形学报, 2003, 8(05):557–561.

[7] CANNY J. A computational approach to edge detection[J]. IEEE Trans. Pattern Anal. Mach. Intell., 1986, 8(6):679–698.

[8] LINDEBERG T. Edge detection and ridge detection with automatic scale selection[J]. Int. J. Comput. Vision, 1998, 30(2):117–156.

[9] JUNEJA M, SANDHU P S. Performance evaluation of edge detection techniques for images in spatial domain[J]. International Journal of Computer Theory and Engineering, 2009, 1(5):1793–8201.

[10] FISCHLER M A, BOLLES R C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography.[J]. Commun. ACM, 1981, 24(6):381–395.

[11] GIL A, MOZOS O M, BALLESTA M, et al. A comparative evaluation of interest point detectors and local descriptors for visual SLAM[J]. 2009, 1–16.

[12] TUYTELAARS T, MIKOLAJCZYK K. Local invariant feature detectors: a survey[J]. Found. Trends. Comput. Graph. Vis., 2008, 3(3):177–280.

[13] HEL-OR Y, HEL-OR H. Real-time pattern matching using projection kernels[J]. IEEE Trans. Pattern Anal. Mach. Intell., 2005, 27(9):1430–1445.

[14] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91–110.

[15] MOREELS P, PERONA P. Evaluation of features detectors and descriptors based on 3D objects[J]. Int. J. Comput. Vision, 2007, 73(3):263–284.

[16] VIKSTEN F, FORSSÉN P E, JOHANSSON B, et al. Comparison of local image descriptors for full 6 degree-of-freedom pose estimation[C]. ICRA'09: Proceedings of the 2009 IEEE international conference on Robotics and Automation. Piscataway, NJ, USA: IEEE Press, 2009: 1139–1146.

[17] HARRIS C, STEPHENS M. A combined corner and edge detection[C]. Proceedings of The Fourth Alvey Vision Conference. 1988: 147–151.

[18] ROSTEN E, PORTER R, DRUMMOND T. FASTER and better: A machine learning approach to corner detection[J]. IEEE Trans. Pattern Analysis and Machine Intelligence, 2010, 32:105–119. http://lanl.arXiv.org/pdf/0810.2434.

[19] MATAS J, CHUM O, URBAN M, et al. Robust wide-baseline stereo from maximally stable extremal regions[C]. vol 22. 2004: 761 – 767.

[20] BAY H, TUYTELAARS T, GOOL L J V. Surf: Speeded up robust features[C]. ECCV (1). 2006: 404–417.

[21] LINDEBERG T. Feature detection with automatic scale selection[J]. International Journal of Computer Vision, 1998, 30(2):79–116.

[22] SHI J, TOMASI C. Good features to track[C]. 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94). 1994: 593 – 600.

[23] LUCAS B D, KANADE T. An iterative image registration technique with an application to stereo vision[C]. IJCAI. 1981: 674–679.

[24] LINDEBERG T. Scale-space[J]. Encyclopedia of Computer Science and Engineering, 2009, 4:2495–2504.

[25] Indexing based on scale invariant interest points[C]. vol 1 2001.

[26] DIAS P, KASSIM A, SRINIVASAN V. A neural network based corner detection method[C]. Neural Networks, 1995. Proceedings., IEEE International Conference on. vol 4. 1995: 2116 –2120 vol.4.

[27] KUMAR R, CHEN W C, ROCKETT P. Bayesian labelling of image corner features using a grey-level corner model with a bootstrapped modular neural network[C]. Artificial Neural Networks, Fifth International Conference on (Conf. Publ. No. 440). 1997: 82 –87.

[28] KIENZLE W, WICHMANN F, SCHOLKOPF B, et al. Learning an interest operator from human eye movements[C]. Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on. 2006: 24 – 24.

[29] TRUJILLO L, OLAGUE G. Synthesis of interest point detectors through genetic programming[C]. GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation. New York, NY, USA: ACM, 2006: 887–894.

[30] MIKOLAJCZYK K, TUYTELAARS T, SCHMID C, et al. A comparison of affine region detectors[J]. Int. J. Comput. Vision, 2005, 65(1-2):43–72.

[31] PCA-SIFT: a more distinctive representation for local image descriptors[C]. vol 2 2004. http://dx.doi.org/10.1109/CVPR.2004.1315206.

