上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

THESIS OF BACHELOR

论文题目：面向移动终端的实时物体追踪技术

学生姓名：冯乐乐

学生学号：5100379048

专　　业：软件工程

指导教师：杨旭波

学院(系)：软件学院

面向移动终端的实时物体追踪技术

摘要

近年来,物体识别和姿态估计越来越成为计算机视觉和增强现实方面的两个重要课题。物体识别旨在根据已知的特征描述来识别图片中的物体，基于模型的姿态估计则决定了已知物体在摄像机所处坐标系内的六个自由度以在摄像机坐标系中绘制三维图形。跟 PC 相比,移动终端具有很多限制。因此,在移动终端上,对实时物体追踪的效果和性能之间的权衡,成为当前一个非常热门的研究课题。

本文分析了现有的基于自然特征的物体识别和追踪技术,设计并实现了一款基于移动终端的实时物体追踪系统,以用于某些 AR 应用。传统的 SIFT 算法具有很好的健壮性，但是特征描述计算量大。FAST 角点检测很快，但是生成的关键点特征信息量少。我们通过实现一个使用 FAST 算子更改后的 SIFT 算法，来实现移动终端的物体识别。而姿态估计可以从识别出的原目标和帧图像的匹配关系得到变换矩阵，以此来显示三维物体。除此之外，我们还实现了一个 PC 端的管理工具，用于统一管理所有预处理的目标图片。我们将详细介绍是如何修改 SIFT 算法使得它更适合移动终端的实时物体追踪。我们对系统的健壮性和性能等方面进行了评估，并实现了一个在现实生活中的演示应用。最后，我们给出了未来工作的进展方向。

关键词：物体识别，姿态估计，增强现实，自然特征，物体追踪

REAL-TIME OBJECT TRACKING AND POSE ESTIMATION ON MOBILE PHONES

ABSTRACT

In the recent years, object recognition and pose estimation are the two main topics in the Computer Graphics and Augmented Reality (AR). Object recognition uses approaches in feature descriptors to recognize some targets in pictures. And pose estimation determines the six degrees of freedom (6DOF) in screen coordinates and then renders some 3D objects according to it. Compared with the Tablet PCs, phones are embedded systems with severe limitations in both the computational facilities and memory bandwidth. Therefore, to trade off the effect of real-time object tracking and performance has become a very popular research topic.

In this paper, we analyze the existing object recognition and tracking technology based on natural features, and implement a real-time object tracking system on mobile phones for some AR applications. The original SIFT is known to be a strong, but computationally expensive feature descriptor. FAST corner detection is fast, but the generated information of key points is poor. We achieved the object recognition on mobile phones through a modified version of the SIFT algorithm using FAST. And pose estimation can be obtained from the transformation matrix matching between the identified targets and frame, and then display three-dimensional objects in the screen. In addition, we implemented a PC client management tool for management of all preprocessed target pictures. We give detail descriptors on how we modified the algorithm to make it suitable for mobile phones. We present evaluations on robustness and performance of the system, and implement a demo application in real world. Finally, we discuss the direction of future works.

Key words: object recognition, pose estimation, Augmented Reality, natural features, object tracking

目 录

第一章 绪论 1
1.1 研究意义 1
1.2 相关工作 1
1.3 研究目标 3
1.4 本文工作成果与贡献 3
1.5 本文结构 4
第二章 自然特征的检测 5
2.1 SIFT 特征变换 5
2.1.1 关键点定位 5
2.1.2 关键点梯度方向的确定 7
2.1.3 特征描述 7
2.1.4 SIFT 算法的优缺点 8
2.2 FAST 角点检测 8
2.3 本章小结 9
第三章 系统架构 10
3.1 软件架构设计 10
3.1.1 概览 10
3.1.2 PC 端软件设计 10
3.1.3 移动端软件设计 12
3.2 本章小结 12
第四章 移动终端的自然特征追踪 13
4.1 PhonySIFT 13
4.1.1 关键点检测 13
4.1.2 描述子的创建 13
4.1.3 描述子的匹配 13
4.2 异常点处理 13
4.2.1 比率检验和对称检验 13
4.2.2 描述子方向检验 14
4.2.3 几何检测 14
4.2.4 单应性检验 15
4.3 目标数据的预处理 15
4.3.1 特征检测 15
4.3.2 数据规格 16
4.4 本章小结 16
第五章 移动终端的姿态估计 17
5.1 姿态估计 17
5.2 三维图像渲染 18

5.3 本章小结 19
第六章 实验和评估 20
6.1 追踪的数据库大小 20
6.2 追踪的目标种类 20
6.3 新增的异常点处理 22
6.4 追踪的健壮性 22
6.5 性能 23
6.6 应用 24
6.7 本章小结 26
第七章 总结和展望 27
7.1 本文工作总结 27
7.2 未来工作 27
参考文献 29
谢辞 30

第一章 绪论

1.1 研究意义

近年来，物体识别(Object Recognition)和姿态估计(Pose Estimation)越来越成为计算机视觉(Computer Vision)和增强现实(Augmented Reality, AR)方面的两个重要课题。

增强现实技术的目标是在屏幕上把虚拟世界嵌套在现实世界，是一种结合虚拟化技术再来观察世界的方式。增强现实一般包含三个方面内容：将虚拟物与现实结合，即时互动和三维。

当前 AR 技术的重点在于物体识别和姿态估计，并在现实世界的基础上显示或增强虚拟对象。物体识别旨在根据已知的特征描述来识别图片中的物体。基于模型的姿态估计则决定了已知物体在摄像机所处坐标系内的六个自由度(Six Degrees Of Freedom, 6DOF)。这些技术在医疗、广告、网上购物、游戏[1]、儿童教育等各种领域具有广泛的应用前景。例如在医疗领域，医生可以利用增强现实技术，轻易地进行手术部位的精确定位。

在过去的几年里，移动终端因其诱人的价格和便携的大小等优势吸引了 AR 技术的目光。但是跟 PC 相比，移动终端具有很多限制，包括有限的计算能力(如低吞吐量，不支持浮点值等)和内存带宽(有限的容量，低内存等)。因此，在移动终端上，对实时物体追踪的效果和性能之间的权衡，成为当前一个非常热门的研究课题。

1.2 相关工作

目前领域中已有一些用于物体识别和姿态估计的工具。ARToolKit[2]是最早被广泛用于 AR 应用的工具之一，它依靠强度阈值(intensity thresholding)识别标记，但是对于大规模的物体识别其性能会急剧下降。

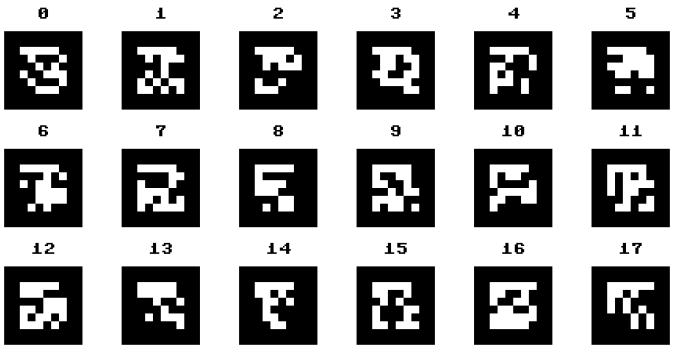

图 1-1 ARTag 使用的标记

ARTag[3]是在 ARToolKit 的基础上，利用了当时计算处理水平的高速发展而被提出的。它使用了更复杂的图像处理基础以及一种类似二维码系统的方法克服了可靠性问题，并对光照影响有了更强的健壮性。但和 ARToolKit 一样，ARTag 也使用了基于标记的追踪技术。图 1-1 显示了一些 ARTag 使用的标记。可以看出，这些标记类似于今天的二维码。

Studierstube Tracker[4]是一个基于移动平台的轻量级追踪库,它使用数字化编码的标记进行基准点识别。它包含了当今绝大多数用于增强现实的标记种类，包括：

(1) 帧标记（Frame markers），如图 1-2（a）。它的追踪鲁棒性很大程度上是依靠与阈值图像上的黑色边框的强对比度实现的。它的原理主要是通过边框内部黑白相间的方块进行编码得到一个唯一的标记符，而内部的图像可以被用来进一步检验识别的正确性。这类标记通常被用于广告，因为厂商可以把他们的 logo 嵌入到标记符里。它的缺点主要是被有色边框占用的区域面积较大，而且不能有任何遮盖物挡住边框。

(2) 分割标记（Split markers），如图 1-2（b）。由于帧标记仍然具有一圈黑色边框用于定义标记的范围，分割标记使用了两个单独的条形码来进一步减少占用的区域。和帧标记类似，这种标记内部的图像不会被用于追踪，因此可以使用任意图案代替。因为它的四周仅有上下两个条形码，因此可以部分遮挡标记而不影响追踪效果。

(3) 点标记（Dot markers），如图 1-2（c）。绝大多数情况下，标记被认为是不受欢迎的，因为它们要覆盖在特定物体上。因此，尽可能少的减少覆盖面积是非常必要的。为了改善这一点，Rohs et al.[5]提出了一种追踪方法，这种方法通过在地图或其他有纹理的平面上绘制点标记来提高追踪的速度和鲁棒性。点标记由二维的、由白色圆环包围的黑色圆圈组成的方格构成。这些点很小，通常只覆盖了整个纹理区域的 1%左右。

(4) 数据矩阵标记（DataMatrix markers），如图 1-2（d）。数据矩阵是一种针对 2D 条形码的 ISO 标准（ISO/IEC16022）。

(5) ID 标记（ID markers），如图 1-2（e）。它是 ARTag 使用的标记种类。

(6) 模板标记（Template markers），如图 1-2（f）。它是 ARToolKit 使用的标记种类。

图 1-2 六种标记种类：(a)帧标记 (b)分割标记
(c)点标记 (d)数据矩阵标记 (e)ID 标记 (f)模板标记

上述基于标记的追踪技术(Marker-Based Tracking)虽然取得了较好的识别和追踪效果，但是用于识别的标记(如二维码等)没有明确意义，尽管有些占用的区域比较小，但在外观上仍然破坏了识别对象的美观性，很难吸引用户的注意力，而开发者和厂商也需要花费大量冗余的时间提前打印它们。除此之外，由于标记的可识别区域比较集中（通常位于图像的四个边框），所以这些技术大多对遮挡(Occlusions)具有较差的健壮性。因此，基于少量标记(marker-less)或自然特征(natural features-based)的识别和追踪就显得尤为重要。

目前已有的自然特征种类有形状特征（Shape Recognition）[6]以及具有更复杂纹理信息的自然特征[7, 8, 9, 10]。

图 1-3 形状特征[6]

