

# 同濟大學

TONG JI UNIVERSITY

## 毕业设计(论文)

课 题 名 称      基于 CMOS 的视觉识别与三维重建应用研究

副 标 题      \_\_\_\_\_

学 院      机械与能源工程学院

专 业      机械设计制作及其自动化（中意班）

学 号      1352050

学 生 姓 名      蔡中天

指 导 教 师      朱传敏

日 期      2017 年 6 月 16 日

## 基于 CMOS 的视觉识别与三维重建应用研究

### 摘要

本研究的内容是基于 CMOS 的三维视觉技术和应用。

其中在理论研究部分重点研究了：CMOS 传感器的特点、图像预处理的各类方法（包括灰度图像、二值图像和图像平滑等理论）、类圆识别算法（主要是对霍夫变化法的学习与应用）和三维重建理论（包括线性光学摄像机模型、畸变非线性模型和不同标定方法）。在之后的实践部分中，运用所学习的专业知识，结合编程，开发出了一套基于 Matlab 的对于特定目标的三维重建系统，并通过精心设计的实验对该系统的准确性进行验证、得到数据结果并进行总结归纳和提出改进方案。

本课题是对三维机器视觉中的目标识别和三维重建基础理论的一次广泛学习和综合应用。读者可以通过本文所提供的理论算法总结和实践案例，对三维计算机视觉技术中的识别技术和三维重建技术的基本流程有一个清晰的认识，也为解决一些特定应用场景中的实际问题提供了一些新思路。

**关键词：**CMOS，计算机视觉，三维重建，目标识别

装  
订  
线

# Research into The Application of CMOS-Based Vision Recognition and 3-Dimensional Reconstruction

## ABSTRACT

The content of this study is CMOS-based 3D vision technology and applications.

The theory study part is focused on: the features of CMOS sensors, various methods of image preprocessing (including grayscale image, binary image and image smoothing theories), circle-similar recognition algorithm (mainly the approach of Hough Transform method), and three-dimensional reconstruction theory (including linear optical model, distortion nonlinear model and different calibration and matching methods). Following is the part of practical application, in which a 3D reconstruction system for specific targets is developed basing on Matlab, applying the previously studied theories and programming skills. Then the accuracy of the system is verified through experiments carefully designed and the results are summarized.

This topic is a comprehensive study and integrated application of the basic theories of target recognition and 3D reconstruction technologies in the field of 3D machine vision. Through the summary of theoretical algorithms and the practical case provided in this paper, the readers can have a clear understanding of the basic process of recognition and 3D reconstruction within 3D computer vision technology. It also provides some new ideas for solving some practical problems in some of the specific application scenarios.

**Key words:** CMOS, Computer vision, 3D reconstruction, Target Recognition

装  
订  
线

## 目 录

1 引言 .....	4
1.1 计算机视觉概述及目标识别和三维重建的意义 .....	4
1.2 CMOS 图像传感器 .....	4
1.2.1 CMOS 简介与原理阐述 .....	4
1.2.2 CMOS 与 CCD 的比较 .....	5
1.2.3 选用 CMOS 的原因 .....	7
1.3 视觉识别与三维重建的国内外研究概况 .....	7
1.4 三维重建的常见方法 .....	8
1.5 本课题来源及所做工作 .....	9
1.6 本文段落安排 .....	9
2 理论与算法研究 .....	10
2.1 图像预处理 .....	10
2.1.1 灰度图像转换 .....	10
2.1.2 二值图像 .....	11
2.1.3 图像平滑 .....	11
2.2 圆形物体识别与提取 .....	14
2.2.1 圆形度法 .....	14
2.2.2 霍夫变换法 .....	14
2.3 摄像机数学模型及三维坐标线性匹配 .....	16
2.4 畸变与非线性成像问题 .....	18
2.5 相机的内外参数与标定 .....	19
3 代码实现与工程实践 .....	21
3.1 实验环境介绍 .....	21
3.2 核心功能的代码实现 .....	23
3.3 软件封装 .....	26
4 测试结果与讨论 .....	30
4.1 测试方案介绍 .....	30
4.2 具体测试步骤与结果 .....	30
4.2.1 实验设计 .....	30
4.2.2 实验结果 .....	31
4.3 结果分析 .....	34
4.4 系统可优化性讨论和改进方案 .....	34
5 结论与展望 .....	36
5.1 课题结论 .....	36
5.2 研究前景展望 .....	36
6 附录 .....	37
6.1 Matlab 代码 .....	37
6.2 代码的简单实用指导 .....	44
参考文献 .....	46
辞 谢 .....	49

装  
订  
  
线

## 1 引言

### 1.1 计算机视觉概述及目标识别和三维重建的意义<sup>[1-2]</sup>

视觉是人类的重要感官之一。人类的视觉系统依靠眼睛获取客观世界的图像信息，经由大脑对于这些信息的分析处理，完成对客观世界中物体的识别和定位，并作出反应。计算机视觉即通过计算机技术来模拟人类的视觉系统。

识别和测量技术是机器视觉最基本的两个应用（一般统称为机器视觉检测技术），可以说在广泛的计算机视觉领域内基本上任何更复杂的发展和应用都是建立在上述两个基本应用之上的。计算机视觉相较于人类视觉拥有得天独厚的优势，例如机器可以在极端工作环境中（如高温、强酸碱性环境等）依然获取图像信息；又如通过使用不同的传感器和摄像机，计算机可以获取人的肉眼观察不到的信息，如红外线、超声波、微波等；再如在优化的算法和强大的计算力的支持下，计算机视觉在简单重复性工作中正确率和效率都远超人类，如在产品颜色分类或产品合格率检测方面的应用。因此，计算机视觉技术有着广泛的工业和商业应用前景。

当然，计算机视觉相较于完全达到人类视觉系统的水准还有着一定的差距，其主要体现在物体识别的准确率和位姿测定的误差上。这种差距最典型的体现就是日前在自动驾驶领域出现的事故：2016年5月美国佛罗里达州的一辆特斯拉Model S型电动轿车在无人驾驶模式下，由于其视觉识别系统未能探测到前方明亮天空下的白色集卡车而导致严重车祸，直接致轿车驾驶员身亡的道路安全事件<sup>[3]</sup>。因此，目前全世界计算机视觉领域的科学家、研究人员与科技企业都在尽力完善算法，都是为了从这两个方面提升计算机视觉性能。

基于二维图像的三维重建技术是计算机视觉和计算机图形学的一个重要分支。由于在工程技术上需要大量三维信息以对物体进行各种各样的分析，因此能够通过摄像机获得的二维图像转化为三维场景拥有极其广泛的实用价值。

三维重建工作的本质是从二维图像中恢复目标物体的三维几何信息并重建其三维轮廓和位置，重建内容一般包括形状、位置、姿态、运动等。对于实体对象的三维重建技术是一种常见的科学问题，且为多个领域的核心问题，例如制造业中的视觉仿真建模、工况及成品视觉检测、同步定位与地图构建（SLAM, Simultaneous localization and mapping）计算机动画、增强现实技术（AR, Augmented Reality）、医疗图像诊断、数字媒体等。例如，在医学上，若患者的病变信息可以在计算机上以3D形式呈现，将为医生做出准确的诊断提供极大的便利，因此具有重要的临床价值。

### 1.2 CMOS 图像传感器

#### 1.2.1 CMOS 简介与原理阐述<sup>[4]</sup>

CMOS（Complementary Metal-Oxide-Semiconductor，互补金属氧化物半导体）本身是一种集成电路的设计制程。其通过在硅晶片模板上制出 NMOS（n-type MOSFET）和 PMOS（p-type MOSFET）制成基本元件，由于 NMOS 和 PMOS 在物理特性上互补，因此称为 CMOS。一般的，该技术可用于制作微控制器、微处理器、静态随机存取存储器等数位逻辑电路。

CMOS 图像传感器的研究始于上世纪 60 年代末，但在当时由于受到工艺技术的限制，发展有限。直到 20 世纪 90 年代初，随着大规模集成电路设计技术和信号处理技术的提高，CMOS 图像传感器才日益受到重视<sup>[5]</sup>，成为固体图像传感器的研发热点。近几年来，随着集成电路设计技术和工艺水平的长足进步，CMOS 图像传感器的一些性能指标已接近甚至超过 CCD（Charge-coupled Device，感光耦合元件）图像传感器<sup>[6]</sup>。

CMOS 图像传感一般由像素阵列、行（列）选通逻辑、定时电路、震荡发生器、偏置发生器、模拟信号处理器（ASP）构成，一般也包括模数转换器(ADC)<sup>[7]</sup>，见图 1.1。

装  
订  
线

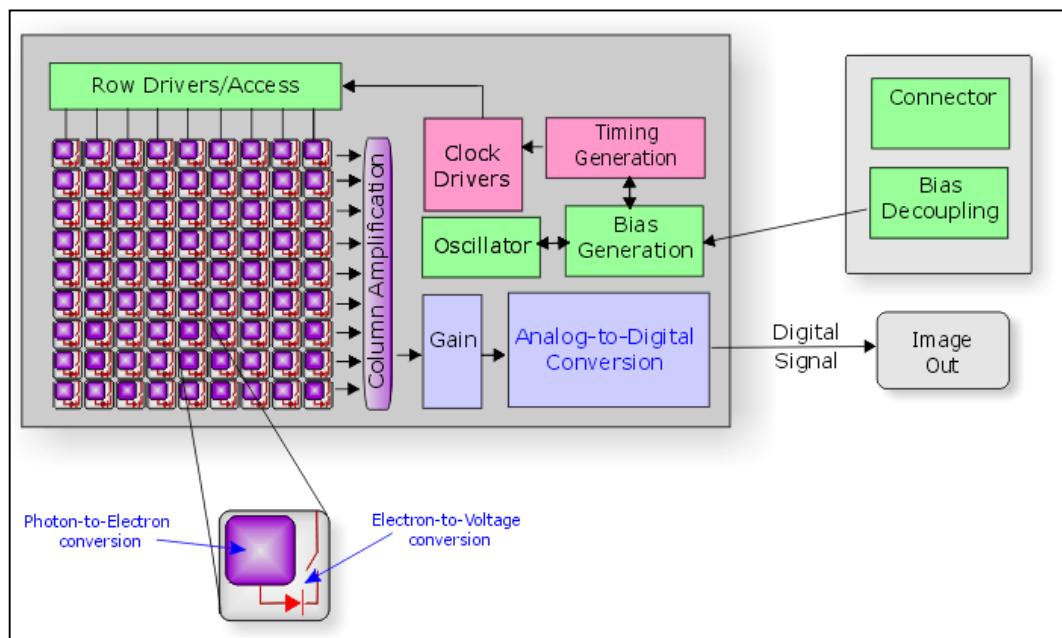


图 1.1 CMOS 图像传感器的一般结构

### 1.2.2 CMOS 与 CCD 的比较<sup>[8-10]</sup>

CMOS 和 CCD 都是基于 MOS 结构的光电转换效应进行光电转换的，它们的主要区别在于对光电转换后的电荷处理方式不同：CCD 通过水平和垂直的转移单元传载电荷，最终统一在负载上输出电压信号；而 CMOS 通过行列解码器直接输出电压信号。由于工作方式、结构和制造工艺的差别，CMOS 传感器一般比 CCD 体积小、功耗低、集成度高<sup>[11]</sup>。以下通过一张表格具体清晰地从技术和性能两方面比较这两种图像传感器：

表 1.1 CMOS 与 CCD 图像传感器技术细节和性能比较<sup>[12]</sup>

指标 / 项目	CCD	CMOS
像素输出信号	电荷包	电压
传感器输出信号	电压（模拟）	比特（数字）
相机输出信号	比特（数字）	比特（数字）
传感器复杂度	低	高
应用复杂度	高	低
研发成本	高	低
灵敏度	高	中
系统噪声	低	中
响应速度	中	中上
动态范围	大	中
响应均匀性	好	中下
信号处理速度	中上	高
随机窗口读取能力	有限	任意局部画面
电源电压	多电源高电压	单电源低电压

CMOS 与 CCD 图像传感器的研究几乎同时起步，但正如上文所说，由于受当时工艺水平的限制，早期的 CMOS 具有分辨率低、噪声大、光照灵、敏度低等缺点而没得到重视和发展。相反，CCD 却因大面积有效感光、均匀采集和小噪声等优势，主宰了图像传感器二至三十年的市场。近 10 年，由于集成电路设计技术和工艺水平提高，CMOS 过去的缺点都已基本克服，而其固有的诸如像素内放大、列并行结构、随机读取等独特的优点却是 CCD 无法比拟的。CMOS 传感器件到今天有了质的飞跃，在各个级别的图像质量需求中都有广泛应用。

简单来看，CCD 芯片输出模拟信号，则需要外部数模转换电路来实现信号的数字化；而 CMOS 直接输出数字信号，给后续信号的处理带来很多方便。从表 1.1 可见，CCD 采用移位寄存单元转移电荷包，因此如果其中一个移位寄存单元出现问题，就会引发一列电荷传输故障，因此 CCD 成品率更低。而采用 CMOS 时由于其控制、时间等逻辑电路高度集成，可大大简化整体系统的硬件设计。例如，对于一个基于 CMOS 的相机系统，由一个 CMOS 图像传感器加上实现逻辑功能的器件、存储器，再配上光学系统和相应的机械部分，就能实现一个完整的数字成像系统；而基于 CCD 的成像系统则相应要复杂一些，此处不做赘述。

可见，与 CCD 相比，CMOS 图像传感器具有集成度高、功耗低、成本低、速度快、响应范围宽、抗辐射性强等优点。CCD 在灵敏度、动态范围和成像质量等方面均优于 CMOS 图像传感

器，这也是在目前大部分专业领域（如电影视界）的行业标准中，固体摄像器件仍采用 CCD 核心图像传感器的原因。

### 1.2.3 选用 CMOS 的原因

之所以本文选用基于 CMOS 的摄像头作为图像采集器，是出于系统成本和其应用场景的普遍性考虑。

如上文所说，目前 CMOS 是民用相机领域的绝对主流。除了在专业影视行业标准中的摄像机仍采用昂贵的高端 CCD 芯片外，普通民众已经很难在日常使用到基于 CCD 的商用图像系统，不论是数码相机、数码摄像机还是网络摄像头。因此选用基于 CMOS 的网络摄像头，完全符合本课题的目的，即在控制成本的前提下设计出一套拥有广泛应用背景的简单物体识别和三维重建系统。