[32] MIKOLAJCZYK K, SCHMID C. A performance evaluation of local descriptors[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2005, 27(10):1615 –1630.

[33] HEIKKILA M, PIETIKAINEN M, SCHMID C. Description of interest regions with local binary patterns[J]. Pattern Recognition, 2009, 42(3):425–436.

[34] OJALA T, PIETIKÄINEN M, MÄENPÄÄ T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns[J]. IEEE Trans. Pattern Anal. Mach. Intell., 2002, 24(7):971–987.

[35] TOEWS M, WELLS III W M. SIFT-Rank: ordinal descriptors for invariant feature correspondence[C]. International Conference on Computer Vision and Pattern Recognition (CVPR). 2009: 172–177.

[36] WINDER S A J, BROWN M. Learning local image descriptors[C]. In CVPR. 2007: 1–8.

[37] CALONDER M, LEPETIT V, FUA P, et al. Compact signatures for high-speed interest point description and matching[C]. Computer Vision, 2009 IEEE 12th International Conference on. 2009: 357 –364.

[38] PELE O, WERMAN M. A linear time histogram metric for improved sift matching[C]. ECCV '08: Proceedings of the 10th European Conference on Computer Vision. Berlin, Heidelberg: Springer-Verlag, 2008: 495–508.

[39] PELE O, WERMAN M. Fast and robust earth mover's distances[C]. ICCV. 2009.

[40] BEIS J S, LOWE D G. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces[C]. CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition. Washington, DC, USA: IEEE Computer Society, 1997: 1000.

[41] LIAW Y C, LEOU M L, WU C M. Fast exact k nearest neighbors search using an orthogonal search tree[J]. Pattern Recognition, 2010, 43(6):2351 – 2358.

[42] LIU T, MOORE A W, GRAY E, et al. An investigation of practical approximate nearest neighbor algorithms[C]. MIT Press 2004: 825–832.

[43] FERHATOSMANOGLU H, TUNCEL E, AGRAWAL D, et al. High dimensional nearest neighbor searching[J]. Information Systems, 2006, 31(6):512 – 540.

[44] GIONIS A, INDYK P, MOTWANI R. Similarity search in high dimensions via hashing[C]. VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999: 518–529.

[45] ARYA S, MOUNT D M, NETANYAHU N S, et al. An optimal algorithm for approximate nearest neighbor searching fixed dimensions[J]. Journal of the ACM (JACM), 1998, 45(6):891–923.

[46] ÖZUYSAL M, CALONDER M, LEPETIT V, et al. Fast keypoint recognition using random ferns[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(3):448–461. http://dx.doi.org/10.1109/TPAMI.2009.23.

[47] ÖZUYSAL M, LEPETIT V, FLEURET F, et al. Feature harvesting for tracking-by-detection[C]. Proceedings of European Conference on Computer Vision. 2006: 592–605.

[48] LEPETIT V, FUA P. Keypoint recognition using randomized trees[J]. IEEE Trans. Pattern Anal. Mach. Intell., 2006, 28(9):1465–1479.

[49] HAND A, CHL M, STRASDAT H, et al. Scalable active matching[M]. unpublished:. 2010.

[50] LEE T, HÖLLERER T. Multithreaded hybrid feature tracking for markerless augmented reality[J]. IEEE Transactions on Visualization and Computer Graphics, 2009, 15(3):355–368.

[51] TEICHRIEB V, DO MONTE LIMA J, APOLINÁRIO E, et al. A survey of online monocular markerless augmented reality[J]. International Journal of Modeling and Simulation for the Petroleum Industry, 2007, 1(1):1–7.

[52] LEPETIT V, FUA P. Monocular model-based 3D tracking of rigid objects[J]. Found. Trends. Comput. Graph. Vis., 2005, 1(1):1–89.

[53] WU Y, HU Z. PnP problem revisited[J]. J. Math. Imaging Vis., 2006, 24(1):131–141.

[54] LIU Y, HUANG T, FAUGERAS O. Determination of camera location from 2-D to 3-D line and point correspondences[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1990, 12(1):28 –37.

[55] HARALICK R, LEE D, OTTENBURG K, et al. Analysis and solutions of the three point perspective pose estimation problem[C]. Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on. 1991: 592 –598.