形状特征，如图 1-3，主要使用轮廓凹度（Contour concavities）生成投影不变特征量进行识别和追踪。算法包含两个主要步骤：

(1) 首先进行形状提取，得到一个有序顶点列表，并过滤掉轮廓长度或面积较小的区域。再检查轮廓信息中的所有凹面，并应用中值过滤得到平滑的凹面。

(2) 然后进行形状识别。对于一个凹面，算法将找到一条双切线及两个切点，并找到两个切点在凹面上的投影点。这四个点加上一条切线以及两条投影切线就是该凹面对应的形状特征。如图 1-4 所示。

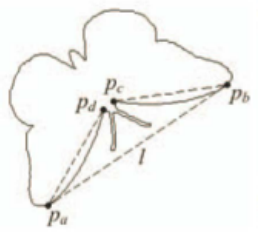

图 1-4 轮廓凹面特征[6]

形状特征主要被应用于公共场所标志牌、游戏卡牌等形状特征明显的领域。该方法计算量小，但其局限性较大，只能用于识别非凸形状，并要求图像至少包含两个凹面。而且由于轮廓的检测是基于阈值检测的，因此要求图片的轮廓相对于背景要具有较高的对比度。

相比于以上提到的基于标记的追踪方法以及基于形状特征的追踪方法，具有复杂纹理信息的自然更特征具有明显的优势。这些自然特征有非常丰富的图片含义，并且美观度高，容易被用户接受，而且不需要开发者或厂商专门打印出来或印在需要识别的物体上。它可以是一本儿童读物中的一幅插画，也可以是地铁里的一张海报，甚至是大富翁棋中的一张道具卡片，都可以被用于增强现实的技术中。因此，本文将以这类自然特征作为进行追踪的特征种类。

1.3 研究目标

通过上节中对各种特征追踪方法的分析和比较，我们得知，具有复杂纹理信息的自然特征的追踪技术更符合市场和工业的需求。本文旨在分析基于自然特征的物体识别和追踪技术，设计并实现一款基于移动终端的实时物体追踪系统。

研究内容如下：

(1) 系统支持识别具有一定自然特征信息(如具有一定关键点且分布均匀、对比度适当且画面清晰、无较多重复图案等)的图片；

(2) 系统可以在一定指标下追踪目标图片，并求解出相机与图片的三维变换矩阵。

为了评估系统的健壮性，本文规定了一些指标进一步测试系统。这些指标是：

(1) 环境条件为可接受的光照条件；

(2) 目标图片在摄像机视野范围内的部分超过整体部分的 50%；

(3) 目标图片所在平面的正交平面与摄像机视角倾斜角度范围在 30°以内；

(4) 目标图片和摄像机之间的遮挡物范围不超过目标图片的 50% (前提是目标物体的特征信息均匀分布)；

(5) 目标图片和摄像机之间的距离在 1m 以内。

1.4 本文工作成果与贡献

本文根据已被提出的一种实时检测和追踪算法 PhonySIFT[10]，实现了一个可用于移动终端的实时物体识别和追踪系统。该算法结合了两种自然特征提取算法，SIFT 和 FAST，取得

了较好的识别和匹配效果。

在该算法的基础上，我们实现了一个包含 PC 端子系统和移动端子系统的实时物体追踪系统。通过 PC 端子系统，我们可以管理所有预处理图片并导出数据库文件作为移动端子系统的输入。移动端子系统则通过预加载数据库文件，向系统注册所有可实时检测到的目标图片，并在识别到某目标图像后，在屏幕相应位置绘制 3D 模型。

除了在文献中提到的三种异常点处理方法，我们还加入了两种新的异常点处理步骤，即比率检验和对称检验，进一步减少了异常点的数量，提高匹配点对的可靠性。

最后，我们将本系统应用于现实世界中的一个场景，即检测和显示商品信息。通过该应用，用户可以实时识别多种商品，并查阅特定的商品信息。

1.5 本文结构

本文由七章组成。

第一章阐述了本文的研究意义和目标，并分析当前已有的一些基于标记的追踪技术以及基于自然特征的追踪技术的优缺点。

第二章将简要描述一些自然特征的匹配技术，包括了尺度不变的特征变换（SIFT）和 FAST 角点特征检测。

第三章为系统架构设计，将详细说明本系统的软件架构设计，包括 PC 端和移动端两个子系统。

第四章将详细阐述本系统在移动终端的物体识别技术以及 PC 端预处理目标图像的详细过程。

第五章介绍了移动终端的姿态估计，将详细阐述如何根据上一章的结果计算，在移动终端计算单应性矩阵，并显示三维物体。

第六章将详细评估系统各方面的指标，包括目标种类、数据库大小、健壮性以及性能。同时，我们还展示了一个可用于现实世界的演示应用。

第七章是对本文工作的总结，并对下一阶段的研究进行展望。

第二章 自然特征的检测

2.1 SIFT 特征变换

尺度不变的特征变换（Scale-invariant feature transform, SIFT）[11]，由于其对旋转、尺度缩放等具有很好的健壮性，因此在目标识别领域具有非常广泛的应用。

SIFT 算法主要包含三个步骤：关键点定位，关键点梯度方向的确定以及特征描述。

2.1.1 关键点定位

SIFT 算法是在不同尺度空间上查找关键点的，而尺度空间的获取需要使用高斯模糊来实现。

高斯模糊是一种图像滤波器，它使用正态分布（高斯函数）计算模糊模板，并使用该模板与原图像做卷积运算，达到模糊图像的目的。

尺度空间（scale space），指的是在图像处理模型中引入一个被视为尺度的参数，通过连续变化的尺度参数，获得多尺度下的尺度空间表示序列，并对这些序列进行尺度空间特征向量的提取。尺度空间的方法将传统的单尺度图像处理技术归入到尺度不断变化的动态分析框架中，更容易获取图像的本质特征。尺度空间中各尺度图像的模糊程度逐渐变大，能够模拟人在逐渐远离目标时目标在视网膜上的成像过程。一个图像的尺度空间 L(x, y, σ)定义为高斯函数 G(x, y, σ)和原图像 I(x, y)的卷积：

$$L(x,y,s)=G(x,y,s)*I(x,y) \tag{2-1}$$

其中，*表示卷积运算，高斯函数 G(x, y, σ)定义如下：

$$G(x,y,s)=\frac{1}{2ps^2}e^{-\frac{(x-m/2)^2+(y-n/2)^2}{2s^2}} \tag{2-2}$$

其中，m、n 表示高斯模板的维度。(x, y)表示图像的像素位置，σ 表示高斯核。σ 越大，图像模糊程度越高，对应的尺度也越大。图 2-1 形象的显示了尺度空间的意义。

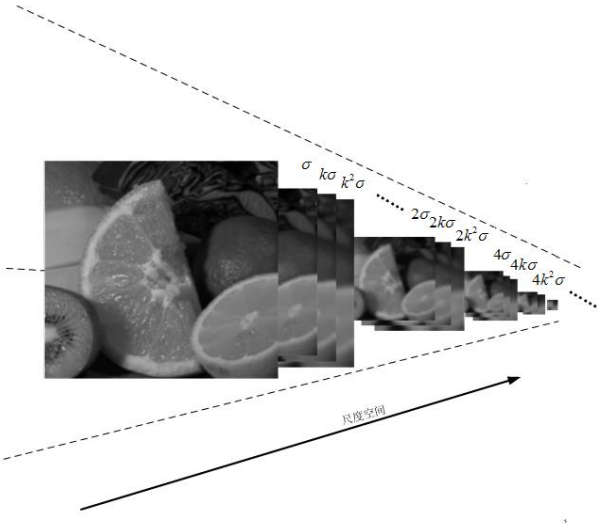

图 2-1 尺度空间[11]

在关键点定位中，作者使用高斯模糊、在不同尺度大小的图片上进行高斯卷积，建立了代表尺度空间的高斯金字塔。高斯金字塔的构建过程，一般首先把图像扩大一倍，在扩大的图像基础上对该尺度下的图像进行高斯模糊，构成高斯金字塔中的最底层，每一层称为一个八度空间（octave）；再以该八度空间下最模糊的图像进行降采样，作为下一个八度空间的初始图像。在高斯金字塔中，八度空间参数（o）和该尺度空间下的层数（s）共同决定了金字塔中唯一的一幅图像。对于（o, s）确定的一幅图像来说，高斯核为

$$s = s_0 2^{o+\frac{s}{S}} \tag{2-3}$$

其中，我们可以在每个八度空间中检测 S 层点，则高斯金字塔中每个八度空间需要有 S+3 层图像。图 2-2 显示了一个高斯金字塔。

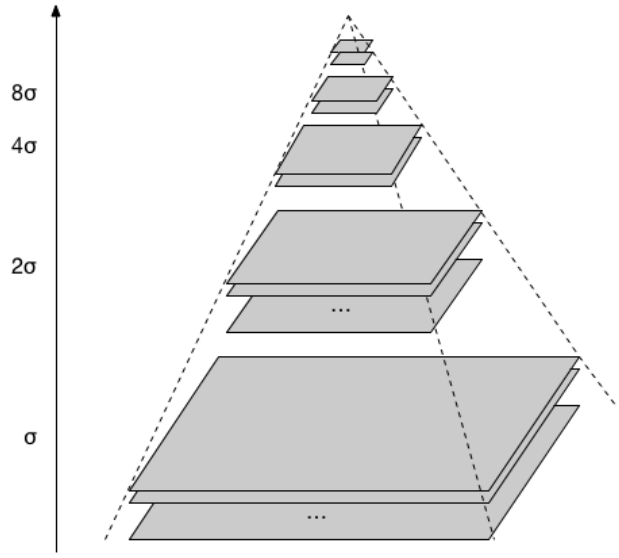

图 2-2 高斯金字塔

在得到高斯金字塔后，我们需要计算高斯差分金字塔（Difference of Gaussians, DoGs），使用高斯差分算子进行极值检测，定位关键点的位置。在实际计算中，即使用高斯金字塔每个八度空间中上下两层图像相减，得到高斯差分图像。高斯差分金字塔中每个八度空间有 S+2 层图像。

最后，我们在三维空间中定位极大值点和极小值点作为候选的关键点。具体做法是，我们比较当前特征点的灰度值和其周围 26 个点的灰度值大小，这 26 个点包括：当前尺度下该点的 8 个邻域、前一个尺度和后一个尺度下该点最近的 9 个点。如图 2-3 所示。

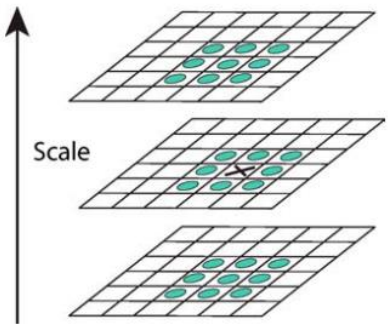

图 2-3 在三维空间中定位关键点

虽然在 DoGs 中定位关键点可以得到很好的尺度不变性，但是其计算量对于计算能力有

限的移动终端是非常繁重的。

2.1.2 关键点梯度方向的确定

为了使特征描述具有旋转不变性，SIFT 算法利用图像的局部特征为每一个关键点分配了一个梯度方向。

对于在 DoGs 中检测到的关键点，我们需要计算其在高斯金字塔对应图像中邻域窗口内像素的梯度和方向分布特征。关键点的梯度方向是基于统计信息确定的。Lowe 提出，通过计算关键点邻域内的像素梯度和方向，形成一个梯度方向直方图，并把直方图中的最高峰对应的梯度方向值作为该关键点的梯度方向。梯度的模和方向定义如下：