因此在基于 CMOS 的情况下，若能验证该套系统的可用性，则基本保证了它可运用于现在市场上大多数的摄像头上，使得其适用范围非常广泛且廉价。

### 1.3 视觉识别与三维重建的国内外研究概况

机器人视觉的发展离不开计算机视觉的发展。1965 年，Robots 通过计算机程序很好的完成了人类所认知的世界到二维世界的转换，其工作也开创了以理解三维场景为目的的三维机器视觉的研究<sup>[13]</sup>。在视觉研究的理论上，以麻省理工学院（MIT）的 Marr 的理论是迄今为止最为完善的理论体系，其影响也最为深远。Marr 理论强调表示的重要性，它从视觉计算理论出发，将最初的原始数据到最终对三维环境重建过程分成了自上而下的三个阶段：早期的视觉处理，即所谓的“要素图”，由图像中的边缘点、线段、顶点、纹理等基本特征组成；中期视觉处理阶段，用不完整的三维信息进行描述，即所谓 2.5 维描述，是以观察者自身的坐标系来重建物体三维形状与位置的部分不完整信息描述；为了完成物体的匹配，需要对 2.5 维描述进一步处理得到物体的完整的三维描述，这就是第三阶段后期视觉处理阶段<sup>[14]</sup>。

自从 Marr 教授提出近乎完整的计算机视觉理论框架之后，国际上不断涌现出立体视觉 成功应用的案例。Boyoon 和 Gaurav 等人采用立体视觉技术跟踪移动机器人，使其在动态环境中成功地感知外界物体的运动<sup>[15]</sup>；Ratheesh 和 Georgy 等人<sup>[16]</sup>利用计算机视觉和高分辨率 GPU 实现了繁杂环境中三维物体的实时跟踪与定位<sup>[16]</sup>；南洋理工大学研发了一种传感器数据融合系统，结合光流矢量、色彩特征以及立体视觉技术估计立体摄像机的运动，该系统可完成对于视觉目标的定位与跟踪，并判断目标的移动速度<sup>[17]</sup>。

机器视觉在我国的起步比较晚，在相当长的时间内机器视觉技术还处于推广普及期，这让我国的机器视觉技术相较于其它国家的顶尖技术还是有着一些差距。但是近年来，国内的视觉识别与重建工作也取得了不小的成果，众多学者提出了许多相对完善的理论。中科院的自动化研究所系统化地介绍了机器视觉理论以及应用其中胡占义、马颂德、吴福朝等是典型代表人物，

最近研究所提出了一种室内场景的三维重建系统<sup>[18]</sup>；浙江大学研究双目立体视觉的基础上研发了用于果蔬采摘的智能机器人，该机器人可以利用双目立体视觉技术获取各种果蔬的三维信息并进行自动定位并采摘<sup>[19]</sup>；哈尔滨工业大学机器人技术与系统国家重点实验室完成了一种新型移动机器人避障系统设计，采用红外传感器与双目视觉传感器之间的结合获得完整的距离信息，并利用这些信息实时帮助机器人感知其位置相对于各种未知环境下的障碍<sup>[20]</sup>。

近几十年中，随着计算机技术的兴起，识别与三维视觉技术取得了飞速的发展，但当今的该技术仍面临着诸多问题与挑战。首先，由于三维视觉是在模拟人类视觉的原理的基础上发展起来的，而人类视觉是综合了心理学、生物学和医学等多方面的复杂学科，人们对这一复杂学科尚存不解之处，因此对于立体视觉系统还没有建立统一的方法和标准<sup>[21]</sup>。而且，针对不同的应用场景，需要建立不同的立体视觉系统，这就对立体视觉的通用性产生了极大的影响；其次，图像信息的获取过程中存在许多干扰因素，需要采用合适的图像预处理方法来提高图像的有效性；再次，立体匹配是计算机视觉中三维重建技术的重中之重，匹配结果的好坏直接影响重建的精度。

#### 1.4 三维重建的常见方法<sup>[22-23]</sup>

从摄像机数目来看,基于图像的三维重建技术可以分为单目视觉法和多目视觉法（常见包括双目、三目等）。

单目视觉法是指仅使用一台摄像机实现三维重建的方法，其优点是实验设备简单，节省成本；缺点是对周围环境的要求比较高,且由于单一的二维图像或图像序列包含的三维信息往往有限，在算法和理论研究水平的限制下，一般无法做到高度的三维还原，在实际应用时效果也往往不是很理想。另一类多目视觉法则是使用多视点的多幅图像实现三维重建,这种方法通过得到不同图像相同特征点之间的匹配关系,根据该匹配关系求出对应特征点的三维空间坐标信息,从而实现三维重建。其优点是通过从不同视角获取的不同图像包含了丰富的三维信息，通过三角关系进行一定的变换和计算即可获得深度信息并完成三维重建；缺点是有时计算量大,时间复杂度较高。

双目视觉法，即使用两台摄像机获取同一个物体的两个不同视角的图像序列,再根据三角原理得到匹配点的深度信息。这种方法是最接近人类三维成像原理的方法，也是最热点的研究领域，本研究所做的工作即为一种双目视觉法。一般来说，采用双目视觉的三维重建工作分为以下五个步骤：

- a. 图像采集；
- b. 相机标定；
- c. 特征提取；
- d. 立体匹配；

装  
订  
线

### e. 三维建模。

#### 1.5 本课题来源及所做工作

本课题的来源是导师的项目组正在进行的一个让四旋翼飞行器自动平衡倒立摆的课题。该课题的研究侧重在于飞行信息反馈和控制算法，而本研究的内容则是该课题的另一个衍生问题，即以如何确定倒立摆的位姿为核心的问题。

在苏黎世联邦理工大学（ETH Zurich） Flying Machine Arena 实验室的 Markus Hehn 和 Raffaello D’Andrea 等所做的类似研究<sup>[24]</sup>中，使用到的定位系统是一套昂贵、专业的室内 Vicon 动作捕捉系统<sup>[25]</sup>（Vicon 是一家专门提供数字光学和视频化运动追踪解决方案的公司），该套系统目前广泛应用于电影特效行业和人体运动分析等领域。而我们希望做到的，是在减少成本的基本要求下，增加源项目的应用范围：即不再需要在配套有专业 Vicon 系统的实验室环境中，而是在任何地点，不论室内室外，都可完成都可让飞行器完成倒立摆自平衡功能。这即为本文所做工作的最初出发点。

本研究具体的工作包括：课题背景分析，简要概述计算机视觉和实时三维重建的技术特点和发展情况；三维重建的技术核心分步分析、总结和工程实现，其中主要包括图像采集、图形识别、立体匹配、标定及三维建模。本研究将着重分别对这几个技术环节进行资料搜集，归纳最常见的实现方法，并比较分析他们的优劣，最后选择最适合本课题背景的方法进行工程实现。本文最后的成果将是一套能够实现较准确且灵敏的能够对类倒立摆物体实现实时三维重建的计算机系统，并尝试验证其可用性。

作为“机械设计制造及其自动化”专业的本科生，作者希望能够在研究生阶段以自动化、机器人控制学及相关算法为主要研究方向，因此本研究也是对于与计算机科学相结合的机器人学领域热点问题的初次探索。在本研究中并没有提出新的理论或算法，而更多是对现存方法的一次广泛而浅显的学习和梳理，并尝试运用这些知识搭建一套能够完成基本要求的计算机系统。因此本课题对于作者今后的研究道路意义重大，且也是一次十分难得的实践经历。

#### 1.6 本文段落安排

本文将分为五大章。

第一章“引言”，主要结合课题背景作文献综述，并对整个课题作简要概述。

第二章“理论与算法研究”和第三章“代码实现与工程实践”为本文的核心，即实现从图像采集到识别与三维重建的一整套系统。这两章将分别分别从理论研究和具体实现的角度对本课题所做工作分步骤进行展开和描述。

第四章“测试结果与讨论”中主要介绍对该系统的测试及调试过程，并作结果分析。

第五章“结论与展望”则是对整个课题的总结和展望。

## 2 理论与算法研究

### 2.1 图像预处理

在计算机图形学中，图像预处理是为了让图像更有利于应用在后续步骤而预先对原始图像进行的处理。一般较复杂的图像处理应用系统都对输入源图像的质量有着不同的要求，而直接经传感器或摄像头提取的原始图像一般包含大量与系统本身不相关的信息，如无关环境背景、光源差异及电子摄像头本身造成的噪点等，若不对它们进行必要的预处理，势必会引起系统功能实现困难甚至失效；另一方面，除了去除无关信息外，预处理也可以通过增强亮度、调整对比度或局部切割放大等技术手段让系统所需求的图像信息更加明显而易于识别。

图像预处理广义上一般包括格式转换、饱和度调整、灰度转换、灰度调整、降噪、锐化、平滑等，甚至也可以包括图像的分割、特征识别与提取、内容理解等。例如：车牌识别系统中一般需要将所拍摄的车牌按照字母分割开来，并作旋转及变形、灰度及对比度处理等，以使字母识别程序能够顺利地应用于每个字母<sup>[26]</sup>；再如由超声或放射成像生成的医学图像中所包含的斑点型噪声容易让医生对于病灶的诊断出现误判，而适当的滤波处理既可滤去噪点又可以保留图像细节信息<sup>[27]</sup>。

而本课题中，图像预处理的核心目的在于使得后续的圆形目标识别工作更加精准快速，即增加识别程序的正确率和定位系统的准确率，因此图像预处理应完成的基本任务即为使得圆形目标在背景中更加清晰明显，并尽量去除其他一切无关信息。本节对本课题可能遇到的图像预处理技术的基础应用作简要介绍：

#### 2.1.1 灰度图像转换

色度学认为光的三原色是红色（Red）、绿色（Green）和蓝色（Blue），三种光相加会成为白色光，即RGB三原色理论。这是由于人类有三种视锥细胞分别对红、绿和蓝光最敏感。在计算机中，当前主流的对颜色的标准数值表示方法就是根据三原色原理产“24字节模式”，即每像素24位（bits per pixel，一般简写为：bpp）编码的RGB值：使用三个8位无符号整数（0到255）以分别表示红色、绿色和蓝色的强度。由这种表述所生成的图像称为“真彩图”，它可以包含一千六百七十多万种颜色组合，对人类的眼睛来说其中许多颜色已无法确切分辨。<sup>[28]</sup>

“灰度图像”是一种不包含色彩信息而只包含颜色深浅和亮度信息的图像，与黑白两色图像不同，灰度图像也包含从纯黑到纯白之间的灰度信息，一般使用每像素8字节存储，即将灰度分为0~255共256个等级（一般地，0表示纯黑，255表示纯白）。灰度图像的好处在于较小的存储空间和出色彩以外较多的可使用信息。<sup>[29]</sup>

真彩图转换为灰度图的数值方法一般使用加权平均算法，根据国际照明委员会（CIE）创立的著名的“CIE1931 色彩空间”<sup>[30]</sup>，及生物学上人类三种视锥细胞的敏感程度差异，一般灰度值表述为：

$$Y_{\text{linear}} = 0.2126R_{\text{linear}} + 0.7152G_{\text{linear}} + 0.0772B_{\text{linear}} \quad (2.1)$$

其中  $Y_{\text{linear}}$  一般称为线性亮度（linear luminance），且可注意到三种原色对应的系数分别为 0.2126、0.7152 和 0.0772，这是由于实验证明人类对绿色敏感度最高，而对蓝色最为不敏感。因此该加权算法得到的灰度图最能够真实反应人类眼中的世界。

上述方法通常用于色彩学研究，但在实际业界常用的电视和视频制式均采用 sRGB 色彩空间的“YUV 色彩编码方案”，如 PAL（世界大部分地区所采用，包括中国和大部分欧洲地区）、NTSC（美国、加拿大、日本等国家和地区实用）、SECAM（主要由俄罗斯、法国等地采用）。在 sRGB 色彩空间中，存在一项非线性矫正“伽玛校正”（Gamma correction），它是现在业界普遍采用的一种对于人类视觉对环境整体亮度自适应补偿机制的算法模拟，是一种非线性亮度调整<sup>[31]</sup>。在 YUV 及相似类别的色彩编码中，图像的 RGB 值已经是经过伽玛校正的非线性值，则非线性亮度  $Y'$  则可简单地由下式得到：

$$Y' = 0.299R' + 0.587G' + 0.114B' \quad (2.2)$$

在 Matlab 中，常用的“rgb2gray”等灰度图生成函数便是使用该算法。

### 2.1.2 二值图像

二值图像（Binary Image）是一种每个像素点都只可能为两种值的图像，其所需存储空间极小。二值图像的两种颜色可以为任何两种颜色，但是通常使用黑、白二色。真彩图像与二值图像之间的转换一般通过设置对 RGB 值的阈值对图像进行划分，即便如此，设置阈值的方式仍多种多样，如对图片全局取平均值，或是设置窗口，对局部分别设置阈值。由此，在 Matlab 中对图像进行二值转换的算法为“imbinarize”，而对于该函数提供了两种方法：“adaptive”或“global”。“adaptive”方法是一种自适应阈值设置，当需要在图片中提取更多颜色差别较大的或是细节化的信息时更为适用；而“global”法则在当目标物体与整幅图片颜色区分较大时更为适用。

由于在本课题的实验环境设置中已将目标物的几何形状尽量简化，且将其与环境进行了颜色上的明显区分，实用全局阈值设置的二值图转换不但能够准确提取目标物，又可以节约计算时间和存储空间，显然二值图像转换相较于灰度图转换在本课题环境下是一种更可取的预处理方法。

### 2.1.3 图像平滑<sup>[32-34]</sup>

在数字图像和影像的获取、传输和处理过程中，由于种种原因会产生各种形态的噪声，使得图片质量受到影响从而提高图像识别，尤其是像本课题所涉及的边缘检测技术的困难程度，

因此图像平滑的预处理技术是不可或缺的。图像平滑算法的本质是“滤波”，且针对不同的噪声模型有不同的滤波函数。以下对一些最普遍和常用的噪声模型和图像平滑算法及它们的特点作分类讨论。