[56] TRIGGS B. Camera pose and calibration from 4 or 5 known 3D points[C]. Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on. vol 1. 1999: 278 – 284.

[57] GAO X S, HOU X R, TANG J, et al. Complete solution classification for the perspective-three-point problem[J]. IEEE Trans. Pattern Anal. Mach. Intell., 2003, 25(8):930–943.

[58] 张彩霞, 胡占义. P3P问题的多解现象的概率研究[J]. 软件学报, 2007, 18(9).

[59] TANG J, CHEN W S, WANG J. A novel linear algorithm for P5P problem[J]. Applied Mathematics and Computation, 2008, 205(2):628 – 634.

[60] SCHWEIGHOFER G, PINZ A. Robust pose estimation from a planar target[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2006, 28(12):2024 –2030.

[61] 吴福朝, 胡占义. 关于P5P问题的研究[J]. 软件学报, 2001, 12(5).

[62] QUAN L, LAN Z. Linear N-point camera pose determination[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999, 21:774–780.

[63] FIORE P. Efficient linear solution of exterior orientation[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2001, 23(2):140 –148.

[64] LEPETIT V, MORENO-NOGUER F, FUA P. EPnP: An accurate O(n) solution to the PnP problem[J]. Int. J. Comput. Vision, 2009, 81(2):155–166.

[65] HARTLEY R I, ZISSERMAN A. Multiple View Geometry in Computer Vision[M]. Secondth ed. Cambridge University Press, ISBN: 0521540518, 2004.

[66] LU C P, HAGER G, MJOLSNESS E. Fast and globally convergent pose estimation from video images[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2000, 22(6):610 –622.

[67] GREST D, PETERSEN T, KRÜGER V. A comparison of iterative 2d-3d pose estimation methods for real-time applications[C]. SCIA '09: Proceedings of the 16th Scandinavian Conference on Image Analysis. Berlin, Heidelberg: Springer-Verlag, 2009: 706–715.

[68] ZHANG Z. Parameter estimation techniques: A tutorial with application to conic fitting[J]. Image and Vision Computing Journal, 1997, 15(1):59–76.

[69] 刘国翌, 李华. 平板图案的跟踪及其在增强现实中的应用[J]. 计算机应用, 2005, 25(7).

[70] FIALA M. Designing highly reliable fiducial markers[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2010, 32(7):1317 –1324.

[71] CHO Y, LEE J, NEUMANN U. A multi-ring color fiducial system and an intensity-invariant detection method for scalable fiducial-tracking augmented reality[C]. International Workshop on Augmented Reality. 1998: 147–165.

[72] NAIMARK L, FOXLIN E. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker[C]. ISMAR '02: Proceedings of the 1st International Symposium on Mixed and Augmented Reality. Washington, DC, USA: IEEE Computer Society, 2002: 27.

[73] ABABSA F E, MALLEM M. Robust camera pose estimation using 2d fiducials tracking for real-time augmented reality systems[C]. VRCAI '04: Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry. New York, NY, USA: ACM, 2004: 431–435.

[74] KATO H, BILLINGHURST M. Marker tracking and HMD calibration for a video-based augmented reality conferencing system[J]. Augmented Reality, International Workshop on, 1999, 0:85–94.

[75] ZHANG X, FRONZ S, NAVAB N. Visual marker detection and decoding in AR systems: A comparative study[C]. ISMAR '02: Proceedings of the 1st International Symposium on Mixed and Augmented Reality. Washington, DC, USA: IEEE Computer Society, 2002: 97.

[76] FIALA M. ARTag, a fiducial marker system using digital techniques[C]. CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol 2. Washington, DC, USA: IEEE Computer Society, 2005: 590–596.

[77] ANTLE A N, MOTAMEDI N, TANENBAUM K, et al. The EventTable technique: distributed fiducial markers[C]. TEI '09: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction. New York, NY, USA: ACM, 2009: 307–313.

[78] COSTANZA E, HUANG J. Designable visual markers[C]. CHI '09: Proceedings of the 27th international conference on Human factors in computing systems. New York, NY, USA: ACM, 2009: 1879–1888.

[79] TENMOKU R, KAIGAWA S, SHIBATA F, et al. Visually elegant and robust semi-fiducials for geometric registration in mixed reality[C]. The Sixth Int. Symposium on Mixed and Augmented Reality. 2007.