$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2} \tag{2-4}$$

$$q(x,y)=\tan^{-1}\left(\frac{L(x+1,y)-L(x-1,y)}{L(x,y+1)-L(x,y-1)}\right) \tag{2-5}$$

构造梯度直方图时，横坐标将圆周 360° 划分为 36 个坐标，并根据邻域点的方向、梯度幅值以及距离关键点的远近构建梯度直方图。其中，方向决定其属于梯度直方图中的哪个横坐标，而幅值和距离关键点的远近决定了加权后的需要增加到对应直方柱的值。

2.1.3 特征描述

最后一步需要计算每个关键点的特征描述。和计算关键点梯度方向类似，特征描述也需要在对应的高斯图像上进行。首先，根据关键点的尺度选择高斯金字塔中的图像，然后计算关键点邻域范围内各点的梯度方向和梯度幅值，并根据上一步中得到的关键点的梯度方向更新这些梯度方向，以达到描述的方向不变性。

方向不变性完成后，我们把关键点的邻域划分为 d*d 个子区域（通常 d = 4 或 d = 3），每个子区域大小和计算关键点梯度方向时的邻域大小相同。每个子区域将作为一个种子点，我们计算每个种子点的量化后的方向（量化个数通常为 8 或 4），其方法和梯度直方图计算相同。为了去除光照对描述子的影响，我们还要对梯度直方图进行归一化处理。最后，将每个直方图中的数据合并，即得到了特征描述向量。Lowe 建议计算 4*4 个子区域中的 8 个量化方向的梯度信息，得到维度为 4*4*8 = 128 的特征描述向量。

图 2-4 显示了维度为 128 的特征描述向量的生成过程。

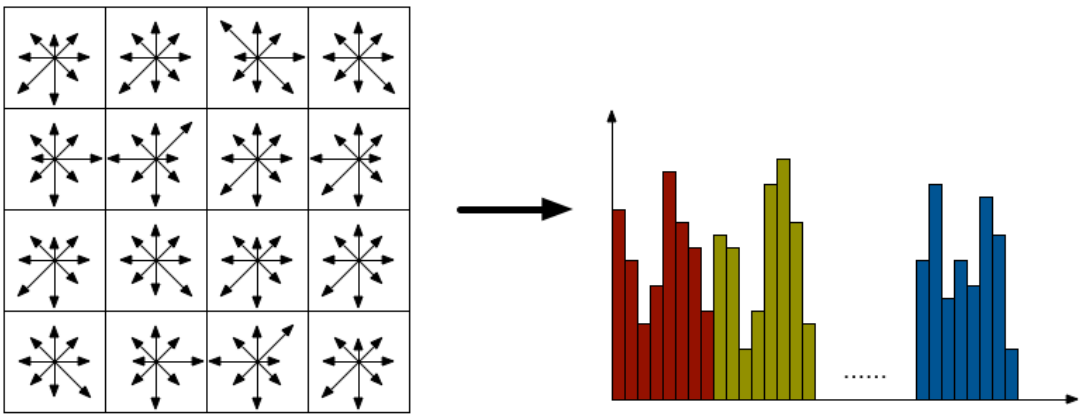

图 2-4 维度为 128 的特征描述向量

大多数 SIFT 算法使用 8 个量化方向以及 4x4 个子区域进行计算，这提供了很好的健壮性，但是需要创建一个较大的特征描述向量（包含了 128 个元素）。

2.1.4 SIFT 算法的优缺点

SIFT 算法在图像的特征提取上有很多优势，如对旋转、尺度缩放有很好的健壮性。但是，SIFT 算法计算量较大，实时性不高，尤其对于移动终端这样计算能力和资源都十分有限的平台，无法发挥其优势。

2.2 FAST 角点检测

目前已知很多种特征检测的方法，如之前提到的 SIFT，并且它们其中有很多效果非常不错，但是，当从一个实时运行的系统角度出发，这些算法的速度都不够快。尤其对于移动终端的大部分设备来说，它们的可计算资源都是十分受限的。

作为上述问题的一个解决方法，FAST（Feature from Accelerated Segment Test）[12]算法由 Edward Rosten 和 Tom Drummond 在他们 2006 年的论文中被提出。FAST 算法主要用于检测图像中的角点，在这样的点附近通常会有较大的灰度值变化。

对于图像中的一个像素点，FAST 通过检验该像素点周围固定半径的圆周上其他像素点的像素值，若在该圆周上有一定数目的连续像素点，它们的像素值和原像素的像素值差，大于或者小于某阈值，那么就认为它是一个角点。具体步骤如下：

(1) 从图片中选取一个像素点 p，下面我们将判断它是否是一个特征点。我们首先把它的密度（即像素值）设为 I_p。
(2) 设定一个合适的阈值 t。
(3) 考虑该像素点周围的 16 个像素。（如图 2-5）
(4) 现在，如果在这个大小为 16 个像素的圆上有 n 个连续的像素点，它们的像素值要么都比 $I_p + t$ 大，要么都比 $I_p - t$ 小，那么它就是一个角点。n 在这里被设定为 12。
(5) 我们现在提出一个高效的测试，来快速排除大部分是非角点的点。该测试仅仅检查在位置 1、9、5 和 13 四个位置的像素（首先检查 1 和 9，看它们是否亮于或暗于阈值。如果是，再检查 5 和 13）。如果 p 是一个角点，那么上述四个像素点中至少有 3 个应该必须都大于 $I_p + t$ 或者小于 $I_p - t$（因为若是一个角点，超过四分之三个圆的部分应该满足判断条件，半圆比包含上述某三个点）。如果都不满足，那么 p 不可能是一个角点。完整的分段测试可以被用于接受的所有候选点。

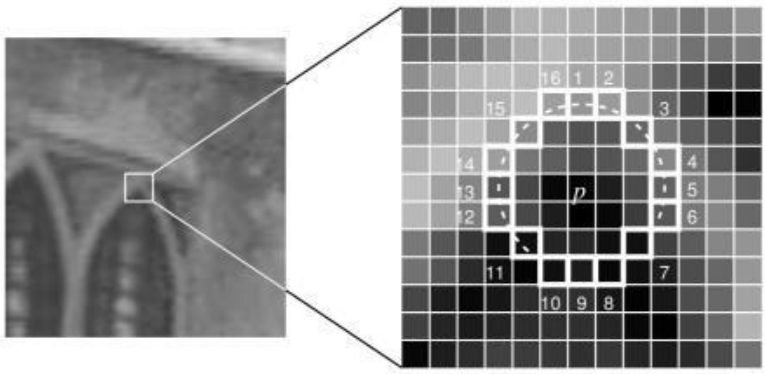

图 2-5 FAST 角点检测[12]

使用上述方法，有一个明显缺点就是相邻的多个特征点都会被检测到，而这个问题可以通过 non-maximal suppression 解决。Non-maximal suppression 为每个检测到的特征点计算分数函数（score function），得到它的分数。该分数是特征点和它周围圆周上所有像素点的绝对偏差的和。考虑两个相邻的特征点，并比较它们的分数值。分数值较低的点将会被剔除。

2.3 本章小结

本章介绍了系统使用了两种基本自然特征检测算法：SIFT 和 FAST。SIFT 算法具有很好的特征检测和匹配效果，不但对旋转、尺度缩放、亮度变化等具有很好的健壮性，而且对于视角变化、仿射变换、噪声也保持一定的稳定性。但其计算量大，不适合计算资源有限的移动终端系统。另一方面，FAST 角点检测具有很好的实时性，但检测出来的特征点特征信息有限，无法得到很好的匹配效果。

针对上述问题，我们结合了两种算法的优点，实现了一种用 FAST 算子代替 SIFT 算子的 PhonySIFT 算法（详见第四章）。

第三章 系统架构

3.1 软件架构设计

3.1.1 概览

本章将讲述软件的系统架构设计。总体架构如图 3-1 所示：

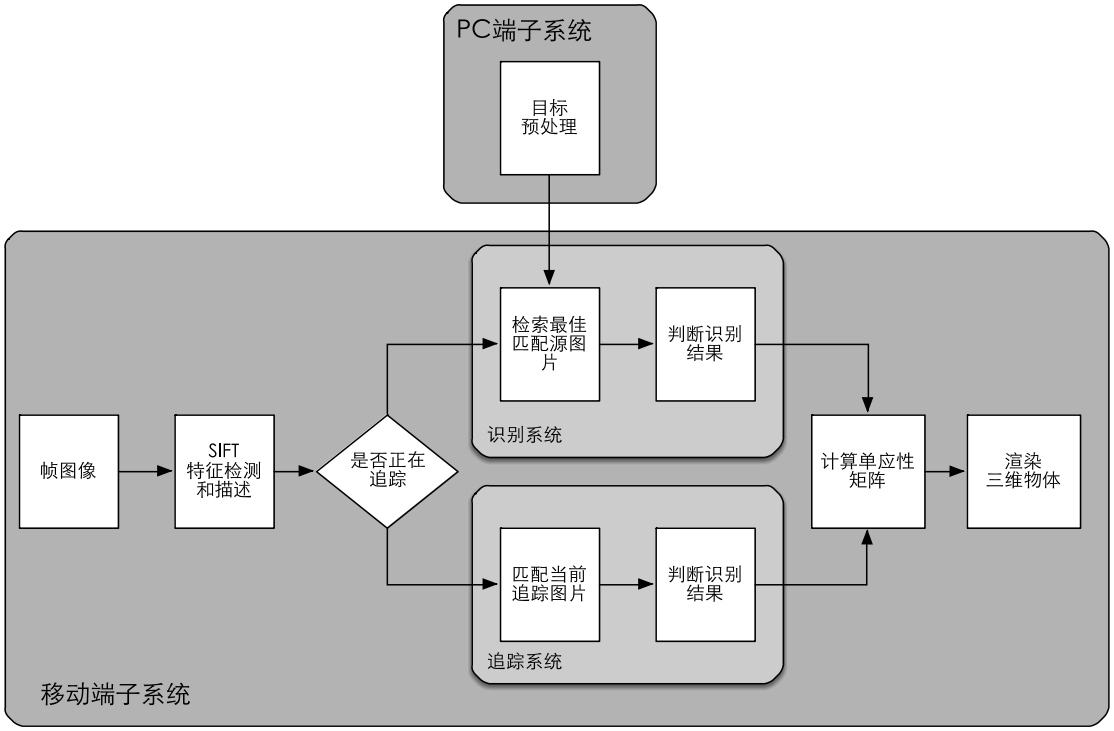

图 3-1 系统架构设计图

系统分为 PC 端和移动端两个子系统。其中 PC 端子系统负责图片目标的预处理，它的输出将会成为移动端子系统的一个输入；移动端子系统负责从摄像机读取帧图片，进行实时物体的识别和追踪。