### (1) 噪声模型

#### a. 椒盐噪声

“椒盐噪声”表现为随机分布的明暗点状噪声（见图 2.1），由于高灰度值的噪点为明亮点形似盐粒而低灰度值的暗点形似黑椒粒而得名。这种噪声一般由成像过程中的短暂停留或数据传输中的数据丢失或其他错误产生。其概率密度函数为：

$$p(z) = \begin{cases} P_a, & z = a \\ P_b, & z = b \\ 0, & \text{else} \end{cases} \quad (2.3)$$

如果  $b > a$ ，像素值  $b$  在图像中显示为一亮点， $a$  值显示为一暗点。当  $P_a$  为零时，表现为“盐”噪声；当  $P_b$  为零时，表现为“胡椒”噪声。



图 2.1 一张带有典型“椒盐噪声”的图像

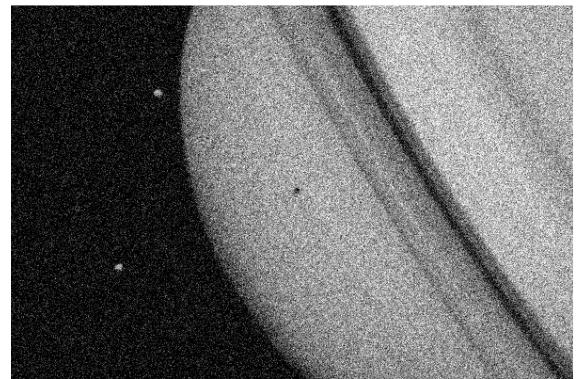


图 2.2 一张带有典型“高斯噪声”的图像

#### b. 高斯噪声

“高斯噪声”指概率密度函数为高斯分布（即正态分布）的噪声，故又称“正态噪声”。与“椒盐噪声”表现出的点状形态不同，“高斯噪声”是连续密集的噪声（见图 2.2），在生活中十分常见。在实践中该噪声模型常用于模拟电子电路噪声和低照明度或高温带来的传感器噪声。其概率密度函数为：

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{(z-\mu)^2}{2\sigma^2} \right\} \quad (2.4)$$

其中  $z$  表示图像元素的灰度值； $\mu$  表示  $z$  的期望； $\sigma$  表示  $z$  的标准差。

### (2) 平滑滤波算法

#### a. 均值滤波

“均值滤波”，或称“领域平均法”，是最简单的一种线性平滑滤波算法。其表达式为：

$$g(x, y) = 1 / N^2 \sum_{(m,n) \in S} f(m, n) \quad (2.5)$$

即对于每一个像素 $(x, y)$ 选取它的一个窗口领域 $S$ （最简单的形式是一个 $N*N$ 的方形窗口， $N$ 为奇数），将中心像素的值修改为 $S$ 中各个像素点值的平均值。

均值滤波器可以有效降低各种类型的噪声，但是由于它容易模糊图像，使得边缘特征变的非常不明显，因此实际中一般不采用。

#### b. 中值滤波

对于空间密度不大的椒盐噪声，“中值滤波”能够起到有效的抑制作用，且可较好的保持图像中的边缘特征。与均值滤波不同的是，中值滤波是一种典型的非线性平滑滤波处理算法，也是最广泛采用的简单算法。所谓中值滤波，就是在灰度值图像中，对于任一个像素点选定一定大小的领域，将该像素点的灰度值修改为它的领域内各点灰度值的中值，并对图像中的每一个像素均做上述处理。该算法可用下式表示：

$$y_{ij} = Med[f(m, n)] \quad (2.6)$$

其中 $(m, n) \in S$ 。该算法其直接结果就是将与周围灰度值差别较大的个别像素点修改为与周边相近的值，以此有效去除点噪声。但需注意的是对于一些尺寸较小的细节（点、线、尖顶等）较多的图像，中值算法适用性较差。

#### c. 高斯滤波

上述的两种线性和非线性的滤波法均属“空间域”中的操作，而高斯滤波则属“频域”方法。该方法的表达式为：

$$G(u, v) = H(u, v)F(u, v) \quad (2.7)$$

其中 $F(u, v)$ 是原始图像的傅立叶变换，可以通过快速傅立叶变（FFT）换得到； $H(u, v)$ 称为传递函数，即所做的具体滤波操作； $G(u, v)$ 是所得到的平滑后的图像的傅立叶变换，可将其通过逆傅立叶变换（IFFT）得到最终图像。

一般地，在频域中，原始图像中的正态噪声、边缘、细小特征等都表现高频，而大面积相近颜色的色块则表现为低频。因此，为了去除噪声，一般传递函数 $H(u, v)$ 使用：

$$H(u, v) = \begin{cases} 1, & D(u, v) \leq D_0 \\ 0, & D(u, v) > D_0 \end{cases} \quad (2.8)$$

其中 $D(u, v)$ 代表频率坐标系中点 $(u, v)$ 到原点的距离； $D_0$ 是一个规定的非负值，为该滤波器的截至频率，即只保留低于 $D_0$ 的低频信号而去处含有噪声的高频信号。因此这种方法常称作“低通滤波法”，在实际使用中对于去除高频高斯噪声并保留其他图像信息的效果较好。

#### d. 其他

除了以上提到的三种常用的基础平滑算法外，还有许多已发展成熟且经过论证并被广泛使用的更复杂的算法。例如非线性算法中的自适应算法，这类方法通过一定的算法可以判断图像中的边缘元素，从而做到平滑图像的同时尽量不模糊边缘，这类算法又称为“边缘保持滤波

器”。又例如“稳健平滑滤波器”，这类算法的“稳健”体现在它们牺牲一定的平滑效果以保持首要任务，即图像边缘不能够被模糊。由于这些算法一般比较复杂，因此本文不做赘述。

## 2.2 圆形物体识别与提取<sup>[35]</sup>

在本课题中，为了准确地定位棒形目标物体的位姿，提出的方案是在木棒的上下两端分别固定乒乓球，让计算机对拍摄到的画面中的乒乓球进行识别和提取。设计该方案的原因是圆形作为最常见的、简单的，且具有旋转不变性的图形，对其的识别算法的研究已经非常成熟，这降低了整个系统的设计难度。

识别圆形物体的算法很多，比较常用的有 Hough 变换法、模板匹配法、模糊识别法、神经网络法等。本文主要介绍两种较简单的、便于短时间内理解并编程实现的方法：圆形度法和 Hough 变换法。

### 2.2.1 圆形度法

圆形度的定义：

$$e = \frac{4\pi A}{C^2} \quad (2.9)$$

其中  $A$  是圆的面积； $C$  是圆的周长。

图形面积：对于数码图像来说，最直接的计算图形面积的方式就是对边界内像素点的个数求和。对于一副像素  $M*N$  的二值图像来说，其显示图形的面积  $S$  的计算公式即为

$$S = \sum_{x=1}^N \sum_{y=1}^M f(x, y) \quad (2.10)$$

$f(x, y)$  是二值图像的像素值，一般 1 表示白色目标物，0 表示黑色背景，显然该公式统计了目标物所占像素个数。

圆形度  $e$  是一个描述物体类圆程度的指数，一个图形的圆形度越接近 1，说明它越接近圆形。类似地，正方形的圆形度为  $\pi/4 \approx 0.79$ ，正三角形的圆形度为  $(\pi \times \sqrt{3}) / 9 \approx 0.60$ 。

圆形度识别法是一种快速简单的类圆识别法，但它的局限性非常多。首先，它要求闭合边界：圆形度法一般通过将图像转为黑白二值图像后找到闭合图形并填充为完整图形后对其做圆形度计算，而边界必须为闭合的才能实现填充步骤。因此该方法对原图像和与处理过程的要求过高，因为拍摄环境稍有轻微的扰动，图像灰度值变化能够使转化为二值图像后的图形边界不完整。其次，这种方法识别准确率不够高，实用性较低。因为图形的圆形度是一个非常宽泛的指数，一个椭圆的圆形度也可以跟三角形一样，圆形度并不能说明该图形的形状，只描述图形跟圆形的接近程度，因此一般该方法不被使用。在本课题中，该方法理论上是可行的，因为乒乓球一类的球类物体不论位姿如何变化，始终会在所拍摄的二维图像中呈现圆形。但由于其对拍摄质量要求较高，在本课题中暂列为候选方案。

### 2.2.2 霍夫变换法<sup>[36-37]</sup>

霍夫变换（Hough Transform）是图像处理中广泛运用于集合图像识别的一种变换法，其本质是将几何坐标下的图形转变到参数坐标下。为了快速地阐明霍夫变换的原理，先由对最简单的几何图形直线的变换为例。

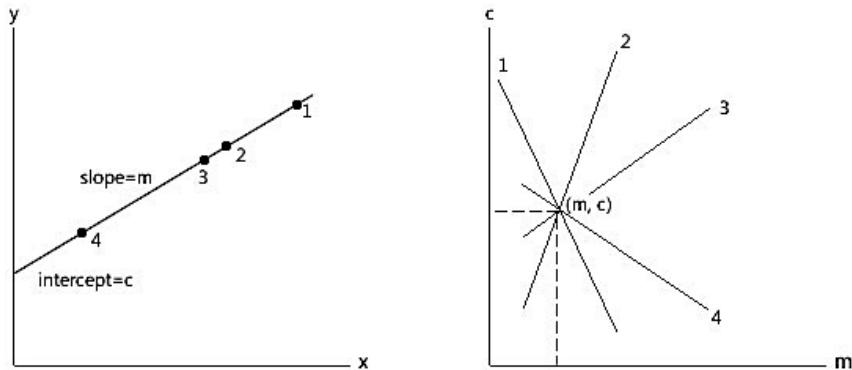


图 2.3 对直线 $y = mx + c$ 的霍夫变换（左：xy域；右：mc域）

假设现要求在图 2.3 左（xy 坐标系）中进行直线特征识别提取，并假设我们已通过一定 的方法在图中取得了四个采样点。已知直线的一般方程为： $y = mx + c$ ，则未知参数为斜率  $m$  和截 距  $c$ 。现将四个采样点的数据表达在  $mc$  坐标系上。即对于点 1，假设其坐标为  $(x_1, y_1)$ ，则通过点 1 的直线所具有的  $mc$  参数可表达为： $c = -x_1 m + y_1$ ，即图 2.3 右中的线 1。对于其他采样点以 此类推。最后在所得到的参数坐标系中找局部最大值（local maxima），即穿过直线最多的点， 该点的坐标  $(m, c)$  即为原直线的斜率和截距。如此便完成了对原图中直线的识别过程。

但是在这种方法中，对于斜率为无穷大的直线无法完成 Hough 变换，因此通常使用极坐标 表示直线。在这种情况下，未知参数为  $(r, \theta)$ ，即半径和角度。

圆形霍夫变换（circle Hough Transform，简称 CHT）即用类似的方法在圆形的参数坐标中 通过找局部最大值识别圆形。任意圆形在平面坐标中可描述为： $(x - a)^2 + (y - b)^2 = r^2$ ，则其 位置参数有三个： $a$ 、 $b$  和  $r$ 。但是在计算机处理中，在参数坐标的维度超过二维后，在其中找局 部最大值的过程的时间消耗和所需存储空间被证明会急剧增加。因此在实际处理中，一般对于 未知半径圆形的识别也不使用三维参数空间，而是增加一步半径估计，转化为一个二维参数空 间问题，这种方法现在被广泛使用，在 Matlab 中的 imfindcircles 函数即使用了降维的霍夫变换 来识别定位圆形特征。

在图 2.4 右的参数空间中，中心穿过线条最多的局部最大值即为目标圆的圆心坐标。圆形霍 夫变换的优点在于对噪点敏感度较低，缺点在于其计算强度较大，因此实际运用范围比较有 限，但是在本课题环境中是一个合适的选择。

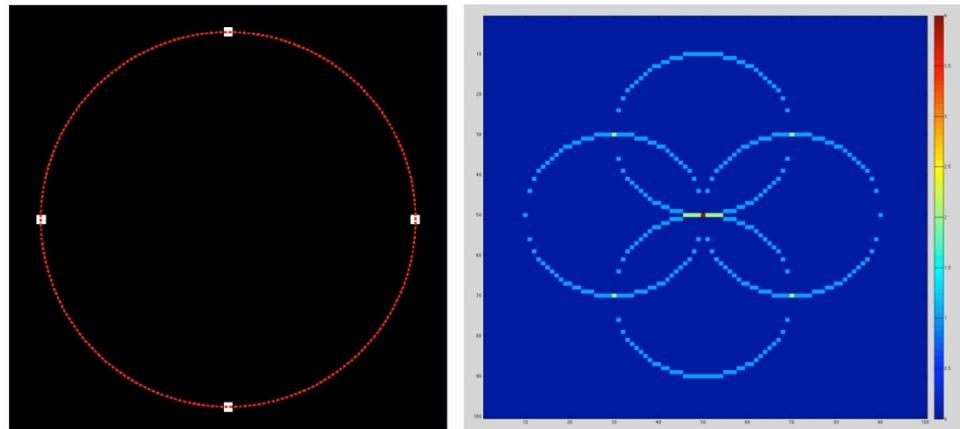


图 2.4 对固定半径圆形的霍夫变换

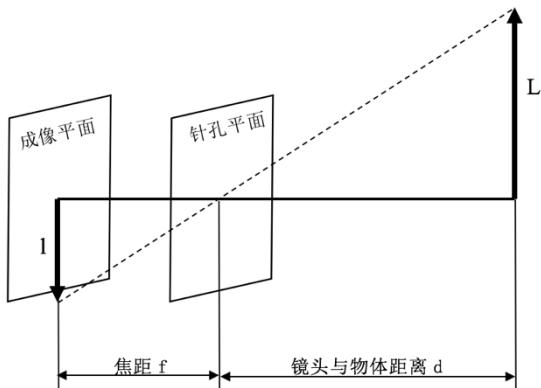
2.3 摄像机数学模型及三维坐标线性匹配<sup>[38-39]</sup>装  
订  
线

图 2.5 小孔成像示意图

一般的，相机的透镜成像模型可以简化为小孔成像模型，如图 2.5 所示。在摄像机模型中，约定以下四个坐标系：

(1) 图像的物理坐标系 ( $OXY$ )