[80] TATENO K, KITAHARA I, OHTA Y. A nested marker for augmented reality[C]. IEEE Virtual Reality Conference, 2007. VR '07. IEEE 2007: 259–262.

[81] SAITO S, HIYAMA A, TANIKAWA T, et al. Indoor marker-based localization using coded seamless pattern for interior decoration[C]. IEEE Virtual Reality Conference, 2007. VR '07. IEEE 2007: 67–74.

[82] SNAVELY N, SEITZ S M, SZELISKI R. Modeling the world from Internet photo collections[J]. International Journal of Computer Vision, 2008, 80(2):189–210.

[83] AGARWAL S, SNAVELY N, SIMON I, et al. Building rome in a day[C]. International Conference on Computer Vision. Kyoto, Japan.: 2009.

[84] 陈靖, 王涌天, 李玉, et al. 基于自然特征点的实时增强现实注册算法[J]. 系统仿真学报, 2007, 19(22).

[85] CHIA K W, CHEOK A D, PRINCE S J D. Online 6 DOF augmented reality registration from natural features[C]. ISMAR '02: Proceedings of the 1st International Symposium on Mixed and Augmented Reality. Washington, DC, USA: IEEE Computer Society, 2002: 305.

[86] SKRYPNYK I, LOWE D G. Scene modelling, recognition and tracking with invariant image features[C]. ISMAR '04: Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality. Washington, DC, USA: IEEE Computer Society, 2004: 110–119.

[87] MOOSER J, YOU S, NEUMANN U, et al. Applying robust structure from motion to markerless augmented reality[C]. IEEE Workshop on Applications of Computer Vision. Snowbird, Utah: 2009.

[88] YUAN M, ONG S, NEE A. Registration using natural features for augmented reality systems[J]. IEEE Transactions on Visualization and Computer Graphics, 2006, 12:569–580.

[89] IRSCHARA A, ZACH C, FRAHM J M, et al. From structure-from-motion point clouds to fast location recognition[C]. CVPR. 2009.

[90] VACCHETTI L, LEPETIT V, FUA P. Stable real-time 3D tracking using online and offline information[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2004, 26(10):1385–1391.

[91] TRIGGS B, MCLAUCHLAN P F, HARTLEY R I, et al. Bundle adjustment - a modern synthesis.[C]. Workshop on Vision Algorithms. 1999: 298–372.

[92] ENGELS C, STEWÉNIUS H, NISTÉR D. Bundle adjustment rules[J]. Photogrammetric Computer Vision (PCV), 2006, 2.

[93] PARK Y, LEPETIT V, WOO W. Multiple 3d object tracking for augmented reality[J]. Mixed and Augmented Reality, 2008. ISMAR 2008. 7th IEEE/ACM International Symposium on, 2008, 117–120.

[94] HINTERSTOISSER S, BENHIMANE S, NAVAB N. N3m: Natural 3d markers for real-time object detection and pose estimation[C]. IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007. 2007: 1–7.

[95] DRUMMOND T, CIPOLLA R. Real-time visual tracking of complex structures[J]. IEEE Trans. Pattern Anal. Mach. Intell., 2002, 24(7):932–946.

[96] COMPORT A I, MARCHAND E, PRESSIGOUT M, et al. Real-time markerless tracking for augmented reality: The virtual visual servoing framework[J]. IEEE Transactions on Visualization and Computer Graphics, 2006, 12(4):615–628.

[97] KLEIN G, MURRAY D. Full-3d edge tracking with a particle filter[C]. Proc 17th British Machine Vision Conference. 2006.

[98] PUPILLI M, CALWAY A. Real-time camera tracking using known 3d models and a particle filter[C]. International Conference on Pattern Recognition. 2006.

[99] ROSTEN E, DRUMMOND T. Fusing points and lines for high performance tracking[C]. ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision. Washington, DC, USA: IEEE Computer Society, 2005: 1508–1515.

[100] PRESSIGOUT M, MARCHAND É. Real-time 3d model-based tracking: Combining edge and texture information[C]. IEEE ICRA' 06. 2006: 2726–2731.

[101] VACCHETTI L, LEPETIT V, FUA P. Combining edge and texture information for real-time accurate 3D camera tracking[C]. Third IEEE and ACM International Symposium on Mixed and Augmented Reality. Arlington, Virginia: 2004: 48–57.