3.1.2 PC 端软件设计

作为辅助子系统，PC 端子系统的主要功能是管理所有可识别的目标图片，进行预处理，并导出二进制文件作为移动端子系统的输入。图 3-2 显示了一个正在工作的 PC 子系统。

图 3-2 PC 子系统：**Target Manager**

本系统的 PC 端子系统包括以下功能：

(1) 添加目标图片，并设置目标名称、长宽值信息；

(2) 应用 PhonySIFT 进行特征检测和特征描述；

(3) 查看关键点，并进行静态匹配测试；

(4) 将(1)(2)中生成的目标图片信息、关键点和特征描述导出到二进制文件中。

静态匹配测试指的是在 PC 端非实时进行图片匹配，这样可以初步验证算法的正确性，并观察算法在不同情况下的健壮性。下列图片显示了在旋转、遮挡、倾斜、模糊和远距离情况下 PC 端的静态测试结果。

图 3-3 静态匹配测试效果。
左图：旋转测试；右图：远距离测试。

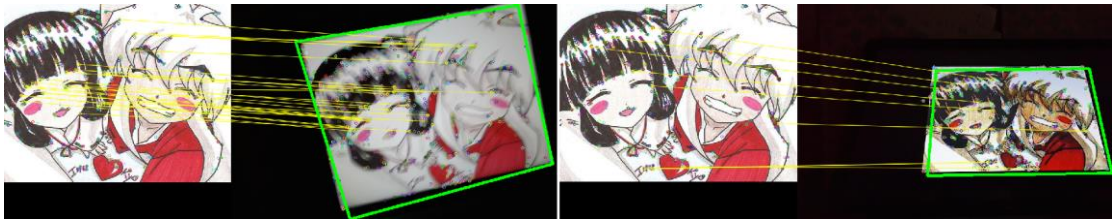

图 3-4 静态匹配测试效果。
左图：模糊测试；右图：倾斜测试。

经过 PC 端子系统处理后，将输出包含有所有目标图片的相关信息，以供移动端子系统读取。

3.1.3 移动端软件设计

移动端子系统有六个主要步骤：

(1) 读取并注册预处理目标图片

(2) 使用 PhonySIFT 提取帧图像中的关键点；

(3) 计算特征描述；

(4) 和数据库中数据进行匹配；

(5) 姿态估计；

(6) 渲染三维图形；

研究和技术难点集中在以下几点：

(1) 特征检测

(2) 特征描述和匹配

(3) 异常点移除

(4) 姿态估计

在特征检测中，我们使用了经过 PhonySIFT 算法检测特征点并计算特征描述，然后尝试和预处理的目标图像进行匹配，并将匹配结果经过多层过滤以移除异常点。最后，我们使用过滤后特征匹配进行姿态估计，并最终在对应位置显示三维物体。

3.2 本章小结

本章概括说明了整体的系统架构设计，并简要说明了每个子系统的主要功能。在下一章中，将重点说明移动端的系统实现细节，并详细解释每个难点的实现方法。

第四章 移动终端的自然特征追踪

4.1 PhonySIFT

在下面的小节中，我们将阐述我们更改后的 SIFT 算法，即称为 PhonySIFT 的算法，并详细解释所有的步骤。

4.1.1 关键点检测

传统的 SIFT 算法使用高斯差分金字塔（DoGs）来构造尺度空间，用于检测关键点。然而，这种方法计算量大，不适合用于移动终端的实时追踪系统。因此，我们采用 FAST 角点检测来替换原高斯差分金字塔，并且同样可以提供较好的可靠性。因为 FAST 不会评估关键点的尺度性，因此我们需要为每个目标建立一个图片金字塔（详见 4.3 节），计算不同尺度的关键点描述。

我们仅需要在 PC 端子系统预处理时计算图片金字塔，而在移动端子系统上，我们只需在当前帧图像应用 FAST 算法得到关键点及其梯度方向，并计算其特征描述。这是因为，PC 端得到的图像金字塔已经规定了实时可识别到的目标图片的范围。

4.1.2 描述子的创建

大多数 SIFT 使用一个 4*4 大小的子区域，计算该区域内 8 个量化后的梯度值，一共需要计算 4*4*8=128 个元素。出于性能和内存的考虑，我们仅使用 3*3 大小的子区域，计算该区域内 4 个梯度值，一共计算 3*3*4=36 个元素。根据 Lowe 的论文，这种优化后的方法的可靠性仅比 128 元素的方法可靠性降低最多 10%。最后，我们将得到维度为 36 的特征描述向量，称为描述子。

4.1.3 描述子的匹配

帧图片中的所有特征描述子将会和数据库中的描述子进行匹配。对于一张图片中的每一个描述子将会和另一张目标图片的所有描述子进行比较，并保留和其向量距离最小的两组匹配，分别称为最佳匹配和次佳匹配，再排除异常点（见 4.2 节）。

4.2 异常点处理

4.2.1 比率检验和对称检验

在上一节提到，描述子匹配时，对于每一个描述子我们会保留和其向量距离最小的两个特征描述子，即最佳匹配和次佳匹配。这是为异常点处理的第一层处理做准备，即比率检验和对称检验。

比率检验和对称检验是得到正确匹配的基本检验方法。这也是我们在原实现的基础上新增的两种异常点处理方法。

对称检验指的是，我们将从两个方向进行上述匹配：对于帧图像中的每一个特征描述，在数据库某目标图片上进行匹配；再从数据库某目标图片出发，在帧图像上进行匹配。这样，我们分别得到了两个方向的特征匹配集合。

比率检测将会检验对称检验中每一组匹配的可靠性。这是建立在这样一个前提下：我们认为若最佳匹配的向量距离很小，而次佳匹配的向量距离很大，则这是一个好的匹配。相反地，若最佳匹配和次佳匹配的向量距离值很接近，那么我们很有可能错误地选择出真正的最佳匹配。因此，我们应该排除这样的匹配。在比率检验中，我们将分别检查两个方向的特征

匹配组中所有匹配的最佳匹配和次佳匹配，计算它们的向量距离比率，若比率小于某一个阈值，则移除该匹配。

这样我们得到了两组相对正确的匹配集合，一组是从帧图片到数据库图片，一组是从数据库图片到帧图片。作为对称检验的最后一步，我们将从两组匹配集合中筛选出对称的部分，即对于一组匹配集合中的某一个匹配，它也存在于另一组匹配集合。

通过比率检验和对称检验，我们可以大幅度提高正确匹配的可靠性，同时降低了筛选过后匹配点对的个数，提高了性能。

4.2.2 描述子方向检验

特征描述子包含了该关键点的特征方向，由于我们的追踪目标是平面目标，所以可以认为所有的关键点具有相似的相对特征方向。

我们计算当前帧的关键点方向相对于数据库中对应关键点方向的相对方向，得到一个相对方向值的集合。然后，我们利用四分位点法分析这些相对方向，估测出一个合理的相对方向范围，并以此为依据，排除那些不在此范围内的相对方向对应的特征匹配。

四分位点法是从一组数据中得到和其他数据明显不同的异常点的常用方法。四分位点的意义在于将数据分割成四个概率等大的部分，并以此为根据评估正常数据的范围。其主要流程如下：

(1) 首先对所有相对方向值从小到大进行排序；
(2) 找到中位点 Q_2，对于大小为奇数的集合，直接取中间数据的值；对于大小是偶数的集合，采用中间两个数据的平均值作为中位点；
(3) 找到上分位点 Q_1 和下分位点 Q_3。首先得到上（下）分位点所在的数据集合，即根据上一步得到的中位点 Q_2，所有小于 Q_2 的值作为求上分位点的集合，所有大于 Q_2 的值作为求下分位点的集合。再按照求 Q_2 的步骤分别求出 Q_1 和 Q_3。
(4) 根据 Q_1 和 Q_3 计算合理的相对方向范围。定义：

$$Q = (Q_3 - Q_1)' \ Coefficient \qquad (4\text{-}1)$$

其中，coefficient 为某一常数，则合理的相对方向范围为$(Q_1 - Q, Q_3 + Q)$。

4.2.3 几何检测

几何检验使用了图片的几何意义。首先，我们将所有的关键点按照它们匹配距离的从小到大（匹配距离即特征描述子的向量距离）排序。从前两个最小的匹配开始，估计其他关键点在当前帧以及数据库中是否在该有向线段的相同一侧。我们使用向量积（cross product）来检测是否在有向线段的同一侧。

假设两个匹配点对为 $M_1 = [(x_1, y_1), (x'_1, y'_1)]$（图 4-1 中的红色点对）和 $M_2 = [(x_2, y_2), (x'_2, y'_2)]$（图 4-1 中的蓝色点对），其中每个匹配点对中的两个点坐标分别表示在帧图像和原目标图片中的像素坐标。我们要检验匹配点对 $M_3 = [(x_3, y_3), (x'_3, y'_3)]$（图 4-1 中的黄色点对）是否是一个错误匹配。 根据 M_1、M_2 我们可以分别得到在帧图像和原目标图片上的一条测试有向线段。

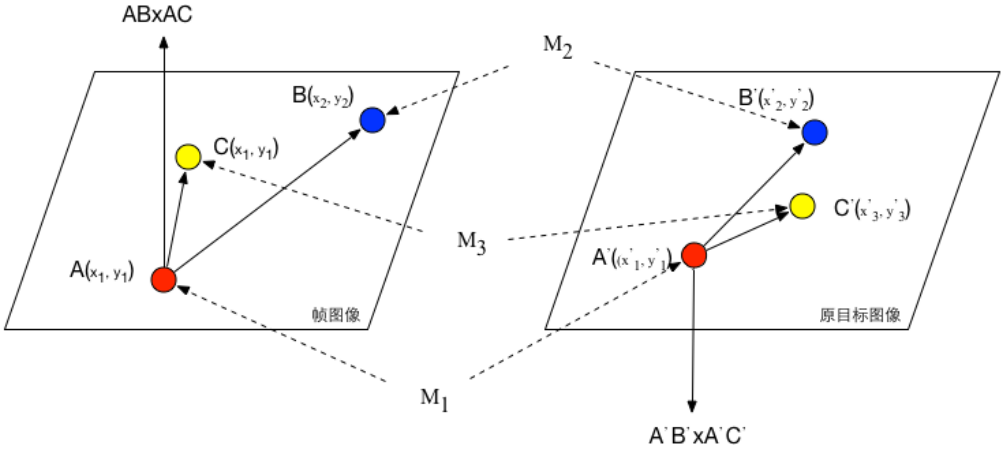

图 4-1 几何检测

由图 4-1 可以看出，帧图像中的向量积结果和原目标图像中向量积的结果相反，即匹配点对 M_3 在帧图像和原目标图像中的点不在测试有向线段的相同一侧。

如果有超过总样本大小的一半数量的点不在同一侧，则认为是一次无效检测。否则，不在同一侧的关键点将会被认为异常点。一般检验的直线数目不超过 20。

4.2.4 单应性检验

最后，我们使用随机抽样一致性算法（RANSAC）计算变换矩阵（详见第五章），并排除异常点。使用 RANSAC 算法，我们可以从包含异常点的特征匹配中，通过迭代的方式估计单应性矩阵的参数。RANSAC 算法的主要流程是：