该二维平面坐标系是假设的物体成像平面，其原点  $O$  为摄像机光轴与该平面的焦点。

(2) 图像的像素坐标系 ( $O_1UV$ )

该坐标系即为最终得到的数码成像结果，该平面与  $OXY$  重合，但它的原点  $O_1$  一般约定为图像的左上角。 $OXY$  与  $O_1UV$  的关系如图 2.6 所示。图像物理坐标系的原点  $O$  即为图像像素坐标系中的  $(u_0, v_0)$ 。

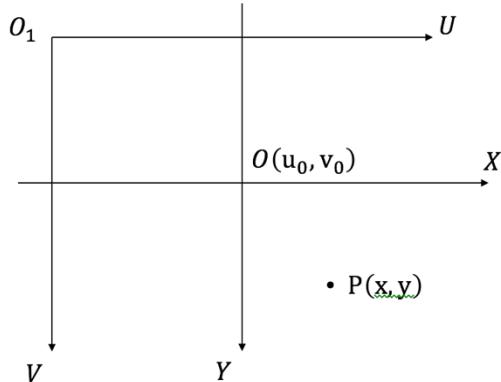


图 2.6 图像坐标系

由此，对于这两个坐标系的转换有如下关系式：

$$\begin{cases} u = \frac{x}{dx} + u_0 \\ v = \frac{y}{dy} + v_0 \end{cases} \quad (2.11)$$

其中 $(x, y)$ 是图 2.6 中点P对应的图像物理坐标， $(u, v)$ 则是其对应的像素坐标； $dx$ 、 $dy$  是 CMOS 传感器的一个像素单元分别在 $x$ 、 $y$ 方向上的长度，一般由厂家给出，或通过标定测得。该式齐次式即可表示为：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.12)$$

### (3) 摄像机坐标系 ( $O_cX_cY_cZ_c$ )

这是一个三维直角坐标系，其原点 $O_c$ 为摄像机光心， $Z_c$ 与光轴重合， $X_c$ 、 $Y_c$ 分别与图像物理坐标系 $X$ 、 $Y$ 平行。

### (4) 世界坐标系 ( $O_wX_wY_wZ_w$ )

该坐标系即表示真实世界的三维空间，其原点为人为自定义。四个坐标系的关系如图 2.7 所示。摄像机坐标和世界坐标可以通过旋转和平移得到，转换关系式为：

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} + \mathbf{t} \quad (2.13)$$

其中 $\mathbf{R}$ 是一个 $3 \times 3$ 的矩阵，代表旋转变换； $\mathbf{t}$ 是一个三维列向量，代表平移变换。该式齐次式即可表示为：

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (2.14)$$

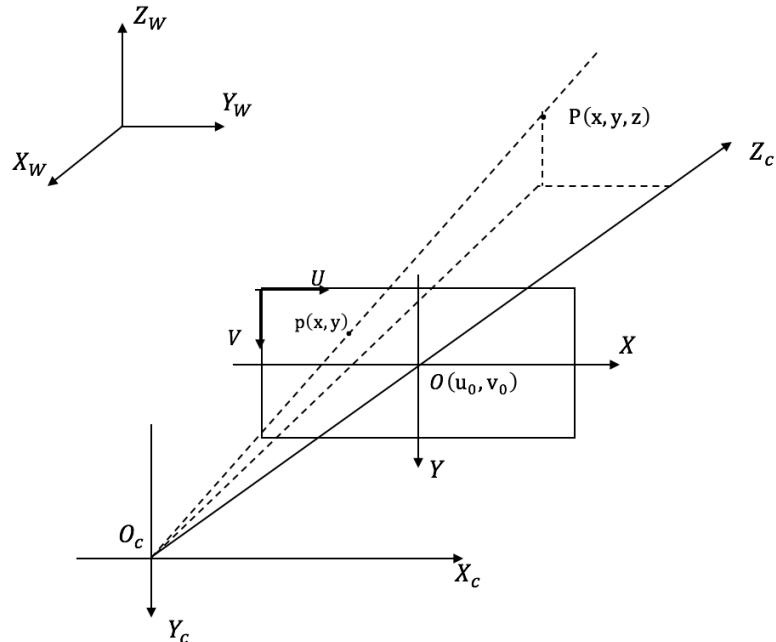


图 2.7 摄像机与世界坐标

装  
订  
线

摄像机坐标与图像物理坐标之间的转换关系式为：

$$\begin{cases} x = f \frac{X_c}{Z_c} \\ y = f \frac{Y_c}{Z_c} \end{cases} \quad (2.15)$$

其中  $f$  是相机焦距。将其化为齐次式为：

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (2.16)$$

由此得出摄像机像素坐标与世界坐标之间的关系：

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (2.17)$$

其中  $f_x$  和  $f_y$  一般称为  $x$  方向和  $y$  方向上的焦距，显然有：

$$\begin{cases} f_x = \frac{f}{dx} \\ f_y = \frac{f}{dy} \end{cases} \quad (2.18)$$

## 2.4 畸变与非线性成像问题

线性模型仅是一种理想条件下的成像模型，在实际情况中，镜头在制作和成像过程中会出现一定的畸变现象，这些畸变会导致空间点的空间位置和它在图像上的像点位置之间的关系并非完全符合线性模型，因此一般为了得到精准的视觉重建效果，需要在考虑畸变补偿。

一般的非线性成像模型可表示为：

$$\begin{cases} \tilde{x} = x + \delta_x(x, y) \\ \tilde{y} = y + \delta_y(x, y) \end{cases} \quad (2.19)$$

式中的 $(\tilde{x}, \tilde{y})$ 是通过章节 2.3 的线性成像模型得到的理想图像物理投影坐标， $\delta_x(x, y)$ 和 $\delta_y(x, y)$ 分别为图像物理坐标系横坐标与纵坐标上的非线性畸变误差。根据 Weng<sup>[40]</sup>的研究，非线性畸变误差可表示为：

$$\begin{cases} \delta_x(x, y) = k_1(x^2 + y^2) + p_1(3x^2 + y^2) + 2p_2xy + s_1(x^2 + y^2) \\ \delta_y(x, y) = k_2(x^2 + y^2) + p_2(3x^2 + y^2) + 2p_1xy + s_2(x^2 + y^2) \end{cases} \quad (2.20)$$

其中，两室的第一项统称为径向畸变(radial distortion)，第二项为离心畸变(centrifugal distortion)，第三项为薄棱镜畸变(thin prism distortion)。公式中的 $(k_1, k_2, p_1, p_2, s_1, s_2)$ 一般统称为非线性畸变参数。

然而大量实验证明，仅考虑上述非线性模型中的径向畸变就能够完整地描述非线性畸变。此外 Tsai<sup>[41]</sup>和 Faugeras<sup>[42]</sup>都曾指出，在考虑非线性畸变的相机标定过程中，引入过多的非线性畸变参数，会导致优化算法变得复杂，不仅不能提高标定的精度，反而会引起标定结果的不稳定。因此，只考虑径向畸变的模型被广泛应用，它的数学表达式为：

$$\begin{cases} \tilde{x} = x + [1 + k_1(x^2 + y^2)] \\ \tilde{y} = y + [1 + k_2(x^2 + y^2)] \end{cases} \quad (2.21)$$

装  
订  
线

## 2.5 相机的内外参数与标定

首先，分别定义内外参数。

摄像机内参数是摄像机本身的属性，不随其位置或环境的变化而变化，但是不同的摄像机之间会有所不同。摄像机的内参和章节 2.3 所研究的相机线性成像模型紧密相关，内参表述图像空间点与图像像素坐标系之间的关系。显然，内参可直接由之前所推导的式 2.17 中的矩阵来表示：

$$\begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.22)$$

其中各个参数的具体定义见章节 2.3。

摄像机的外参数是指物体或者世界坐标系在摄像机坐标下的描述，同样从章节 2.3 中推导的世界坐标系与摄像机坐标系之间的关系，类似于内参部分，可以在式 2.17 中得到外参的矩阵形式：

$$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \quad (2.23)$$

而相机的标定即是求解相机内外参数的过程。关于相机标定的研究非常多，现有的相机标定方法可以大致分为两大类：传统的相机标定方法和相机自标定方法。前者是指根据固定的相机模型，基于形状、尺寸已知的标定参照物求出相机的内外参数；而后者是指不需要借助外界信息，仅利用图像之间的点对应信息来获取相机参数的方法。

目前主流研究中一般使用张正友教授提出一种介于传统相机标定方法与相机自标定方法之间标定方法。这种方法，既解决了传统相机标定方法操作频繁、设备复杂等缺点，又较自标定方法精度高。该方法只需要利用相机从不同角度拍摄如图 2.8 所示的标定板的多幅图像，通过标定板上面各个特征点与其在成像平面上的投影点之间的相互关系，即可求出相机的内外参数。值得注意的是由于相机普遍存在的非线性畸变，因此每幅照片中网格的投影成像需经过极大似然估计以得到理想的线性内外参数，再通过一定的迭代方法得到非线性畸变参数。作为一种被世界各地学者所广泛采用的相机标定方法，该方法已被集成于 MATLAB 中的“Camera Calibration Toolbox for Matlab”工具箱，非常易于使用。

装

订

线

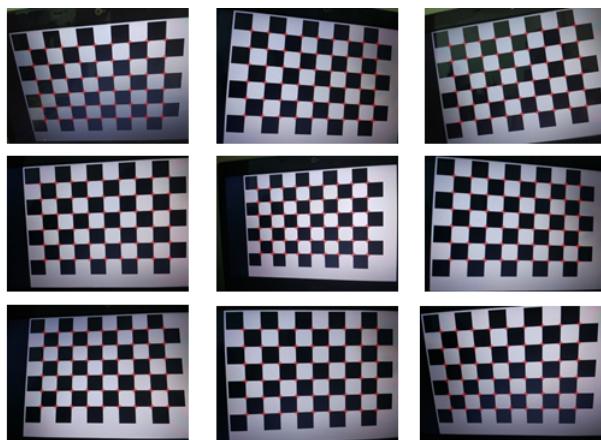


图 2.8 张正友标定法所使用的标准标定板

但是在本研究中，如将在章节 3.1 中所描述的，实验使用的摄像头厂家参数较为完备，且具有畸变较小的特点，且由于场地和工具限制等因素，本研究将暂时采用更为快捷的传统标定法，即通过摄像头的摆放和调整得到理想的拍摄角度并直接得到所有的内外参数。

### 3 代码实现与工程实践

#### 3.1 实验环境介绍

本节将分三部分介绍实验环境：硬件环境、软件环境和实验平台的搭建。

##### （1）硬件环境

计算机：MacBook Pro (Retina, 13-inch, Early 2015)。该计算机是一台较为常见的由苹果公司生产的个人电脑，其处理器为 2.7 GHz Intel® Core™ i5-5257U CPU，内存为 8 GB 1867 MHz DDR3，显卡为 Intel Iris Graphics 6100 1536 MB。

摄像头：Logitech® HD Webcam C270（见图 3.1）。本研究中采用两台由罗技公司生产的 C270 高清 USB 网络摄像头作为图像采集设备。该摄像头基于一颗 3.60mm\*2.00mm 的 CMOS 感光元件，像素尺寸 2.80  $\mu\text{m}$ ，镜头焦距 4mm。作为一款的常见家庭电子设备，其优点在于其拍摄分辨率和帧数可调、镜头畸变较小、具有可靠稳定性且价格适中，在本文所参考的一些视觉识别的研究和试验中也曾被选为图像采集器。该摄像头最高支持 1280\*960 的分辨率，但在实验阶段考虑到计算设备的数据处理能力，暂选定用 960\*720 分辨率进行图像采集。<sup>[43]</sup>



图 3.1 Logitech® HD Webcam C270

##### （2）软件环境

计算机操作系统：Windows 7 专业版。

开发环境：MathWorks® MATLAB R2017a Win-64。MATLAB 是一种用于算法开发、数据可视化、数据分析以及数值计算的高级技术计算语言和交互式环境。除了矩阵运算、绘制函数/数据图像等常用功能外，MATLAB 还可以用来创建用户界面及与调用其它语言编写的程序。尽管 MATLAB 主要用于数值运算，但利用为数众多的附加工具箱（Toolbox）它也适合不同领域的应用，例如控制系统设计与分析、图像处理、信号处理与通讯、金融建模和分析等。由于 MATLAB 具有极为数学友好型的编程语言、附加工具箱选择广泛和自带用户界面创建功能等优点，用于本课题的算法研究和调试非常合适。

## (3) 实验平台的搭建

最终实验场地的搭建效果图和部分尺寸如图 3.2 所示，被拍摄区域由两块较大的白色挡板围起，中间地面也由白纸铺垫，该区域面积约为  $500\text{mm} \times 500\text{mm}$ 。经过一些调整后发现两台摄像机被置于图 3.2 中所示位置时能够较好地拍摄到目标区域，在随机光源下拍摄效果如图 3.3 所示。根据章节 2.3 的坐标系设置方法，可以得到各个坐标系的位置关系。