[102] REITMAYR G, DRUMMOND T W. Going out: Robust tracking for outdoor augmented reality[C]. Proc. ISMAR 2006. IEEE and ACM. Santa Barbara, CA, USA: IEEE CS, 2006: 109–118.

[103] RIBEIRO M I. Kalman and extended kalman filters: Concept, derivation and properties[J]. Institute for Systems and Robotics, Lisboa Portugal, 2004.

[104] STRASDAT H, MONTIEL J M M, DAVISON A J. Real-time monocular SLAM: Why filter?[C]. Int. Conf. on Robotics and Automation, Anckorage, AK. 2010.

[105] DAVISON A J, REID I D, MOLTON N D, et al. MonoSLAM: Real-time single camera SLAM[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 26(6):1052–1067.

[106] CHEKHLOV D, PUPILLI M, MAYOL W, et al. Robust real-time visual SLAM using scale prediction and exemplar based feature description[C]. CVPR. 2007: 1–7.

[107] MONTEMERLO M. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association[D]. PhD thesis. Pittsburgh, PA: Robotics Institute, Carnegie Mellon University, 2003.

[108] PUPILLI M, CALWAY A. Real-time camera tracking using a particle filter[C]. Proc British Machine Vision Conference. 2005.

[109] EADE E, DRUMMOND T. Scalable monocular slam[C]. CVPR (1). 2006: 469–476.

[110] Monocular SLAM as a Graph of Coalesced Observations[C]. 2007.

[111] CHLI M, DAVISON A J. Automatically and efficiently inferring the hierarchical structure of visual maps[C]. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). 2009: 387–394.

[112] WILLIAMS B, KLEIN G, REID I. Real-time slam relocalisation[C]. Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. 2007: 1–8.

[113] CUMMINS M, NEWMAN P. FAB-MAP: Probabilistic localization and mapping in the space of appearance[J]. International Journal of Robotics Research, 2008, 27(6):647–665.

[114] SIVIC J, ZISSERMAN A. Video google: A text retrieval approach to object matching in videos[C]. ICCV. Washington, DC, USA: IEEE Computer Society, 2003: 1470.

[115] NISTER D, STEWENIUS H. Scalable recognition with a vocabulary tree[C]. CVPR. Washington, DC, USA: IEEE Computer Society, 2006: 2161–2168.

[116] SCHINDLER G, BROWN M, SZELISKI R. City-scale location recognition[C]. CVPR. 2007.

[117] ZHANG G, QIN X, HUA W, et al. Robust metric reconstruction from challenging video sequences[C]. CVPR. 2007: 1–8.

[118] KLEIN G, MURRAY D. Parallel tracking and mapping for small AR workspaces[C]. ISMAR 2007. 2007: 225–234.

[119] KLEIN G, MURRAY D. Improving the agility of keyframe-based slam[C]. European Conference on Computer Vision. vol 2. 2008: 802–815.

[120] MEI C, SIBLEY G, CUMMINS M, et al. A constant time efficient stereo slam system[C]. Bristish Machine Vision Conference. 2009.

[121] FISCHER J, BARTZ D, STR$\alpha\beta$SER W. Occlusion handling for medical augmented reality using a volumetric phantom model[C]. VRST '04: Proceedings of the ACM symposium on Virtual reality software and technology. New York, NY, USA: ACM, 2004: 174–177.

[122] BERGER M O. Resolving occlusion in augmented reality: a contour based approach without 3d reconstruction[C]. CVPR. Washington, DC, USA: IEEE Computer Society, 1997: 91–96.

[123] KASS M, WITKIN A, TERZOPOULOS D. Snakes: Active contour models[J]. International journal of computer vision, 1988, 1(4):321–331.

[124] KIM H, SOHN K. 3D reconstruction from stereo images for interactions between real and virtual objects[J]. Signal Processing: Image Communication, 2005, 20:61–75.

[125] BAI X, WANG J, SIMONS D, et al. Video snapcut: Robust video object cutout using localized classifiers[J]. ACM Trans. Graph. (Proc. SIGGRAPH 2009), 2009.

[126] PRICE B L, MORSE B S, COHEN S. LIVEcut: Learning-based interactive video segmentation by evaluation of multiple propagated cues[C]. ICCV. 2009.