(1) 从特征匹配样本中随机抽取一个 4 个匹配点对，作为初始样本；
(2) 根据这 4 个匹配点对计算变换矩阵；
(3) 根据样本集、变换矩阵和误差度量函数计算满足当前变换矩阵的一致集，并返回一致集中的点对个数；
(4) 根据当前一致集中点对的个数判断是否是最大一致集，若是则更新当前最大一致集；
(5) 更新当前错误率 p，若 p 大于允许的最小错误率则重复(1)至(4)进行迭代，直到当前错误率小于允许的最小错误率。

4.3 目标数据的预处理

4.3.1 特征检测

为了保证系统的健壮性，PC 端子系统需要提前处理目标图片，检测不同尺度的目标图片中的关键点。我们首先把原图片经过一定的高斯模糊作为初始图片，然后以此图片构造图片金字塔。金字塔的每一层比上一层长宽缩小根号 2 倍。金字塔的最高层和最低层定义了可识别的目标大小范围。通常金字塔层数为 7-8 层，在每一层图片上使用 FAST 算法检测角点，并计算其描述子。图 4-2 显示了测试图片对应的图像金字塔。

图 4-2 图像金字塔

经过图像金字塔的角点检测后，上述测试图片最终得到的所有关键点如图 4-3 所示。

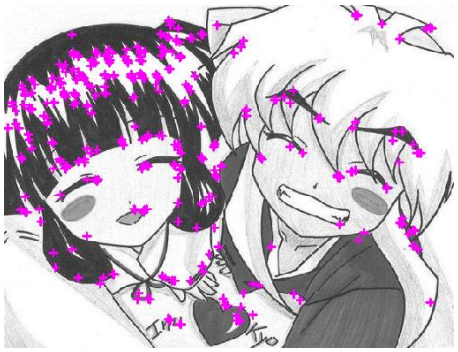

图 4-3 检测的关键点

4.3.2 数据规格

PC 端子系统共需维护 3 个数据：包含有目标图片所有相关信息的二进制文件 target.dat，包含有 PC 端显示所需的 JSON 文件 target.json，以及包含有所有目标图片的文件夹 Targets。

二进制文件将会被传递给移动端子系统，其中包含了所有目标图片的名称、长宽值、关键点、特征描述子以及图片本身的信息。

JSON 文件包含了所有目标图片的名称和长宽值。其格式如图 4-4 所示：

```
1  {
2      "targets": [
3          {
4              "height": 444,
5              "name": "Illustration0.jpg",
6              "width": 580
7          },
8          {
9              "height": 344,
10             "name": "Illustration1.jpg",
11             "width": 600
12         }
13     ]
14 }
```

图 4-4 JSON 文件格式

4.4 本章小结

本章详细阐述了系统在移动终端的物体识别技术，介绍了更改后的 PhonySIFT 算法，对其中的关键技术做出了详细解释。同时，我们还阐述了如何进行异常点处理，以尽可能多地过滤错误的特征匹配。之后，我们指出了 PC 端子系统在预处理时使用的 PhonySIFT 与移动端实时检测时使用的算法的区别，并对 PC 端预处理的关键步骤进行了详细说明，给出了数据库文件所需的数据规格等内容。

第五章 移动终端的姿态估计

帧图像是通过一个数码摄像头捕捉显示世界中的某个场景（如平放了一张目标图像的书桌）得到的。因此，一张帧图像实际上就是由一个 3D 场景投影到一个 2D 平面上。在本章中，我们将讨论如何得到帧图像和原目标图像之间的投影关系，以用于显示三维物体。

5.1 姿态估计

在得到帧图像中的特征点和原始目标图片中的特征点之间的匹配关系后，就可以生成一个 2D 到 3D 的顶点对应关系，计算单应性矩阵，以此来估测摄像机姿态。

正如本章开头讲到的，我们首先考虑一个 3D 顶点和它在帧图像之间的投影关系。通常，我们知道这种投影关系可以使用一个 3x4 的矩阵来表示。投影矩阵也可以成为变换矩阵，我们可以这样理解，当把原目标图片整体覆盖在 3D 坐标系中的摄像机屏幕上时，通过变换矩阵的平移和旋转后，即可得到在帧图像中投影的图像。如图 5-1 所示。

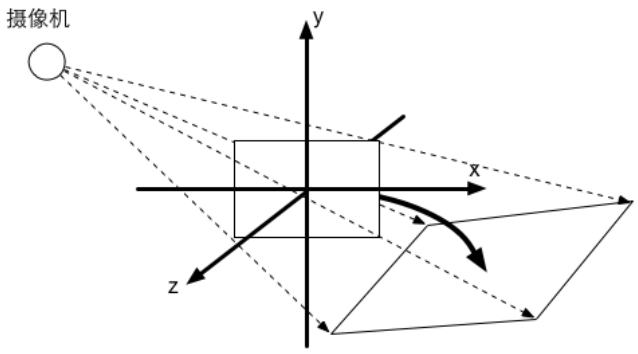

图 5-1 3D 顶点和帧图像之间的投影关系

我们使用以下摄像机模型进行计算。在这个模型中，我们使用一个投影变换来将 3D 顶点坐标（原始目标图片中的每一个特征点对应一个 3D 顶点坐标（x_i, y_i, 0））投影到帧图片上，从而形成一个 3D 场景视图。

$$m = A\left[R \mid t\right]M \tag{5-1}$$

或

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{5-2}$$

其中各变量含义如下：

(1) (u, v)是帧图像中的特征点的像素 2D 坐标；

(2) (X, Y, Z)是对应的在世界坐标系中的 3D 坐标；

(3) 矩阵 A 是摄像机内置矩阵。由于移动终端的摄像机一般为针孔摄像机，矩阵 A 则是摄像机的一些内置参数。(c_x, c_y)表示主点（principal point），通常是图像的中心；f_x、f_y 是以像素为单位的焦距大小；

(4) 矩阵[R|t]表示 2D 到 3D 的旋转和平移矩阵，即变换矩阵。

因此，当已知帧图像中的特征点的像素坐标和原始目标图片中的特征点的世界坐标后，就可以求得变换矩阵[R|t]。

通过变换矩阵[R|t]，我们可以把任意顶点(X, Y, Z)转换到摄像机坐标系中，这种转换可以通过下面的公式 5-3 来表示：

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t \tag{5-3}$$

例如，当我们取定原目标图片的四个顶点，由于在图像平面坐标系中，图像左下角为坐标原点，所以其像素 2D 坐标分别为(0, 0)、(width, 0)、(width, height)和(0, height)。而在摄像机坐标系中，坐标原点为图像中心，因此分别对应了摄像机坐标系中的 3D 顶点(-width/2, -height/2)、(width/2, -height/2)、(width/2, height/2)和(-width/2, height/2)。我们将四个顶点的 3D 坐标通过变换矩阵转换，即可得到它们在摄像机坐标系中真实的 3D 坐标。图 5-2 显示了这样的变换结果。

图 5-2 通过变换矩阵得到 4 个顶点的 3D 坐标

5.2 三维图像渲染

系统使用了 OpenGL ES 进行渲染。首先，根据 5.1 节中的摄像机内置矩阵 A 设置 GL 的世界坐标系中摄像机的视角和长宽比。它们之间的关系如下所示：

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} w/(2 \times \tan(fovX)) & 0 & w/2 \\ 0 & h/(2 \times \tan(fovY)) & h/2 \\ 0 & 0 & 1 \end{bmatrix} \tag{5-4}$$

其中 w 和 h 分别表示摄像机屏幕的长和宽；fovX 和 fovY 分别表示摄像机横纵视角大小。

在移动端子系统启动时，系统为 OpenGL 设置好相关参数，并为每一个预处理过的目标图片绑定一组具有随机颜色的 4 个立方体组成的 3D 图形。一旦在帧图像中识别到目标并得到变换矩阵后，就会在屏幕相应位置绘制图形。图 5-3 显示了在识别到不同的目标图片时，系统渲染了不同颜色的立方体组。图 5-4 则展示了不同姿态下的 3D 图像渲染效果。

图 5-3 不同目标图片识别效果

图 5-4 姿态估计效果。从左往右的姿态估计效果分别为
正面、右面、后面、左面。

5.3 本章小结

本章详细解释了移动终端姿态估计所用的技术和原理，阐述了如何根据帧图像和原始目标图片之间的匹配关系计算变换矩阵，以及 3D 物体渲染的流程和原理，并给出了不同姿态估计下的 3D 物体渲染结果。

第六章 实验和评估

6.1 追踪的数据库大小

预处理数据库的大小往往对预加载和检测时间影响较大。预加载指的是移动端在启动时需要读取 PC 端子系统生成的二进制文件，将所有可以识别的目标信息注册到移动端子系统，这些信息包括了可识别目标的关键点和特征描述等信息。检测指的是在未追踪状态时，移动端子系统需要扫描上一步中所有注册过的目标图片并进行匹配，以尝试找到一张最佳匹配目标。因此，数据库的大小将直接影响到这两种时间。

为此，我们以数据库大小为变量，取范围从 5 至 80，且数据库中每张图片大小均为 580*450，每张图像的特征点个数在 500 左右，对预加载和检测时间两个因素进行了评估。

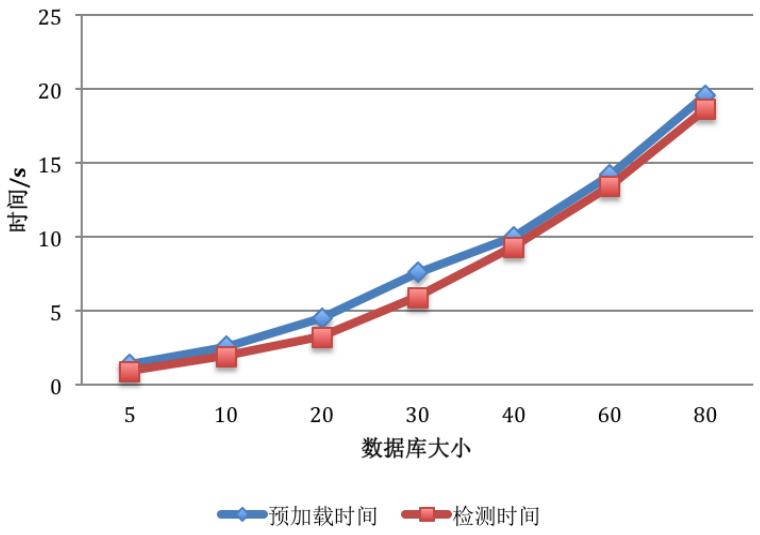

图 6-1 数据库大小对性能的影响分析结果

由图 6-1 可以看出来，随着数据库大小不断增大，时间的消耗基本符合线性关系。而合理可识别目标的数据库范围在 20 张左右，即预加载时间和检测时间不超过 5 秒。