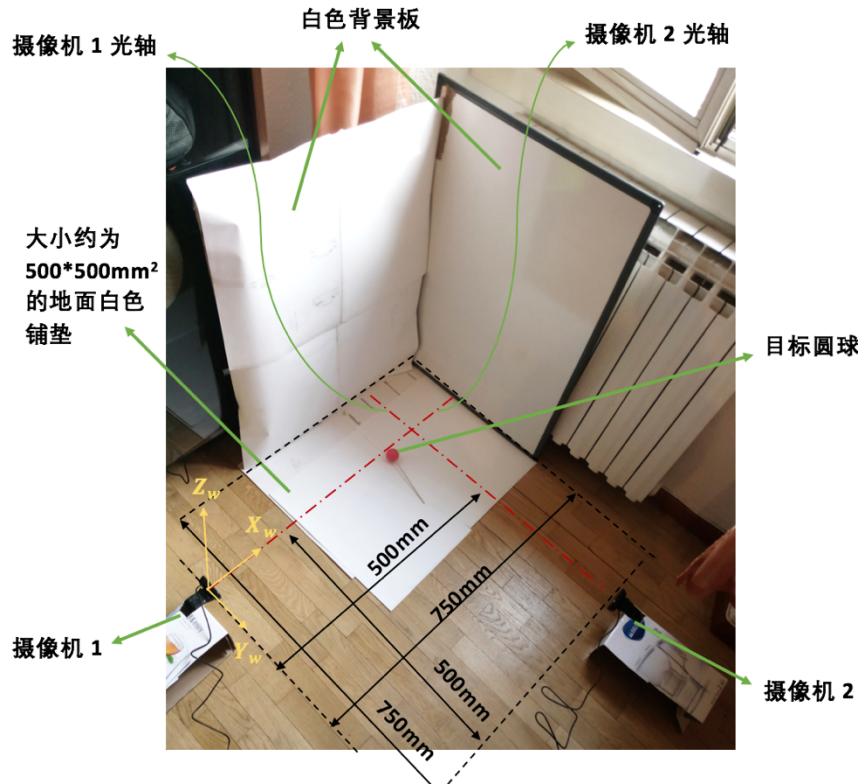
装  
订  
线

图 3.2 实验场地搭建示意、部分尺寸和世界坐标系位置

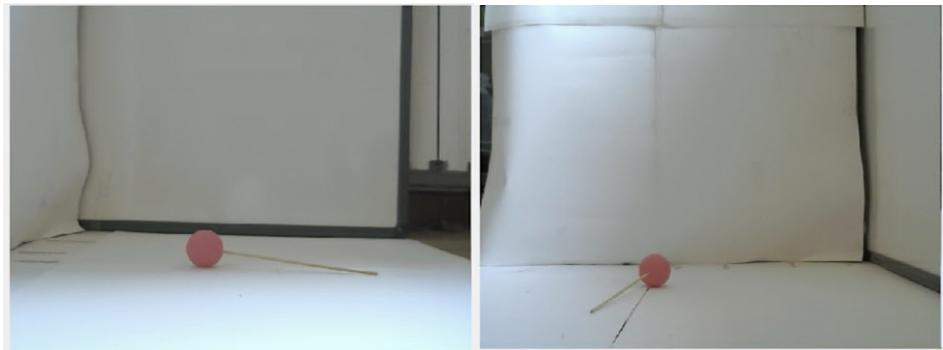


图 3.3 拍摄效果图（左：摄像头 1 右：摄像头 2）

实测试验中，将目标物组合成了如图 3.4 中显示的样子，即两个颜色（粉色）与背景板颜色显著不同的乒乓球被一根细竹签串起。经过实验证，这种颜色区分的方法中目标提取效果满

足要求，具体在下一节中将作阐述。实际操作中为了避免实验人员手部上的个别特征被误认为目标特征，应避免手部出现在被拍摄区域中，或可佩戴白色手套。

另外，均匀明亮的优质光源对于图像预处理算法的实效性和识别算法的准确性也有着举足轻重的作用。在本研究中，使用了拼接而成的较长的竹签，因此可以在拍摄范围以外进行操作而不构成图像干扰，且将光照条件调整至如图 3.4 所示样式，与图 3.3 的对比下可以看出目标圆球在合适的光照条件下显得更加突出和清晰。

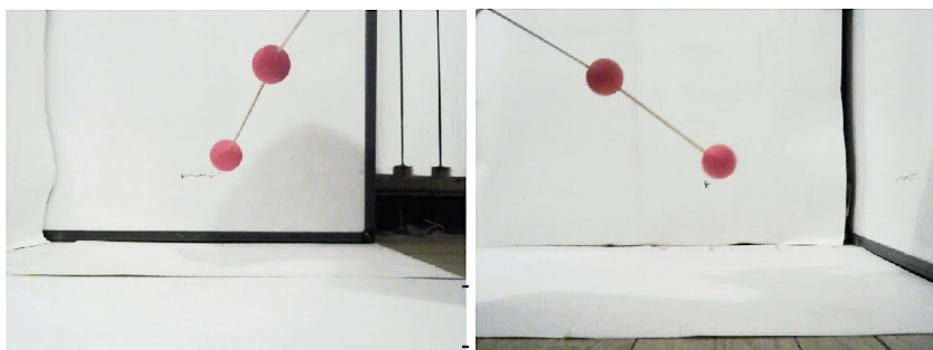


图 3.4 实际实验的目标物形态和光源效果图

装

订

线

### 3.2 核心功能的代码实现

本节将分步骤介绍各个重要模块代码编程的主要思路、主要命令和编码过程中重新编写的函数文件及其各自功能，并在必要时配以实际测试图片以验证程序的可用性。

#### （1）图像采集

在这个环节中，本实验采用了 Matlab 的一个重要工具箱（Toolbox）：“用于通用视频接口的图像采集支持工具包”（Image Acquisition Toolbox Support Package for OS Generic Video Interface），该工具箱一般需要另行下载安装。

通过调用该工具箱的“imaqhwinfo”函数可以查看连接到计算机的摄像头的各种信息，比较重要的包括数量、名称、编号、所支持分辨率等。使用“videoinput”则可导入指定摄像机、指定分辨率的视频流。

综合运用这些函数，本实验中编写了一个可在主程序中直接调用的，功能为通过指定编号的摄像头导入图像流并将播放于特定指针指向的窗口的的函数“CallCamera”，具体可见附件中的“CallCamera.m”文件。

#### （2）图像预处理

视频的本质是快速播放的图片，因此对实时视频的处理也可以看作是对每一帧的图片进行的处理。通过上述的图像采集工具箱中的“getsnapshot”函数即可截取视频当前的一帧。

如在 3.1 节中相关部分所述，在本实验环境设置中，已将目标物与背景颜色特意明显地区分开来。因此简单地调用“imbinarize”函数，并选用“global”选项，就可以有效地将特殊颜色的目标物提取出来，并将原图转化为二值图像，效果见图 3.5 和图 3.6。这种方法的原理是通过给图像的 RGB 值添加阈值来分割图像使其成为黑白二值图像，由此处两图可见区分的效果比较理想。如此便完成了所需的图像预处理的过程。

最终将截取图片和二值化图片函数均整合到了新函数“ImgPreprocessing.m”中。

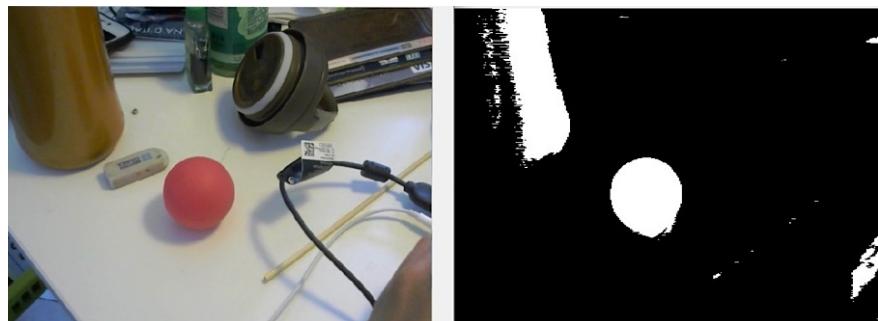


图 3.5 在普通环境下的图像预处理效果

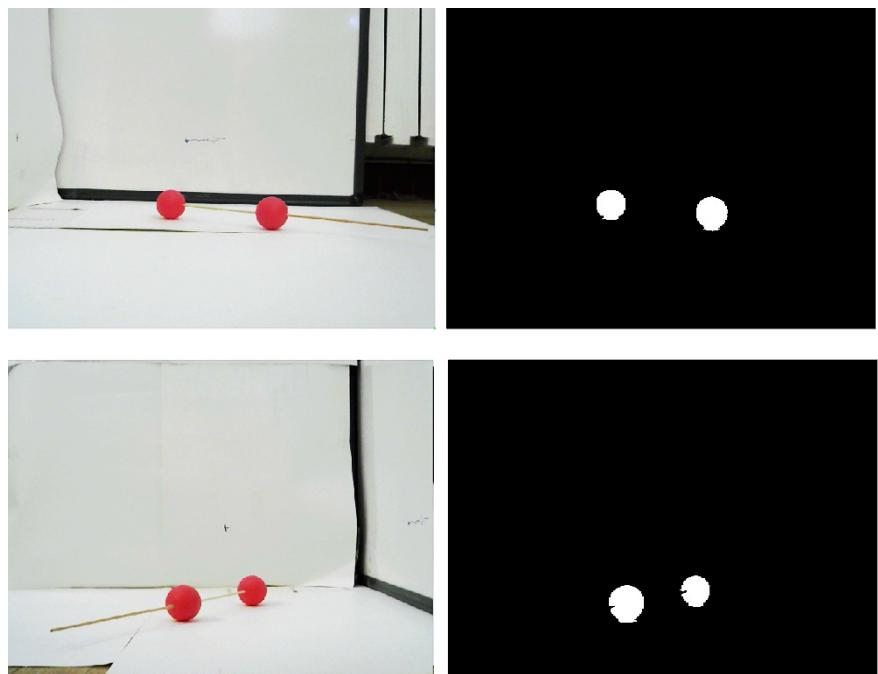


图 3.6 在实验环境下的图像预处理效果

### (3) 目标识别与提取

如在 2.2.2 中所提及，在 MATLAB 中对于一副图像中类圆图形的识别，直接使用 “[centers,radii,metric] = imfindcircles(A,radiusRange)” 函数即可。该函数运用霍夫变换法，通过对边界特征作圆形霍夫变换后，在参数空间中找局部最大值判断圆形位置和大小。该函数可运

用于不论彩色图片或是灰度图片，但为了检测结果的稳定性和准确性，此处直接将其运用于分割后的二值图像，效果见图 3.7。



图 3.7 在普通环境下 Matlab 对二值化图片的圆形识别效果

可见在视频流输入的同时，程序能够实时地识别出半径在规定范围内的圆形物体，并实时地将其位置和半径输出。为了可视化识别结果，这里使用了“viscircles”函数，其结果就是图 3.7 中二值图里的蓝色线框圆形。由此可见该识别和提取程序已基本满足了课题要求。

最终该功能也被整合到了上一大段落所提及的“ImgPreprocessing.m”函数中。

#### （4）三维坐标系匹配和三维建模

这一部分即为章节 2.3 中所研究的相机数学模型和三位线性还原的代码实现。以一个球形目标物为例，在本实验中，一旦确定了某台摄像头的世界坐和拍摄角度，并在该摄像机焦距、感光元件像素尺寸已知的情况下，假设在上一目标识别过程中程序正确地识别了目标的圆心位置，则根据章节 2.3 中的内容，可以确定一条连接世界坐标中从相机镜头光心到目标物中心的一条直线。同理，如果两个摄像机分别确定了内外参数并均正确探测到了目标物，就可以在世界坐标系中确定两条直线，显然其交点便是目标物的圆心。

这段程序被整合于“Matching.m”文件中。由于这段程序较为复杂，因此还嵌套了其他的自编函数。

例如“findIntersection.m”，这是一个输入两台摄像头分别测得的圆心的图像像素坐标系( $O_1UV$ )、焦距和像素尺寸，即可算得两条三维线条并得到其交点坐标的函数。其中值得注意的是由于不可避免的实验误差，不论来自于源图像质量、识别算法准确性或是实验环境摆放时的人为误差，都会造成得到的两条三维直线在大多数情况下没有交点，如此一来本研究实际需做到的事是找两条直线的最小距离点而不是交点，如图 3.7 所示。因此本函数所使用的算法是将两条三维线 $l_1$ 、 $l_2$ 投影至地平面 $O_wX_wZ_w$ 得到直线 $l'_1$ 、 $l'_2$ ，通过矩阵法解一元二次方程得到这两条二维直线的交点 $P'$ ，随后将该点 $X_w$ 、 $Z_w$ 坐标带入任意一条三维直线的式子，得到近似距离最小点的三维坐标 $P(X_w, Y_w, Z_w)$ 。

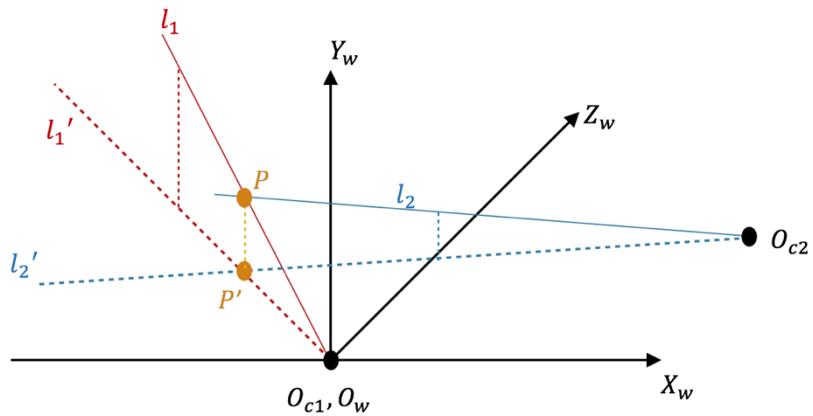


图 3.7 三维匹配关系示意图

最后在“Matching.m”函数中合并了三维建模功能，即调用自编函数“plotIn3D.m”。在“plotIn3D.m”函数中，输入参数为之前计算所得到的两线条近似距离最近点，并用 MATLAB 自带的“plot3()”函数将其谱出。在本实验中，交点计算和三维重建程序的触发情况分别为“两摄像头均探测到一个目标物”或“两摄像头均探测到两个目标物”，其余情况下，如均为检测到目标物、两台摄像头探测到的目标物数目不等，或是目标物多于两个等情况，出于保持整个程序精简和实验本身的测试性目的，均直接进行报错并不触发交点计算和三维建模。

### 3.3 软件封装

本实验中为了界面的用户友好性，采用了 MATLAB 自带的图形用户界面（Graphical User Interface，简称 GUI）设计模块，直接输入命令“guide”即可新建或打开已有的 GUI 设计模块。

出于减少进程加快程序速度的目的，在本实验中建立了两个窗口，一个用于初始的标定和调整，另一个用于正式实验阶段的目标识别和三维重建。

程序运行初始时由于摄像机的手动摆放难免具有较大的误差，因此须作一定的标定和微调。用于微调和标定的界面设计“adjustment.fig”如图 3.8 所示。其中，蓝色边框为程序启动按钮；红色边框中的四个 axes 控件用于显示图像，axes1 和 axes3 分别是两个摄像头所采集的分辨率为 640\*480 的原始图像，而 axes2 和 axes4 分别为经过预处理且将程序所识别出的圆形目标高亮圈出的处理后图像；绿色边框为程序输出的调整命令，具体在下文中作阐述；黄色边框为所探测到的圆心的图像物理坐标显示（即原点位置在图像中心），当探测到一个目标物时同时给出横纵坐标，而当检测到两个目标物时在横坐标框内显示“/”而在纵坐标框内显示两个目标物之间的纵坐标差值。该窗口的功能为调整摄像头的视角，包括三方面：俯仰角、滚转角和平偏角，其位置关系如图 3.9 所示。