[127] KAKUTA T, VINH L B, KAWAKAMI R, et al. Detection of moving objects and cast shadows using a spherical vision camera for outdoor mixed reality[C]. VRST '08: Proceedings of the 2008 ACM symposium on Virtual reality software and technology. New York, NY, USA: ACM, 2008: 219–222.

[128] ELGAMMAL A M, HARWOOD D, DAVIS L S. Non-parametric model for background subtraction[C]. ECCV. London, UK: Springer-Verlag, 2000: 751–767.

[129] ZHONG J, SCLAROFF S. Segmenting foreground objects from a dynamic textured background via a robust kalman filter[C]. ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision. Washington, DC, USA: IEEE Computer Society, 2003: 44.

[130] RAO N I, DI H, XU G. Panoramic background model under free moving camera[C]. FSKD '07: Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery. Washington, DC, USA: IEEE Computer Society, 2007: 639–643.

[131] BOYKOV Y, VEKSLER O, ZABIH R. Fast approximate energy minimization via graph cuts[J]. IEEE Trans. Pattern Anal. Mach. Intell., 2001, 23(11):1222–1239.

[132] CRIMINISI A, CROSS G, BLAKE A, et al. Bilayer segmentation of live video[C]. CVPR. Washington, DC, USA: IEEE Computer Society, 2006: 53–60.

[133] SUN J, ZHANG W, TANG X, et al. Background cut[C]. ECCV. 2006: 628–641.

[134] YIN P, CRIMINISI A, WINN J, et al. Tree-based classifiers for bilayer video segmentation[C]. CVPR. 2007: 1–8.

[135] CASTLE R O, GAWLEY D J, KLEIN G, et al. Video-rate recognition and localization for wearable cameras[C]. Proc 18th British Machine Vision Conference, Warwick, Sept 11-13, 2007. 2007: 1100–1109.

[136] LIU T, KENDER J R. Optimization algorithms for the selection of key frame sequences of variable length[C]. ECCV (4). 2002: 403–417.

[137] GIANLUIGI C, RAIMONDO S. An innovative algorithm for key frame extraction in video summarization[J]. Journal of Real-Time Image Processing, 2006, 1(1):69–88.

[138] TRUONG B T, VENKATESH S. Video abstraction: A systematic review and classification[J]. ACM Trans. Multimedia Comput. Commun. Appl., 2007, 3(1):3.

[139] ANGELI A, FILLIAT D, DONCIEUX S, et al. A fast and incremental method for loop-closure detection using bags of visual words[J]. IEEE Transactions on Robotics, Special Issue on Visual Slam, 2008.

[140] EADE E, DRUMMOND T. Unified loop closing and recovery for real time monocular slam[C]. British Machine Vision Conference (BMVC). 2008.

[141] BROWN M, LOWE D G. Recognising panoramas.[C]. ICCV. 2003: 1218–1227.

[142] HARTLEY R I. Self-calibration from multiple views with a rotating camera[C]. ECCV. vol 1. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1994: 471–478.

[143] WANG J, BHAT P, COLBURN A, et al. Interactive video cutout[J]. ACM Trans. Graph., 2005, 24(3):585–594.

[144] ZHONG B, YAO H, SHAN S, et al. Hierarchical background subtraction using local pixel clustering[C]. ICPR. 2008: 1–4.

[145] COMANICIU D, MEER P. Mean shift: A robust approach toward feature space analysis[J]. IEEE Trans. Pattern Anal. Mach. Intell., 2002, 24(5):603–619.

[146] SMET P D, PIRES R L V. Implementation and analysis of an optimized rainfalling watershed algorithm[C]. IS&TSPIE's 12th Annual Symposium Electronic Imaging 2000. 2000: 759–766.

[147] KLEIN G, MURRAY D. Parallel tracking and mapping on a camera phone[C]. Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on. 2009: 83–86.

[148] DANIEL W, ALESSANDRO M, TOBIAS L, et al. Real-time panoramic mapping and tracking on mobile phones[C]. Proceedings of IEEE Virtual Reality Conference 2010 (VR'10). 2010.

[149] 林倞, 杨珂, 王涌天, et al. 移动增强现实系统的关键技术研究[J]. 中国图象图形学报, 2009, 14(3):560–564.