预加载时间主要用于读取所有预处理目标图像的特征描述向量，而检测时间的时间消耗主要在于匹配当前帧图像特征描述向量和与原目标图像中的特征描述向量，因此两者的时间消耗相差不大，并且均与目标图像中包含的特征点个数成线性关系。

6.2 追踪的目标种类

为了评估对不同种类的追踪目标的追踪效果，我们使用 PhonySIFT 算法检测了七种类型的追踪目标，分别是插图、儿童读物、广告、电影海报、杂志封面、人物照片以及 Logo，如图 6-2 所示。每个种类的测试图片尽量概括了该种类在现实世界中的不同对象，每个种类共包含 20 张测试图片。

图 6-2 七种追踪的目标种类(a)-(g)：插图，儿童读物，
广告，电影海报，杂志封面，人物照片，Logo

我们对每种类型的追踪目标分别进行旋转测试、倾斜测试、遮挡测试，分析测试过程中成功追踪目标的比例。

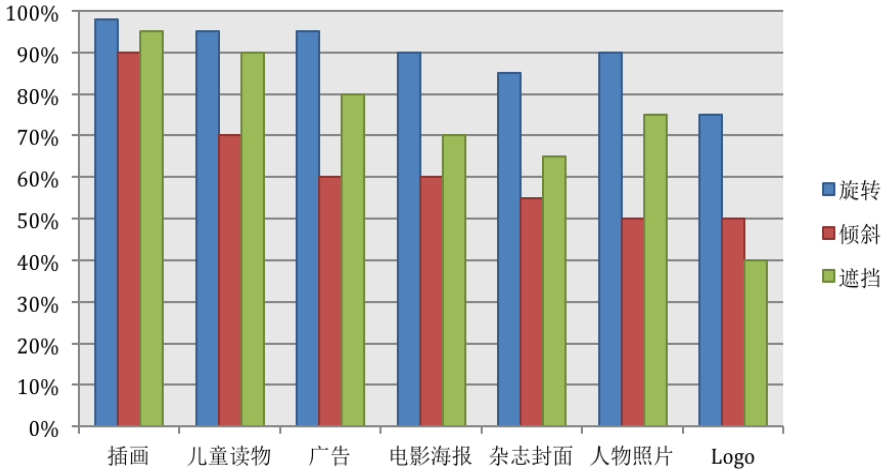

图 6-3 追踪的目标种类测试结果

由图 6-3 可以看出，从综合效果来讲，插画类型的追踪目标在各方面都具有较好的追踪效果，而测试的 Logo 类型的图像表现较差。对于旋转测试，绝大部分测试图像都具有良好的追踪效果，除了 Logo 类型的目标图像，识别率都在 80%以上。倾斜测试的情况普遍低于其他两种测试。而遮挡测试对于绝大多数测试目标来说正确率都在 70%以上。

我们分析产生这样结果的原因。插画由于本身色彩丰富，通常具有明显的角点信息和强对比度，因此具有较多特征明显的关键点且关键点分布均匀，追踪效果好。而 Logo 类型相对来说纹理简单，角点不明显，细节部分较少，且很多 Logo 包含了较多的文字以及重复性图案，因此生成的关键点少，特征描述的质量较差，追踪效果也最差。儿童读物包含了少量文字，关键点分布比较集中，但总体特征比较明显。广告、电影海报和杂志封面中，文字图案较多，因此在进行倾斜或遮挡时，文字图案的特征波动较大，影响了追踪效果。对于杂志封面和人物照片，这类目标通常只由少量人物或图案组成，并且对比度不明显，以及很多模糊的图片，使得无法找到更多的关键点。

通过对不同类型的追踪目标的追踪效果的分析，我们总结出好的追踪目标所具有图像特性有如下几点：

(1) 具有明显角点，避免圆形图案。由于 PhoneySIFT 使用 FAST 角点检测算法定位关键点，对于圆形是无法检测到有效关键点的。

(2) 避免重复图案。重复的图案会使生成到的特征描述缺少独特性，检测器区分分别它们之间的差别，从而影响检测结果。

(3) 关键点分布均匀，尽量覆盖整个图像。如果关键点集中在某一个区域，则图像会对遮挡、目标丢失等情况具有较差的健壮性。

(4) 具有较强的局部反差，避免模糊的细节，轮廓分明。FAST 角点检测算法是依靠检测关键点及其周围像素之间的像素差来判定的，因此模糊、反差较小的轮廓将无

法检测到足够多的特征点。

6.3 新增的异常点处理

在第四章中，我们提到了五种异常点处理方法，分别为比率检验、对称检验、描述子方向检验、几何检验和单应性检验。其中对称检验和比率检验是我们在原实现方法基础上新增的两种异常点处理步骤。为了评估这两种新增的异常点处理对追踪效果的影响，我们选取了四种特殊情况，即模糊、旋转、倾斜和远距离下的图像，对比它们在添加这两种新的异常点处理前后的匹配效果。图 6-4 给出了添加两种异常处理方法前后计算单应性矩阵的对比结果。

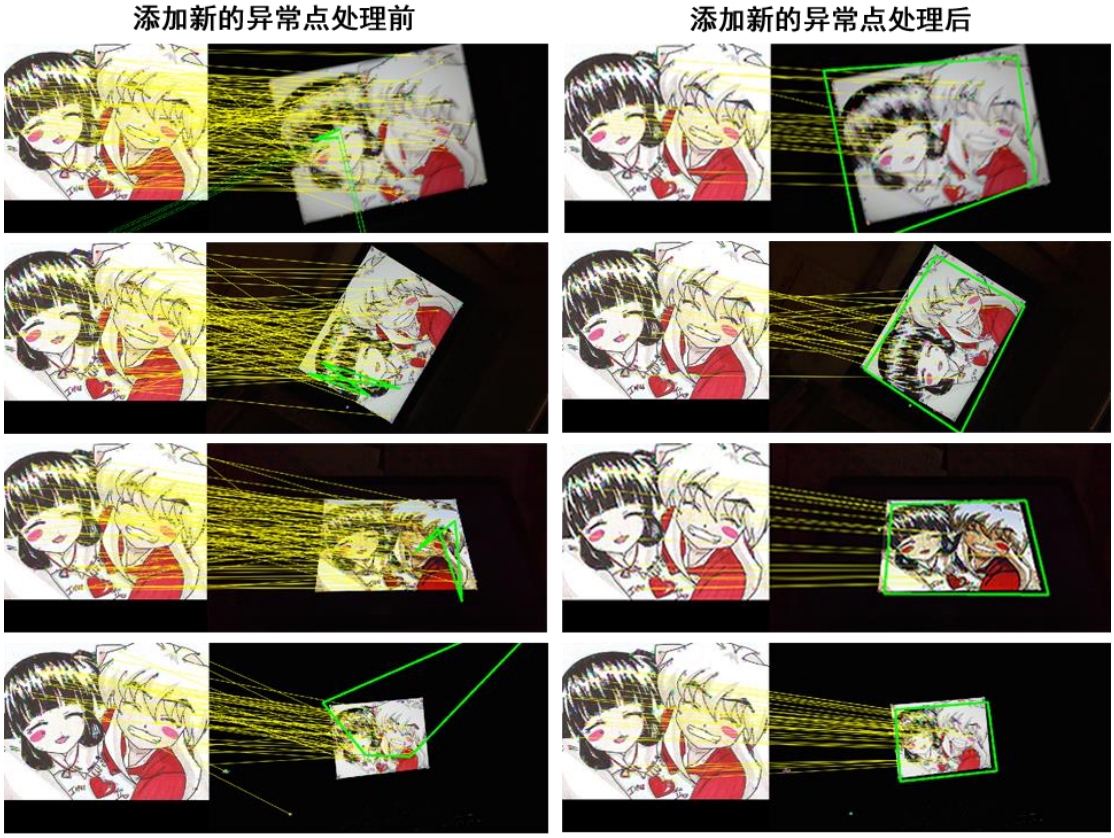

图 6-4 异常点处理前后对比

图 6-4 中第一行对应了模糊匹配的结果，第二行对应旋转匹配结果，第三行对应倾斜匹配结果，最后一行对应远距离匹配的结果。从图中可以看出，在添加比率检验和对称检验前，虽然产生的匹配点对数目较多，但由于包含了一定数目的错误匹配，对变换矩阵的计算结果产生了很大的影响。添加了比率检验和对称检验的异常点处理方法后，匹配点对的数目大幅度下降，而且剩余点对的质量也有很大提升，使得系统在特殊情况下的健壮性有了大幅度提高。

综上，我们新增的两种异常点处理方法，即比率检验和对称检验，进一步减少了异常点的数量，不仅较大程度上提高了系统的可靠性，而且由于比率检验和对称检验一开始就过滤了大量匹配点对，减轻了后面三种异常点处理方法的负担，因此提高了系统匹配的性能。

6.4 追踪的健壮性

为了评估在不同条件下系统的健壮性，我们进行了追踪的健壮性测试。

基于插画目标集，我们一共进行了 5 种不同的测试序列来分别评估系统在丢失目标、倾斜、远距离、遮挡和旋转方面的健壮性。序列 1 测试了目标图片逐渐在摄像机视野范围内丢失的情况；序列 2 测试了目标图片所在平面的正交平面与摄像机视角倾斜角度逐渐扩大的情况；序列 3 测试了摄像机逐渐远离目标图片的情况；序列 4 测试了目标图片和摄像机之间的遮挡物范围逐渐扩大的情况；最后，序列 5 测试了目标图片在不同旋转角度下的情况。

图 6-5 中的每一列显示了序列 1 至序列 4 的测试情况，而第一行显示了好的识别效果，第二行则表明了系统开始丢失追踪目标时的情况，最后一行则显示了系统完全丢失目标的情况。