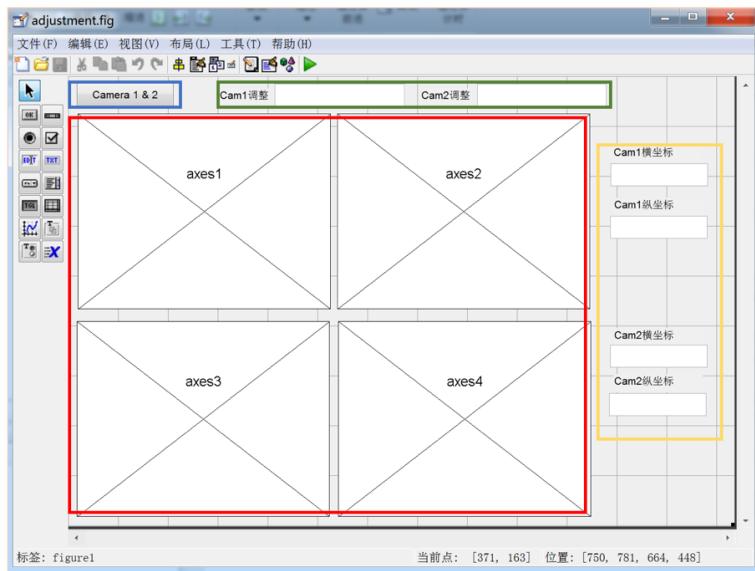


图 3.8 “adjustment.fig”图形用户界面设计图

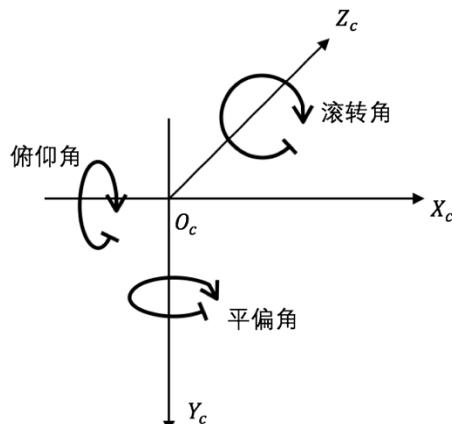
装  
订  
线

图 3.9 俯仰角、滚转角和平偏角与相机坐标的位置关系示意图

对于平偏角和俯仰角的调零，实验中用一个目标球完成：将球置于摄像头光轴上的任意位置，当识别出的圆心不在图像像素坐标系的原点位置时，通过光心和圆心位置程序给出摄像头调整命令，如图 3.10 所示中的红色框内所显示。对于平偏角调整命令可以为“Turn left”或“Turn right”；而对于俯仰角调整命令可以为“Turn up”或“Turn down”。直到当目标物圆心几乎与图像像素坐标系原点重合时，即相当于俯仰角和平偏角均等于零时，调整完成。在本研究中约定当目标物的横纵物理坐标小于 5 个像素值时调整完成，如图 3.10 的黄色框所示的坐标数值均达到调零要求。

对于滚转角，试验中用两个球作标定：将两球置于水平地面上且其连线与理想相机像素平面平行，则此时两目标球的圆心应该在像素平面上纵坐标相等，如不，则程序给出相应调整命

令，如图 3.11 红色方框所显示，调整命令可以为“Turn clockwise”或“Turn anti-clockwise”，此处也如对于摄像机平偏角和俯仰角的调零，约定当图 3.11 黄色方框内的数值小于 5 时视为调整完成。

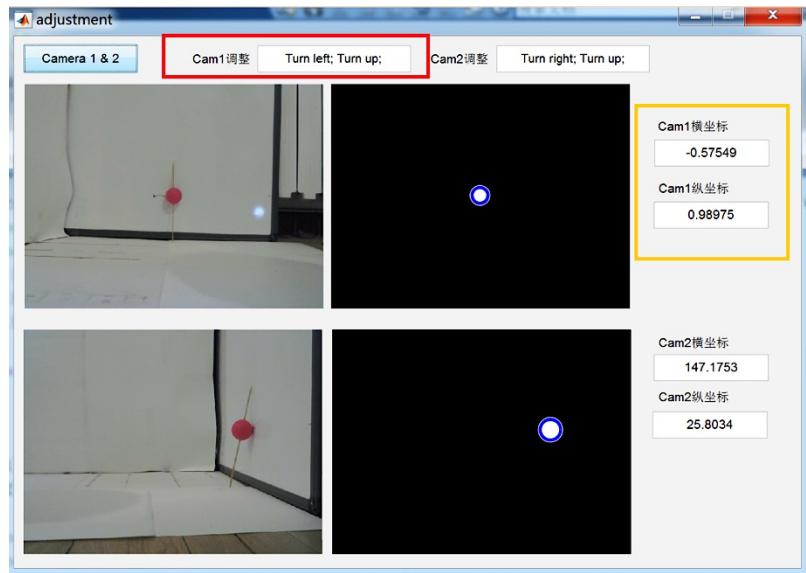


图 3.10 俯仰角和平偏角的调整



图 3.11 滚转角的调整

在两摄像头拍摄视角调整完成后，即进入主程序“GUI.fig”，其界面如图 3.13 所示。其中蓝色方框是程序启动按钮；红色方框中的 axes1 和 axes3 分别显示两个摄像头所采集到的原始图像；紫色方框显示最终得到的三维重建视图；绿色方框为探测状态监测，显示“one”或“two”以表示探测到的目标物个数，显示“more than two”、“none”或“not matched”以表

示探测到的目标数多于两个、没有探测到目标物，或两摄像头探测到的目标物数量不同等错误情况，在这些情况下，三维还原程序不启动；黄色方框中在探测到一个球的情况下显示三维匹配后目标物的三维世界坐标，在探测到两个目标物时显示两目标物球心之间的距离，在其他情况下显示“error”。图 3.14 展示了三维还原的效果图。

装

订

线

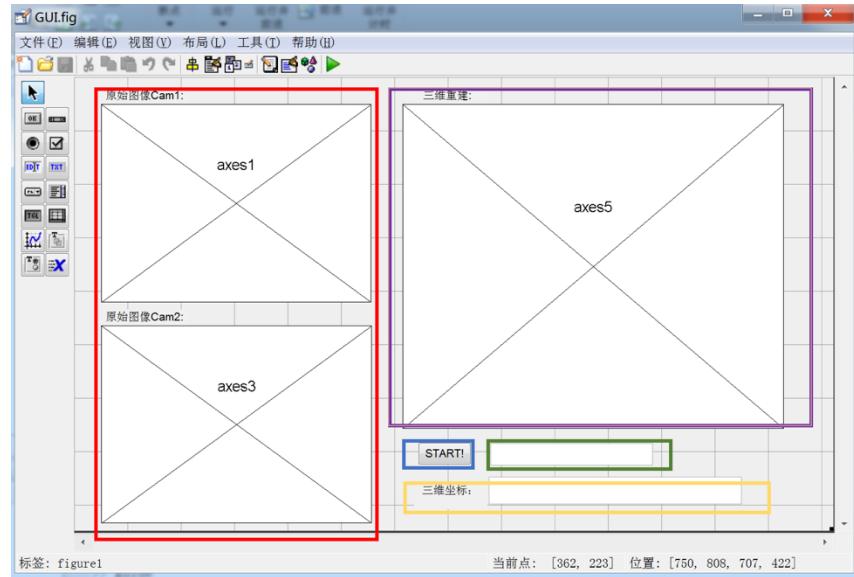


图 3.13 “GUI.fig”图形用户界面设计图

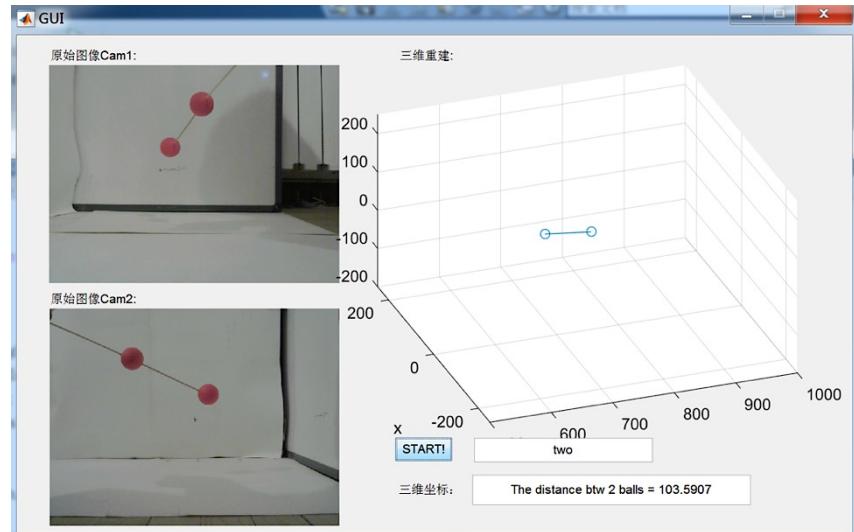


图 3.14 三维还原效果

## 4 测试结果与讨论

### 4.1 测试方案介绍

这一套基于 CMOS 的目标识别和三维重建系统可被看作是一套视觉测量系统，因此定义其优劣的标准就是测量的准确性。由此，针对本研究的测试目的就是确定三维重建的空间坐标的确切性。

但是在实际操作中，显然没有足够准确的设备条件以确定实际的球心三维坐标作为参考标准，因此拟用以下方案作为测量系统的准确性判据：

#### （1）两目标球之间的距离

理想情况下，当两目标球之间的距离由竹签固定后，所测的它们相对距离是不变的。因此可通过观察实际实验中所测的两球之间距离的变化情况和方差评估测量系统的准确度。

#### （2）以地面为参考标准目标球还原后纵坐标的准确性和稳定性

假设整套系统足够精确（理想情况），则当目标球在地面做随机运动时，经三维还原后的纵坐标 $Y_w$ 值应为常数。因此拟从这方面入手，观察目标球紧贴地面运动过程中纵坐标 $Y_w$ 的变化情况和方差，并推测可能的误差原因。

#### （3）两目标球球心之间的连线相对于地面的角度

假设将本课题成果运用于章节 1.5 中所提到的四轴无人机上的倒立摆自平衡研究中，那么倒立摆相对于地面的角度极为重要，因为这是倒立摆质心运动学建模的核心参数。此外，由本课题导师的另一个课题中所研究的一种角标刻度仪即可得到目标棒与地面的精确夹角。但是由于本研究完成于异地，无法使用现有的角标仪进行测量参考，因此在本文中暂无法讨论该测试方案的实施。

### 4.2 具体测试步骤与结果

#### 4.2.1 实验设计

根据章节 4.1 的论述，本课题特别设计了两个实验，以测试三维还原系统的精度。下面详细两个实验的设计：

##### （1）实验一

将两球之间的实际距离固定于 100mm，以一确定路径做运动，并将探测到的两球之间距离作实时记录，观察其变化趋势和准确性。为了尽量对所能拍摄到的区域做测试，此处将路径规划为图 4.1 中所示的两种形式。

其中，路径 1 是在尽量保持两球连线垂直于地面的条件下进行的，这种路径覆盖了大部分的拍摄范围，且其测试结果对于观察镜头畸变及摄像头位置与拍摄角度对于系统准确性的影响

很有帮助；路径 2 是在保持紧贴地面的球位置不变的情况下进行的，其拍摄范围主要集中于拍摄画面的中部，使得镜头畸变的影响减小，该测试路径可看作是对于实际运用中倒立摆及其他类似移动目标物的运动模式的模拟，这种路径可以综合测试系统的三维还原准确性。

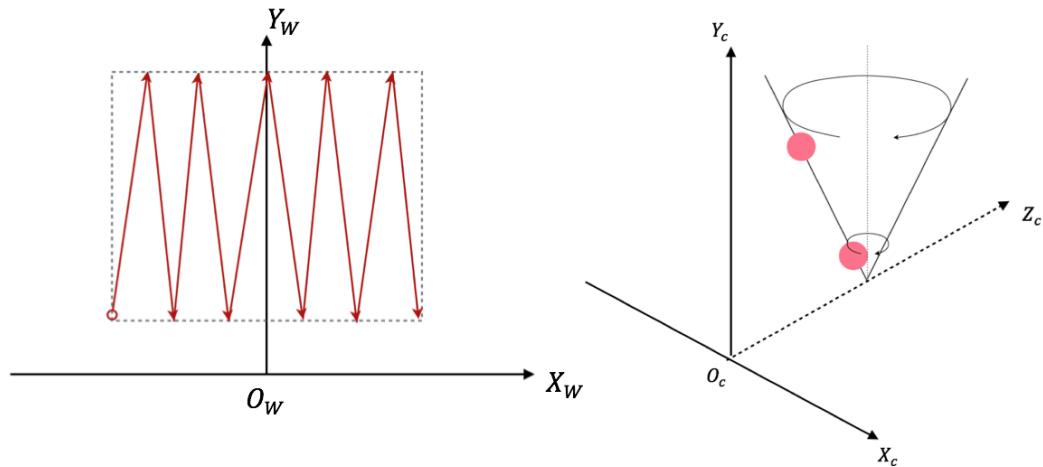


图 4.1 实验一的两种路径规划（左：路径一；右：路径二）

装

此外实验还另外设置一个对照组：对于静止目标的测试。在这种情况下观察所探测到的目标物之间距离变化可以得到大致得到系统误差，该误差主要来源于图像采集器和程序本身的稳定性。

## （2）实验二

订

只用一个目标物，使其紧贴地面沿着与实验一中的路径 1（见图 4.1 左）相同的路径运动，并将最终三维匹配结果中的 $Y_w$ 坐标值与实际地平面高度做比较。该测试主要用于观察相机坐标与世界坐标的匹配程度（即摄像机位置与拍摄角度）。