# 攻读博士学位期间主要研究成果

## 论文发表

1. Guofeng Zhang, **Zilong Dong**, Jiaya Jia, Tien-Tsin Wong, Hujun Bao. Efficient Non-Consecutive Feature Tracking for Structure-from-Motion[C]. European Conference on Computer Vision (ECCV), 2010.

2. **Zilong Dong**, Lei Jiang, Guofeng Zhang, Qing Wang, Hujun Bao. Live Video Montage with a Rotating Camera[J]. Computer Graphics Forum (CGF), 2009, 28(7): 1745-1753.

3. **Zilong Dong**, Guofeng Zhang, Jiaya Jia, Hujun Bao. Keyframe-Based Real-Time Camera Tracking[C]. IEEE International Conference on Computer Vision (ICCV), 2009.

4. Guofeng Zhang, **Zilong Dong**, Jiaya Jia, Liang Wan, Tien-Tsin Wong, Hujun Bao. Refilming with Depth-Inferred Videos[J]. IEEE Transactions on Visualization and Computer Graphics (TVCG), 2009, 15(5):828-840.

5. **董子龙**, 章国锋, 邵元龙, 华炜. 基于汉字标志的增强现实系统[J]. 中国图象图形学报, 2009, 14(7):1463-1468.

6. 姜翰青, 章国锋, **董子龙**, 华炜, 刘新国, 鲍虎军. 基于图像序列的交互式快速建模系统[J]. 计算机辅助设计和图形学报, 2008, 20(9):1196-1203.

7. 章国锋, 秦学英, **董子龙**, 华炜, 鲍虎军. 面向增强视频的基于结构和运动恢复的摄像机定标[J]. 计算机学报, 2006, 29(12):2104-2111.

8. Rui Wang, Wei Hua, **Zilong Dong**, Hujun Bao, Qunsheng Peng. Synthesizing trees by plantons[J]. The Visual Computer, 2005, 22(4):238-248.

# 致 谢

值此论文完稿之际，感到多年来受到了太多人的关心和支持，谨以此致谢。

首先，对我的导师鲍虎军教授致以深深的谢意。在我攻读博士学位的整整七年光阴，鲍老师一致给予我耐心的指导和帮助，我的每一点进步、每一份成果，无不包含着他的敦敦教诲和殷殷期望。从博士论文的选题、探索到完成，鲍老师关注每一个细节，将一个科研的门外汉领进了计算机科学的殿堂，在学术上和生活上都使我受益匪浅，更让我领悟了“学高为师、身正为范”的真谛。

感谢任重和王锐师兄，他们是我初涉科研的学习楷模。他们坦率的为人、独到的见解以及未来的把握都给我留下了深刻的印象，使我在研究中深受启发。

感谢已经毕业离校的刘华、李岩、许栋师兄，虽然在科研上没有直接的合作，但是他们对后学的关心和指导让我深知博士学习的孤独和压力，知其可为，亦知其不可为。

尤其感谢章国锋博士，因为我的每一篇论文都是和章博士合作的成果。章博士在我的博士论文工作上付出的心血和指导让我心怀感激，他做事一丝不苟，对科研倾情投入，完成一个卓越博士难有的贡献，始终是我学习的榜样。

感谢香港中文大学的贾佳亚教授，在我发表第一篇国际会议论文的过程中，他给予我很多指导和帮助，让我了解了如何进行高水平的科学研究。

感谢实验室的每一位同学，特别是陈禄、冯伟、刘峰、姜翰青、潘明皓、邵元龙、沈中伟、施晓晗、田丰林、姚冠红、张沐阳，每一次和你们交谈辩论，都能碰撞出思想的火花，祝大家永远幸福快乐。

感谢我的兄弟们，在研究工作遇到瓶颈的时候，总是你们陪我品茶聊天，使我可以重振旗鼓，投入实验室工作，特别是陈亮、丁轶群、富浩、林存宝、林剑、何耀杨、毛方贵、汪洪波、许秋儿、朱文敏，祝你们永远幸福快乐。

感谢我亲爱的父母，父母近三十年无私关爱与付出，将我培养成人，一篇博士论文远非我报答养育之恩的终点，我将继续努力，以更大的成绩回报父母，以及所有关心过我成长的亲人们。

最后，感谢评审老师百忙之中抽得时间对我的研究工作提出批评指正。

董子龙

2010年7月