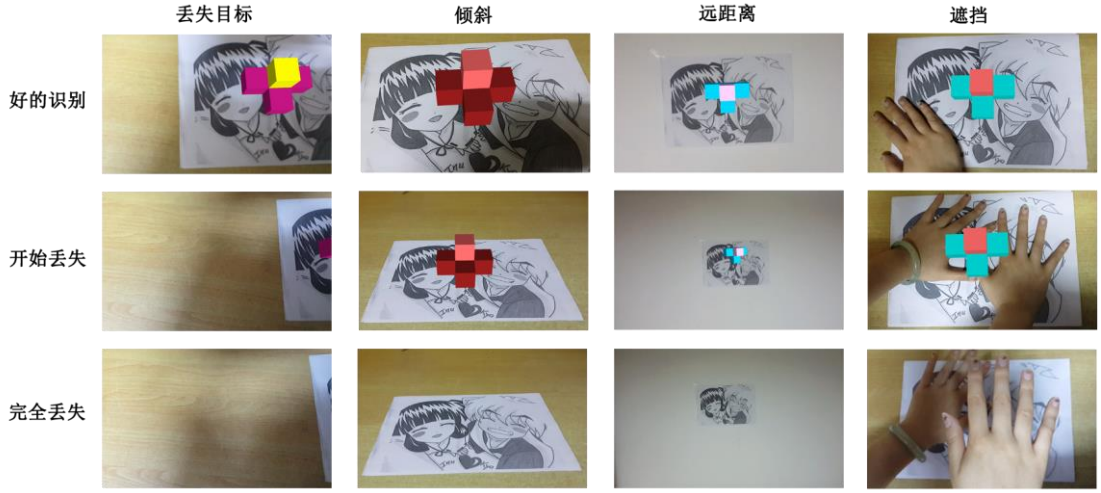

图 6-5 追踪的临界测试结果

根据测试结果可以看出，当目标图片在摄像机视野范围内的部分少于整体部分的 20%时，系统仍可以追踪到目标；当目标图片所在平面的正交平面与摄像机视角倾斜角度不超过 45°时，系统可以追踪到目标；远离目标时，当目标整体所在区域面积超过摄像机视野范围的 10%时，系统可以追踪到目标；目标图片和摄像机之间的遮挡物范围超过 60%时，系统仍可以追踪到目标。

图 6-6 显示了当旋转目标图片时系统的追踪效果。可以看出，在不同的旋转角度下，系统都可以取得较好的追踪效果。

图 6-6 旋转测试结果。从左到右旋转角度大约为 30°、90°、210°、270°

6.5 性能

我们对 PhonySIFT 的性能进行了更详细的评估，以用于分析可改进和提高性能的方向和潜力。

我们对算法的三个主要步骤进行了评估，分别是关键点检测、特征描述以及匹配和异常点移除，其中关键点检测又包括了 FAST 角点检测和计算关键点方向两个主要步骤。如图 6-7 所示，我们可以看出关键点检测和特征描述占据了整个算法时间的 90%以上，而计算关键点方向和计算特征描述子两个步骤，是影响 PhonySIFT 性能的最主要因素。这种结果意

味着要想进一步提高 PhonySIFT 的性能，我们需要着重从这两方面入手，尝试减少计算的复杂度，或使用更高效的数据结构等。

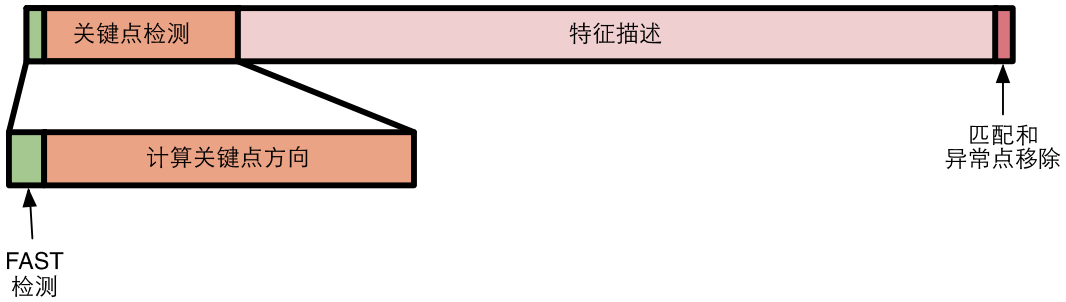

图 6-7 PhonySIFT 详细的性能分析

为了测试 PhonySIFT 较原生 SIFT 的性能提升情况，我们对两种算法进行了性能测试，分别测试了两种算法在关键点检测、特征描述、匹配和特征点移除三个方面的性能差异，并给出了总时间的比较结果。

我们使用测试机配置如表 6-1 所示。

表 6-1 测试机配置

机型	三星 Galaxy S4
CPU	1.9GHz，四核，高通 Adreno320
内存	16GB ROM，2GB RAM

由于算法的相关时间和追踪图片的范围、大小以及目标图片的关键点数量等因素有关，因此我们给出的范围是 90%以上测试图片所覆盖的性能结果。其中，原生 SIFT 计算描述时使用子区域大小为 4*4 像素，计算每个像素对应的 8 个量化后的梯度值。

表 6-2 原生 SIFT 和 PhonySIFT 的性能比较（单位：秒）

算法	关键点检测	特征描述	匹配和异常点移除	总时间
原生 SIFT	11.0 – 15.0	19.0 – 25.0	1.0 – 1.5	31.0 – 41.5
PhonySIFT	0.1 – 0.3	0.2 – 0.7	0.1 – 0.3	0.4 – 1.3

由表 6-2 可以看出，经过修改后的 PhonySIFT 比原生的 SIFT 在各方面的性能提高非常显著。其中，因为使用了 FAST 算子代替原 SIFT 算法中的高斯差分金字塔来定位关键点的位置，因此在关键点检测这一步中取得了非常明显的速度提升结果，提升倍数在 50 至 100 多倍。在特征描述这一步中，由于我们减小了计算描述子时子区域的大小，并将 8 个量化后的梯度值缩减为 4 个，大幅度减小了特征描述向量的维数，使得性能提升了 30 至 90 倍。由于特征描述向量的减少，也影响了匹配和异常点移除的性能。

总体来说，经过改进后的 PhonySIFT 算法的计算复杂度更好地适应了移动终端的特性，在移动终端的实时物体追踪上取得了较好的应用效果。

6.6 应用

为了验证系统在现实生活中的应用，我们使用本系统来识别和追踪现实生活中的几种不

同商品，包括零食、口香糖、意面、饮料和报纸等。如图 6-8 所示。

图 6-8 演示应用使用的现实物体，从左到右依次是
报纸插图、零食、饮料、意面、口香糖

该应用可以识别现实世界中的特定商品，在屏幕上显示出该商品的名称、价格、介绍等信息。通过该应用，用户可以方便的得到商品的相关信息。

这些商品包含了多种不同材质和特性，例如饮料的瓶身为圆柱形，零食为塑料包装，报纸的印刷效果，口香糖商品面积小等。由此说明本系统在现实世界同样可以取得较好的识别和追踪效果。

我们提前拍摄了这些物体的照片，经过 PC 子系统的预处理后导出二进制文件，并传递给移动端子系统。在移动端子系统，我们将商品对应的相关信息绑定到预处理的追踪目标上，以便在识别后在屏幕上显示相应信息。

图 6-9 演示应用显示的商品信息

通过该应用，我们也演示了使用该系统的一般步骤：首先，将需要在移动端识别的目标图片在 PC 端子系统进行处理，PC 端子系统将识别出来的特征描述等信息导出成二进制文

件；然后，我们将导出的二进制文件移动至移动端子系统目录下，以便其在启动时可以读取文件进行预加载；我们还可以在移动端子系统定制识别到相应物体时的动作，例如显示三维物体或者显示相关文字信息等；最后，我们把生成的应用文件安装到测试手机上即可。

6.7 本章小结

本章对系统进行了各方面的评估。通过评估数据库大小对性能的影响，我们得出了在合理预加载和检测时间内的数据库大小。我们选择了七种常见的目标种类，评估它们在三种常见情况，即旋转、倾斜和遮挡下的正确追踪的比率，以说明我们的系统对于哪些类别具有更好的追踪效果。同时，我们还给出了好的追踪目标所具有的图像特性。接着，我们通过评估在目标丢失、倾斜、遮挡、远距离等情况下的追踪效果，给出了系统追踪的在各种条件下的健壮性，实验证明系统在大多数情况下都能较好的识别和追踪。我们还进行了 PhonySIFT 的详细性能分析，并和原生 SIFT 的性能对比。经过原生 SIFT 的比较，我们发现 PhonySIFT 的时间性能提高非常显著，达到了 40 倍以上。最后，我们实现了一个可以应用于现实生活的演示应用，用户可以使用该应用识别特定商品，得到该商品的相关信息。

第七章 总结和展望

7.1 本文工作总结

本文结合了目前比较成熟的两种特征提取算法，SIFT 和 FAST，实现了一个可用于移动终端的实时物体识别和追踪系统。原始的 SIFT 算法虽然可以取得较好的识别和匹配效果，但计算量大，不适合用于移动终端的实时物体追踪系统。而传统的 FAST 算法虽然特征提取速度快，但特征描述信息过少，无法提供可靠的匹配的效果。

为了避免以上算法的弊端，本文设计并实现了一个包含 PC 端子系统和移动端子系统的实时物体追踪系统，结合了上述两种特征提取算法，实现了一个使用 FAST 算子修改后的 PhonySIFT 算法。本系统可以经过预处理带有自然特征的目标图片，在移动终端实时识别和追踪目标，并在相应位置显示三维物体。其中，PC 端子系统负责管理所有可追踪的目标图片，并导出二进制数据库文件作为输入传递给移动端子系统；移动端子系统负责真正的目标识别和追踪。

然后，本文针对系统在各方面的表现进行了评估和测试。我们发现修改后的 PhonySIFT 算法在丢失目标、倾斜、遮挡、远距离、旋转等方面都取得了较好的健壮性。在性能方面，PhonySIFT 算法相比与原生 SIFT 算法性能提高了 40 倍以上，在移动终端取得了算法复杂度和可靠性上的有效平衡。

通过演示应用，我们展示了一个可以在现实世界中识别和追踪商品的移动应用，用户可以通过该应用得到相关的商品信息等。

本文的研究结果是在移动端进行增强显示的初步尝试，其结果表明移动端实时物体追踪技术的巨大潜力。

7.2 未来工作

尽管 PhonySIFT 算法在性能上有了很大提高，但面对特征点数目较大的目标时其性能略显不足。这是由于本系统是基于识别的追踪系统，即在识别目标后，继续使用 PhonySIFT 算法进行匹配，时间耗费较大。除此之外，在一些极端情况，例如目标图片倾斜角度较大、图片模糊程度较高时，追踪效果急剧下降。

针对以上两点，在接下来的研究中，可以从两方面考虑进一步优化系统：

(1) 使用基于模板匹配的追踪系统，即根据已知的匹配信息预测下一帧特征点的所在位置，进行小范围的面片匹配，从而缩小匹配范围，提高性能。

(2) 为了进一步提高匹配可靠性，我们可以考虑增加其他辅助实时识别和追踪算法，从其他角度得到不同的匹配信息，从而增强匹配结果的可靠性。例如，Ferns 算法可以将物体识别视为分类算法，将运算量较大的部分转移到分类器训练过程中，对局部图像进行快速匹配。这不仅可以提高识别时间，也可以用于基于匹配的追踪系统中。

在实际应用方面，本系统具有非常广泛的应用前景。下一步的工作可以将本系统应用到广告、家电购物演示、商品信息查询等方面。例如，用户可以通过识别商家的 LOGO 得到该商家最新的活动和优惠信息；通过识别杂志上面的家电图片在家中模拟放置相应家电时的效果；还可以通过识别商品包装显示商品信息等。

在经过技术和商业应用上的优化和加强后，本系统将更接近可用于实际应用的增强现实系统。

参考文献

[1] Hagbi N, Grasset R, Bergig O, et al. In-place sketching for content authoring in augmented reality games[C]//Virtual Reality Conference (VR), 2010 IEEE. IEEE, 2010: 91-94.