线

同样的，在该实验中设置一个对照组，即测试对于单个静止目标物的探测结果中 $Y_w$ 坐标值的稳定性，以反应该实验中系统误差的影响。

### 4.2.2 实验结果

本章节通过实验测得数据样本的平均值、样本无偏方差和以图表方式表示的实测路径记录和数据结果来展现实验结果。其中，

样本平均值：

$$\bar{\mathbf{X}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \quad (4.1)$$

样本方差（无偏）：

$$S^2 = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{X}})^2 \quad (4.2)$$

本课题的实验中样本容量均为 $N = 100$ 。

## (1) 实验一

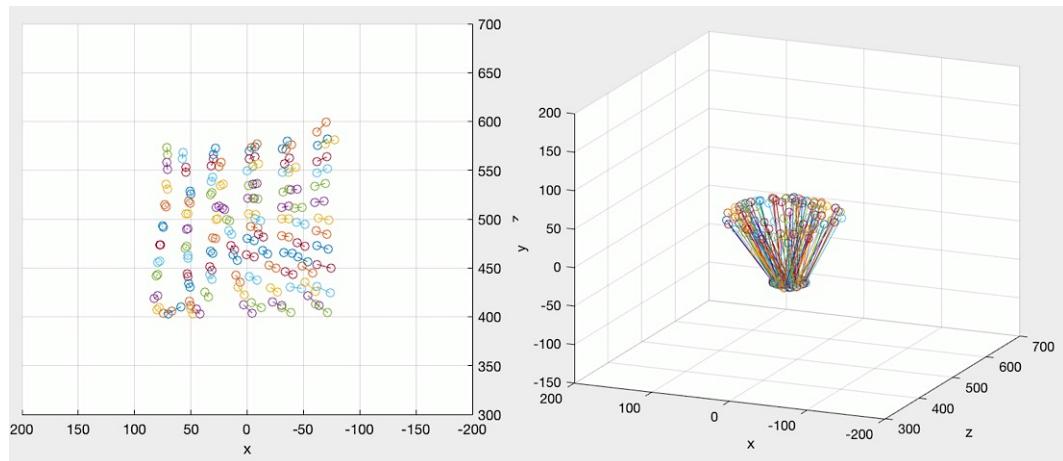


图 4.2 实验一路径记录（左：路径 1；右：路径 2）



图 4.3 实验一测试数据图（左：路径 1；右：路径 2）

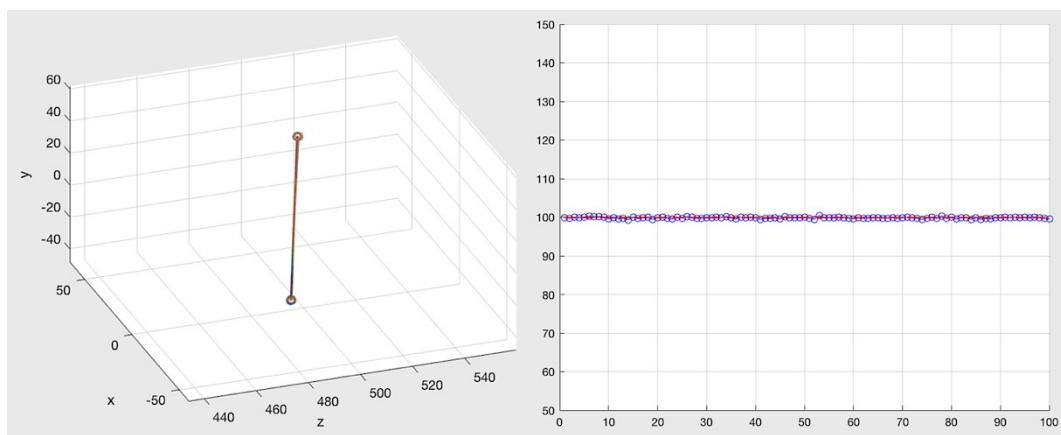
装  
订  
线

图 4.4 对照组实验结果

实际两球间距： $D_{True} = 100\text{mm}$ 。

路径 1：

平均值  $\bar{D}_{1-1} = 100.5924$ ; 样本方差  $S_{1-1}^2 = 13.9918$ 。

路径 2:

平均值  $\bar{D}_{1-2} = 100.2735$ ; 样本方差  $S_{1-2}^2 = 3.6755$ 。

对照组:

平均值  $\bar{D}_{1-C} = 99.8298$ ; 样本方差  $S_{1-C}^2 = 0.0405$ 。

实际路径记录和测试数据请见图 4.2、图 4.3 和图 4.4。

## (2) 实验二

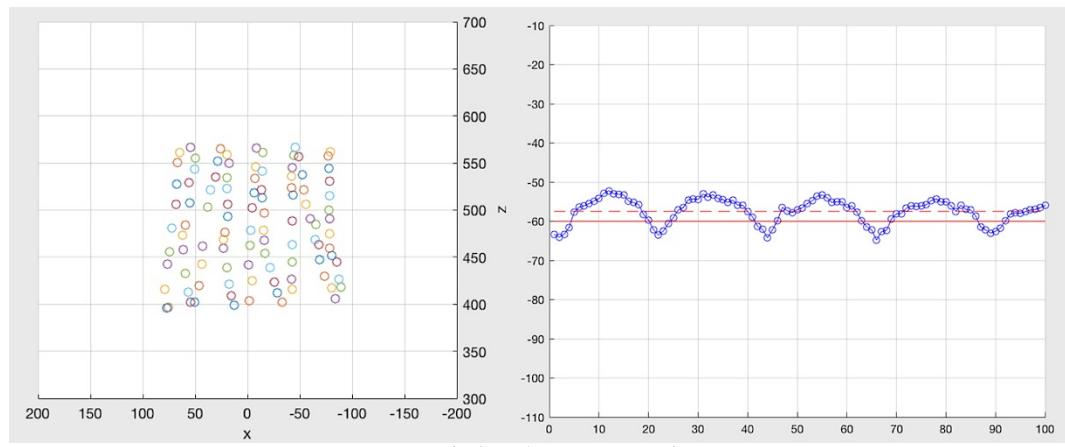


图 4.5 实验二路径记录及测试数据图

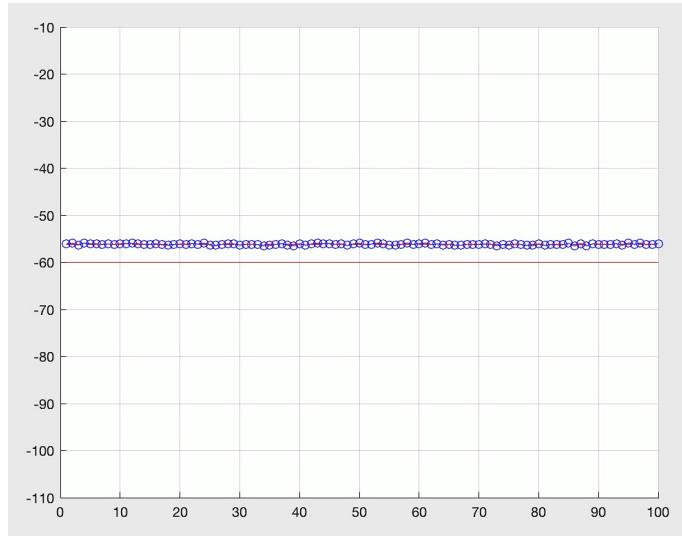


图 4.6 对照组实验结果

路径 1:

平均值  $\bar{Y}_{2-1} = -57.4748$ ; 样本方差  $S_{2-1}^2 = 10.1264$ 。

对照组:

平均值  $\bar{Y}_{2-C} = -56.1725$ ; 样本方差  $S_{2-C}^2 = 0.0225$ 。

实际路径记录和测试数据请见图 4.5 和图 4.6。

#### 4.3 结果分析

根据每记录 100 个数据点所花费的时间，可以得到系统的刷新频率约为 89.5 帧/分钟，即约 1.5Hz。

从两个实验的对照组情况来看，在目标物保持静止的情况下样本方差均比较小，其中  $S_{1-c}^2 = 0.0405$ 、 $S_{2-c}^2 = 0.0225$ ，将它们与其他测量结果的方差相比较，发现系统误差几乎可以忽略。这也说明实验设计和程序本身的稳定性是得到保障的。

从实验 1 的结果来看，从均值的角度，该系统对于目标物之间距离的测量是较为理想的。但是从数据点的监测情况来看，如图 4.3 所示，测量结果有较大波动值。尤其从图 4.3 左的从路径一中收集到的数据来看，波动有较明显的周期性，结合路径一所设计的来回波动路线来看，显然测量结果的误差与目标物出现在拍摄画面的位置有关。从这一点可以确定的是相机拍摄角度的调零存在问题。

值得注意的是，实验 1 的路径二相比于路径一最大的不同在于在路径二中，目标物更多地出现在拍摄画面的中心位置。因而从结果看， $S_{1-1}^2 = 13.9918$  而  $S_{1-2}^2 = 3.6755$ ，路径二下收集的样本方差仅为路径一的 26.3%，显然这其中还有画面边缘处镜头畸变的影响。从图 4.3 右的数据点情况来看，当目标物大致处于画面中心时，距离测量的误差基本可以控制在  $\pm 5\text{mm}$  以内。

从实验 2 的结果来看，样本方差  $S_{2-1}^2 = 10.1264$ ，效果不够理想，且与实验 1 的路径一结果相差不大。与实验 1 的不同之处在于，实验 2 的测量平均值与真实值相差也较大，约有 3 至 4 毫米，该结果进一步说明相机坐标与世界坐标的匹配，也即拍摄位置、俯仰角、滚转角和平偏角的调零存在问题。另外由于测量平面为地平面，整体处于画面中下部，因此也可推测实验结果收到一定镜头畸变的影响。

#### 4.4 系统可优化性讨论和改进方案

从实验结果来看，现有算法的运行基本是有效的，且系统具备了所需功能。但是问题在于提高算法的速度，从而提高程序的运行速度，进而提升测量的频率。

在实际运用中，对于实时测量系统，测量频率是系统有效性的基础，比如对于倒立摆的平衡任务，对于倒立摆的动力学建模可以得到一个对应的最小反馈频率（需要进一步研究确定），若系统的测量频率小于该值，则当系统作出反应前倒立摆已经失去平衡。

从实际实验的结果来看，镜头畸变和镜头标定的准确性是影响三维重建准确性的主要因素，不可忽略。在本研究的实验中，由于时间和场地的限制，镜头的支架和实验挡板都是用日常用品剪裁制作而成，显然其准确度无法得到保障。在本研究中采用的传统标定法的前提下，

无论在测试阶段或是实际运用的过程中，该类基于 CMOS 图像的三维重建系统须运用光学实验中常用到的稳定且方便调整的云台结构作为图像采集器的支架。

而更有效的解决方案是采用更自动化的标定方法，即采用例如章节 2.5 中提到的基于标定板的张氏标定法，或基于图像边缘特征识别的相机自标定法。使用这类非传统的标定方法的优点在于不需要非常准确地对相机位置和拍摄角度进行调整，而是直接运用所拍摄画面进行相机位姿的逆求解，这样不论相机的状态如何，都能准确地定义相机坐标和世界坐标之间的关系。另外，诸如张氏标定法一类的标定法可以直接求解相机镜头的畸变参数，从而对最终结果做出调整，减小畸变可能带来的巨大影响。

实验中发现的另一个能够有效减小畸变影响到方法就是调高图像采集的分辨率。在焦距不变的情况下，在更高分辨率下，所拍摄画面的直接视觉效果就是图像变小、边远，且拍摄范围更大，使得同样位置的目标物比原来更接近画面的中心。然而这样对于计算力支持的要求便大大提高，可能会使得画面变的不流畅，从而使得系统响应滞后。因此，在实际实验中选取了既在所用计算机性能承载范围内，又能够在限定距离下拍摄到足够大范围的分辨率，即 960\*720。

因此在本课题的改进和后续研究中，除了使用标准的光学实验设备外，建议利用 Matlab 中张氏标定法的工具箱对相机分别进行标定，并将镜头畸变纳入考虑。同时，需要计算能力更强大的计算机的支持，以加快系统的响应速度并减小算法的滞后性。

经过以上改进方案后，再结合具体运用情景，这套基于 CMOS 的视觉识别和三维重建系统将可以做到准确、可靠和快速。

装  
订  
线

## 5 结论与展望

### 5.1 课题结论

本研究主要研究了基于 CMOS 的三维视觉技术和应用，其中重点研究了图像预处理的各类方法、类圆识别算法和三维重建理论，并通过所学习的理论知识开发出了一套基于 Matlab 的对于特定目标的三维重建系统，并设计实验对该系统的准确性进行验证。

总体来说该系统具备了所设计的功能并在实验中基本通过了一系列测试。但同时，测试结果也说明了系统前期设计上存在一定的缺陷，导致了测试结果不够理想。本研究通过对数据结果的分析得出造成误差的原因，并进行了总结，以期在后续的研究或应用中解决。

### 5.2 研究前景展望

从第四章中的实验及其结果可以看到，本研究所设计的这套系统最大的误差来源即是传统相机标定法中由于实验设备精度所带来的影响，和相机镜头本身的畸变误差。如前文所述，对于相机位姿的标定前人也已做过许多相关研究，并得出优秀的成果。根据所查找的文献和作者在研究之中的总结，本课题的未来发展和延伸基本可以从以下这几方面切入：

- (1) 进一步提升实效性，即测量频率。这需要在确保完成特定功能的前提下尽量降低算法的复杂度，或为系统配以更加强大的计算能力支持。但是一般对于嵌入式系统和实际应用来说，不论考虑到可行性或是成本，计算能力总是有限的，因此对于算法的优化非常重要。
- (2) 进一步提升准确性。准确性的提升可以从识别算法和三维重建的准确度分别提高。从本研究的结果来看，作者认为三维重建的准确度提高空间巨大，比如使用更稳定的支架、云台、实验台，和利用更好的标定方法以求得更加精准的内外参数。
- (3) 拓宽应用场景。这个问题是目前学界和业界最热点的发展方向，其中需要解决的问题包括如何识别更复杂的目标物（比如手势、人脸、汽车等）、如何去除生活环境下的随机环境噪声（比如烟雾、雨雪天气等）和系统总体的稳定性等等。目前该领域的研究逐渐偏向与各类机器学习算法的结合。