[2] Kato H, Billinghurst M. Marker tracking and hmd calibration for a video-based augmented reality conferencing system[C]//Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on. IEEE, 1999: 85-94.

[3] Fiala M. ARTag, a fiducial marker system using digital techniques[C]//Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005, 2: 590-596.

[4] Schmalstieg D, Wagner D. Experiences with handheld augmented reality[C]//Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on. IEEE, 2007: 3-18.

[5] Rohs M, Schöning J, Krüger A, et al. Towards real-time markerless tracking of magic lenses on paper maps[C]//Adjunct Proceedings of the 5th Intl. Conference on Pervasive Computing (Pervasive), Late Breaking Results. 2007: 69-72.

[6] Hagbi N, Bergig O, El-Sana J, et al. Shape recognition and pose estimation for mobile augmented reality[J]. Visualization and Computer Graphics, IEEE Transactions on, 2011, 17(10): 1369-1379.

[7] Pirchheim C, Reitmayr G. Homography-based planar mapping and tracking for mobile phones[C]//Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on. IEEE, 2011: 27-36.

[8] Kim K, Lepetit V, Woo W. Scalable real-time planar targets tracking for digilog books[J]. The Visual Computer, 2010, 26(6-8): 1145-1154.

[9] Wagner D, Reitmayr G, Mulloni A, et al. Pose tracking from natural features on mobile phones[C]//Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 2008: 125-134.

[10] Wagner D, Reitmayr G, Mulloni A, et al. Real-time detection and tracking for augmented reality on mobile phones[J]. Visualization and Computer Graphics, IEEE Transactions on, 2010, 16(3): 355-368.

[11] Lowe D G. Object recognition from local scale-invariant features[C]//Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Ieee, 1999, 2: 1150-1157.

[12] Rosten E, Drummond T. Machine learning for high-speed corner detection[M]//Computer Vision–ECCV 2006. Springer Berlin Heidelberg, 2006: 430-443.

谢辞

作为本科学业的毕业设计，本文凝结了我最后一年的很多精力和心血，而在这个过程中，也受到了很多老师和同学的无私帮助。

首先，我要感谢我的指导老师。杨旭波老师。从项目的立意、方向和最终定稿方面都给了我非常重要的意见和建议。在遇到困难时，杨旭波老师的鼓舞和激励使我坚持走了下来。除此之外，杨老师对学术的专注和严谨态度也给我留下了深刻印象，他不仅教会了我学术上的研究方法，也让我明白做学术要耐得住寂寞、狠下功夫才能取得成果的道理。在今后的学习生涯中，我会一直保持这样的态度，希望在专业上取得更好的成绩。

其次，我要感谢实验室的何贞毅、邓念晨、黄磊、曹阳等学姐学长的帮助，他们在进行自己的毕业设计时，仍给了我很多建议。还要感谢同样在实验室进行毕业设计的林之雨同学，我们之间的交流给我了很多灵感。

同时，也感谢四年来给了我很多帮助的任锐老师，是在任老师的帮助下，我开始更加热爱软件工程这个专业。还要感谢肖双九老师，在仅有的几次由肖老师教授的课程中，我看到了女性在这个领域的光芒，让我摆脱了“女生学不好计算机”这个思想禁锢，更让我坚定了学习的动力和信心。也感谢四年来给予我帮助的其他所有学院老师。

最后，感谢四年来我的室友们，她们的幽默、乐观和友善深深感染着我，我们一起度过了愉快的大学四年生活。以及我所爱和爱我的家人们，是他们的关心和照顾才让我一步步离梦想更近，无论何时他们都是我最坚强的后盾，而他们也是我前进的最大动力。感谢所有一路上相遇并帮助我的人们，正是你们的相伴让我在成长的道路上走得更远。

REAL-TIME OBJECT TRACKING AND POSE ESTIMATION ON MOBILE PHONES

Object recognition and pose estimation are the two main topics in the Computer Graphics and Augmented Reality (AR). This paper analyzed the existing object recognition and tracking technology based on natural features, and implemented a real-time object tracking system on mobile phones. The object recognition on mobile phones is achieved through a modified version of the SIFT algorithm using FAST. Pose estimation can be obtained from the transformation matrix matching between the identified targets and the frame. In addition, we implemented a PC client tool for management of all preprocessed target pictures. It was observed in experiments that the modified algorithm is 40 times faster than the original SIFT. We argue that the system performs well on mobile phones and discuss the direction of future works as well.

The Augmented Reality (AR) is currently focused on object recognition and pose estimation, and displays the enhanced virtual objects in the real world. These techniques can be used for advertising, online shopping, games, children's education and so on.

Object recognition uses approaches in feature descriptors to recognize some targets in pictures. And pose estimation determines the six degrees of freedom (6DOF) in screen coordinates and then renders some 3D objects according to it. But tracking from natural features is a complex problem and demands high computational power, especially for mobile phones.

Mobile phones are attractive targets for AR due to their lower price and portable size, but have even more limited performance than the Tablet PCs. Phones are embedded systems with severe limitations in both the computational facilities and memory bandwidth. Therefore, to trade off the effect of real-time object tracking and performance has become a very popular research topic.

In this paper, we present a natural feature tracking system capable of tracking full 6DOF at real-time using the built-in camera of the phone. We achieved this by examining two approaches in feature descriptors, namely SIFT and FAST. The original SIFT is known to be a strong, but computationally expensive feature descriptor. FAST corner detection is fast, but the generated information of key points is poor.

The system is divided into two subsystems: PC subsystem and mobile subsystem. The output of PC subsystem will become an input of mobile subsystem. The mobile subsystem is responsible for reading frame images from the camera, recognizing and tracking objects in real-time.

We call the modified technique PhonySIFT, which combines SIFT and FAST. The hybrid tracking system is shown to be robust and faster. We back up our claims by a detailed evaluation of the tracker’s properties and limitations.

Traditional SIFT algorithm uses Difference of Gaussians pyramid (DoGs) to detect key points. However, this method is computationally intensive and not suitable for real-time tracking system on mobile phones. Therefore, we use FAST corner detector to replace the original DoGs, and can also achieve good reliability. Since FAST does not estimate a feature’s scale, we

reintroduce a picture of the pyramid to describe features over different scales.

Most SIFT use 4 ×4 sub-regions with 8 gradient bins (128 elements) to calculate descriptors for each key point. For performance and memory considerations, we use only 3 × 3 sub-regions with 4 gradient bins each (36 elements). According to Lowe's papers, the reliability of this method is only up to 10% worse than the best variant with 128 elements.

We perform the matching in two directions, that is, for each point in the frame we find the two best matches in the target image, and then we do the same thing for the features of the target image, finding their two best matches in the frame. And then remove outliers.

Our outlier removal works in five steps. The first step is ratio test. We verify that the ratio of the distance of the best match over the distance of the second best match is not greater than a given threshold. The ratio test is applied to both of the match sets. The second step is symmetrical matching. From the two sets, we extract the matches that are in agreement with both sets. The third step uses the feature orientations. We correct all relative feature orientations to absolute rotation using the feature orientations in the database. We estimate a main orientation and use it to filter out all features that do not support this hypothesis. The forth step uses simple geometric tests. All features are sorted by their matching distance, and starting with the least distance features, we estimate lines between two of them and test all other features to lie on the same side of the line in both camera and object space. The final step uses homographies in an RANSAC fashion to remove outliers.

In order to ensure robustness, the system requires a feature database to be prepared beforehand. Data acquisition starts by blurring the target with Gaussian kernel, then builds an image pyramid, each level scaled down with a factor of $1/\sqrt{2}$ from the previous one. The largest and smallest pyramid levels define the range of scales that can be detected at runtime. The Pyramid layers usually 7-8 layers, using FAST corner detection algorithm on each layer to detect its features.

Pose estimation uses a so-called pinhole camera model. In this model, a scene view is formed by projecting 3D points into the image plane using a perspective transformation. With matching features between camera frames and original targets, it is possible to generate a correspondence from 2D points to 3D points. We use it to obtain the final homography and then render a 3D model in the corresponding position in the camera coordinate.

In order to evaluate the tracking results of different target types, we use PhonySIFT algorithm to detect seven types of tracking targets: illustration, children's books, advertisements, posters, magazine covers, pictures and logos. We performed rotate, tilt, and occlusion test for each type, and analysed the success ratio of the tracking process. From the results, illustrations obtained good track in all respects, while logos performed poor. For the rotation test, most of the test image has good tracking ability with above 80% recognition rate, except logos. The success ratios of the tilt tests are generally lower than the other two tests. By analyzing the tracking abilities of different target types, we concluded some good characteristics of the targets. First, the features in the image need to be sharp and non-circle. Second, try to improve the contrast of the image in general or choose an image with details that are more “edged”. Third, avoid repeated motifs (even if rotated and scaled). And last, the more balanced the distribution of the features in the image, the better the image can be detected and tracked.

We created five different test sequences to showcase some different practical situations: losing target, tilt, scale, occlusion, and rotation. According to the results of robustness tests, our

system is able to work well with these sequences. Finally, we evaluated the performance of our PhonySIFT and the original SIFT. The performance improvement of PhonySIFT is very significant in all aspects. This is mainly because of the use of the FAST corner detector instead of the original DoGs to locate features. It also depends on the reduced size of sub-region and gradient bins when creating descriptors. Overall, the computational intensity of PhonySIFT can fit the characteristics of mobile phones much better. We have achieved a good result of real-time object tracking on mobiles.

A major weakness of this approach is the rather limited tilt angle (40-50 degree) and image blur they can tolerate. For the two shortcomings, further optimization of the system can be considered in two ways: first, use tracking based on template matching to predict the location of the matching feature points the next frame. Second, in order to further improve the reliability of matches, we can consider adding other real-time identification and tracking algorithm to get a different perspective of matching information, thereby enhancing the reliability of the results match. For example, Ferns object recognition algorithm may be regarded as classification algorithm. We can transfer the larger part of the operation to the classifier training process to match partial image fast.

After the above optimization, the system can be much closer to the practical application of augmented reality systems.