此外，从相机定位衍生出的许多自标定方法和应用领域，如同步定位与地图构建（SLAM 或 Simultaneous localization and mapping）、增强现实技术（AR 或 Augmented Reality）、医疗图像诊断和工况及成品视觉检测等也是当前三维视觉领域中三维重建技术的主要研究热点方向。因此作者认为结合以上三个问题，对本课题进一步分析、研究和改进，是本研究后续工作的努力方向，且可以看到其应用范围广泛，具有广阔的前景。

6 附 录

## 6.1 Matlab 代码

(a) adjustment.m

```

% --- Executes on button press in pushbutton1.

function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

try
    [vidOriginal1]=CallCamera(3,handles.axes1);
    [vidOriginal2]=CallCamera(2,handles.axes3);
    pause(0.05);

catch
    set(handles.edit6,'string','ERROR:...，“ÓÓΣ≤...ºÓ¥Í€Û”');
end

while pi>0
    if get(gcf,'CurrentCharacter')==32
        delete(gcf);
        break;
    end

    [centers1, radii1,frame1]=ImgPreprocessing(vidOriginal1);
    [centers2, radii2,frame2]=ImgPreprocessing(vidOriginal2);
    showBiIm(handles.axes2,centers1, radii1,frame1);
    showBiIm(handles.axes4,centers2, radii2,frame2);
    pause(0.15);

    showAdjPosition(centers1,handles.edit1,handles.edit2);
    showAdjPosition(centers2,handles.edit3,handles.edit4);
    showAdjSolution(centers1,centers2,handles.edit6,handles.edit7)
    clear frame1;
    clear frame2;

end

```

(b) CallCamera.m

```

function [vidOriginal]=CallCamera(numCam,handleAx)

```

装  
订  
线

```
vidOriginal=videoinput('winvideo',numCam,'YUY2_960x720'); %  
'I420_640x480"YUY2_640x480  
camRes=get(vidOriginal,'videoResolution');  
nBands=get(vidOriginal,'NumberOfBands');  
axes(handleAx);  
hImage=imshow(zeros(camRes(2),camRes(1),nBands));  
preview(vidOriginal,hImage);  
end  
  
(c) findIntersection.m  
  
function [xmin,ymin,zmin]=findIntersection(centers1,centers2,p,f)  
% line from camera 1  
positionX1=centers1(1)-480;positionY1=360-centers1(2);  
point1_0=[0,0,0];point1_1=[positionX1*p,positionY1*p,f];  
n1=point1_1 - point1_0;  
% line from camera 1  
positionZ2=centers2(1)-480;positionY2=360-centers2(2);  
point2_0=[500,0,500];point2_1=[500-f,positionY2*p,500+positionZ2*p];  
n2=point2_1 - point2_0;  
% ??????????  
project1=[n1(1),n1(3)];  
project2=[n2(1),n2(3)];  
xzmin=([project1;project2]*[0 -1;1 0])\0;500*(project2(2)-project2(1));  
xmin=xzmin(1);zmin=xzmin(2);  
ymin=zmin*positionY1*p/f;  
end  
  
(d) GUI.m  
  
% --- Executes on button press in pushbutton1.  
function pushbutton1_Callback(hObject, eventdata, handles)  
% hObject handle to pushbutton1 (see GCBO)  
% eventdata reserved - to be defined in a future version of MATLAB  
% handles structure with handles and user data (see GUIDATA)  
clc;  
xBaseArr=zeros(1,100);xBaseI=1;  
disArr=zeros(1,100);disI=1;  
f=4; p=2.8e-3;  
try
```



frame=getsnapshot(vidOriginal);  
I1=frame(:,:,3);  
frame=imbinarize(I1,'global');  
[centers, radii] = imfindcircles(frame,[25 75]);  
end  
(f) Matching.m  
function [xmin,ymin,zmin,ballNum]=Matching(centers1,centers2,p,f,handlesEditState)  
sizeCenters1=size(centers1);sizeCenters2=size(centers2);  
ballNum=sizeCenters1(1);  
xmin=0;ymin=xmin;zmin=xmin;  
if sizeCenters1(1)~= sizeCenters2(1) % «Ú "≤<sup>a</sup> μ»  
    set(handlesEditState,'string','not matched');  
    ballNum=0;  
elseif sizeCenters1(1)==0 % √<sup>a</sup> ” - «Ú  
    set(handlesEditState,'string','none');  
elseif sizeCenters1(1)==1 % “<sup>a</sup> Π^ «Ú  
    set(handlesEditState,'string','one');  
    [xmin,ymin,zmin]=findIntersection(centers1,centers2,p,f);  
elseif sizeCenters1(1)==2 % i Ω Π^ «Ú  
    set(handlesEditState,'string','two');  
    [centers1,centers2]=ReOrder(centers1,centers2);  
    % 2\*lines from camera 1  
    centers1\_1=centers1(1,:);centers1\_2=centers1(2,:);  
    centers2\_1=centers2(1,:);centers2\_2=centers2(2,:);  
    [xmin1,ymin1,zmin1]=findIntersection(centers1\_1,centers2\_1,p,f);  
    [xmin2,ymin2,zmin2]=findIntersection(centers1\_2,centers2\_2,p,f);  
    xmin=[xmin1;xmin2];ymin=[ymin1;ymin2];zmin=[zmin1;zmin2];  
else % ∂‡”Ω Π^ «Ú  
    set(handlesEditState,'string','more than two');  
end  
% xmin=1.5\*xmin;ymin=1.5\*ymin;zmin=1.5\*zmin;  
end  
(g) plotExpArr.m  
function plotExpArr(Arr,I,true)  
a=1:I;  
Arr=Arr(a);

装订线

Mean=mean(Arr).\*ones(1,I);  
True=true.\*ones(1,I);  
figure();grid on;hold on;  
axis ([0 100 true-50 true+50]);  
plot(a,Arr,'bo-');  
plot(a,True,'r-');  
plot(a,Mean,'r--');  
Xmean=mean(Arr);S2=sum((Mean-Arr).^2)/(I-1);  
Xmean  
S2  
end  
(h) plotIn3D.m  
function [dis]=plotIn3D(xmin,ymin,zmin,ballNum,handleResult)  
ballNumStr=num2str(ballNum);  
switch ballNumStr  
case '1'  
how='o';  
case '2'  
how='o-';  
end  
figure(1)  
grid on;  
hold on;  
plot3(zmin,-xmin,ymin,how);  
axis ([300 700 -200 200 -150 200]);  
view(-20,40);  
xlabel('z');ylabel('x');zlabel('y');  
% æ»"È·◊±Í  
if ballNum==1  
str3Dx=num2str(xmin);str3Dy=num2str(ymin);str3Dz=num2str(zmin);  
str3D=[str3Dx,' ',str3Dy,' ',str3Dz];  
set(handleResult,'string',str3D);  
dis=0;  
elseif ballNum==2  
dis=((xmin(2)-xmin(1))^2+(ymin(2)-ymin(1))^2+(zmin(2)-zmin(1))^2)^0.5;  
strDis=num2str(dis);

```
strDis=['The distance between 2 balls = ',strDis];
set(handleResult,'string',strDis);

end
end

(i) ReOrder.m

function [centers1,centers2]=ReOrder(centers1,centers2)
if centers1(1,2)<=centers1(2,2)
    temp1=centers1(1,:);temp2=centers1(2,:);
    centers1=[temp2,temp1];
end
if centers2(1,2)<=centers2(2,2)
    temp1=centers2(1,:);temp2=centers2(2,:);
    centers2=[temp2,temp1];
end
end

(j) showAdjPosition.m

function showAdjPosition(centers,handleX,handleY)
sizeCenters=size(centers);ballNum=sizeCenters(1);
if ballNum==1
    positionX=centers(1)-480;
    positionY=360-centers(2);
    strPositionX=num2str(positionX);
    strPositionY=num2str(positionY);
    set(handleX,'string',strPositionX);
    set(handleY,'string',strPositionY);
elseif ballNum==2
    set(handleX,'string','/');
    difPositionY=abs(centers(2,2)-centers(1,2));
    strdifPositionY=num2str(difPositionY);
    set(handleY,'string',strdifPositionY);
else
    set(handleX,'string','error');
    set(handleY,'string','error');
end
end

(k) showAdjSolution.m
```

```
function showAdjSolution(centers1,centers2,handleEditCam1,handleEditCam2)
sizeCenters1=size(centers1);sizeCenters2=size(centers2);
ballNumCam1=sizeCenters1(1);ballNumCam2=sizeCenters2(1);
% Cam1
if ballNumCam1==0
    set(handleEditCam1,'string','No ball detected.');
elseif ballNumCam1==1
    if centers1(1)-480>0
        AdjCam1X='Turn right;';
    else
        AdjCam1X='Turn left;';
    end
    if 360-centers1(2)>0
        AdjCam1Y='Turn up;';
    else
        AdjCam1Y='Turn down;';
    end
    AdjCam1=[AdjCam1X,'',AdjCam1Y];
    set(handleEditCam1,'string',AdjCam1);
elseif ballNumCam1==2
    if centers1(1,1)>centers1(2,1)
        temp1=centers1(1,:);temp2=centers1(2,:);
        centers1(1,:)=temp2;centers1(2,:)=temp1;
    end
    if centers1(1,2)<centers1(2,2)
        set(handleEditCam1,'string','Turn clockwise.');
    else
        set(handleEditCam1,'string','Turn anti-clockwise.');
    end
    else
        set(handleEditCam1,'string','More than 2 balls detected.');
    end
% Cam2
if ballNumCam2==0
    set(handleEditCam2,'string','No ball detected.');
elseif ballNumCam2==1
```

装  
订  
线

```
if centers2(1)-480>0
    AdjCam2X='Turn right;';
else
    AdjCam2X='Turn left;';
end
if 360-centers2(2)>0
    AdjCam2Y='Turn up;';
else
    AdjCam2Y='Turn down;';
end
AdjCam2=[AdjCam2X,'',AdjCam2Y];
set(handleEditCam2,'string',AdjCam2);
elseif ballNumCam2==2
    if centers2(1,1)>centers2(2,1)
        temp1=centers2(1,:);temp2=centers2(2,:);
        centers2(1,:)=temp2;centers2(2,:)=temp1;
    end
    if centers2(1,2)<centers2(2,2)
        set(handleEditCam2,'string','Turn clockwise.');
    else
        set(handleEditCam2,'string','Turn anti-clockwise.');
    end
else
    set(handleEditCam2,'string','More than 2 balls detected.');
end
end

(1) showBiIm.m
function showBiIm(handleAx,centers,radii)frame)
axes(handleAx);
imshow(frame);
viscircles(centers, radii,'EdgeColor','b');
end
```

## 6. 2 代码的简单实用指导

将上一章节的所有 m 文件置于同一个文件夹中，并调整 matlab 的根目录至该文件夹。要运行摄像头位姿调零程序，则运行 adjustment.m 文件；要运行主程序，则运行 GUI.m 文件，并根据所需进行的实验调整源代码。

另外，上一章节中 adjustment.m 文件和 GUI.m 文件由于其他不相关代码太过于冗长，故只给出了一段相关代码，且实际运行时，必须配合 adjustment.fig 和 GUI.fig 文件，即他们各自所对应的图形用户界面设计一起实用。

装

订

线

## 参考文献

- [1] Dana H. Ballard, Christopher M. Brown. Computer Vision. Prentice Hall, 1982
- [2] 尚明珠,解凯. 几种三维重建方法的比较[J]. 哈尔滨师范大学自然科学学报,2003,05:43-46.
- [3] BBC 中文网. 美国调查特斯拉自动驾驶汽车致死车祸[EB/OL]. (2016-7-1) [2017-4-7].  
[http://www.bbc.com/zhongwen/simp/science/2016/07/160701\\_tesla\\_crash\\_investigation](http://www.bbc.com/zhongwen/simp/science/2016/07/160701_tesla_crash_investigation)
- [4] Wicked Sago. What is CMOS Memory? [EB/OL]. (2013-3-3)[2017-4-7].  
<http://wickedsgo.blogspot.com/2011/04/what-is-cmos-memory.html>
- [5] Pierre Magnan. Detection of visible photons in CCD and CMOS: a comparative view[J]. Nuclear Instruments and Methods in Physics Research A, 2003, 504: 199-212.
- [6] El Gamal A., Eltoukhy H.. CMOS image sensors[J]. Circuits and Devices Magazine, IEEE, 2005,21(3): 6-20.
- [7] 李继军,杜云刚,张丽华,刘全龙,陈建芮. CMOS 图像传感器的研究进展[J]. 激光与光电子学进展,2009,04:45-52.
- [8] 金宝智.图像传感器：CCD 与 CMOS 的对比[J].现代电视技术,2005(05): 80-81.
- [9] 熊平. CCD 与 CMOS 图像传感器特点比较[J]. 半导体光电,2004,(01):1-4+42.
- [10] 杰瑞. CCD 与 CMOS 传感器的比较[J]. 电器评价,2005,(05):12-13.
- [11] 董博彦.COMS 图像传感器的测试与分析[D].天津大学:2005
- [12] 石东新,傅新宇,张远. CMOS 与 CCD 性能及高清应用比较[J]. 通信技术,2010,(12):174-176+179.
- [13] D. Marr, et al. A theory of human Stereo disparity [J]. Science. 1976, 194: 283-287.
- [14] 李慧霞. 室内智能移动机器人规则物体识别与抓取[D].北京交通大学,2016.
- [15] Boyoon Jung, Gaurav S. Sukhatme. Real-Time Motion Tracking from a Mobile Robot [J]. International Journal of Social Robotics, 2010, 2(1): 63-78.
- [16] Ratheesh Kalarot, Georgy Gimel, John Mirris. 3D Object Tracking with a High-Resolution GPU Based Real-Time Stereo [C]. Proceedings of the 27<sup>th</sup> Conference on Image and Vision Computing, 2012: 394-399.
- [17] Zhen Jia, Arjuna Balasuriya, Subhash Challa. Sensor Fusion-Based Visual Target Tracking for Autonomous Vehicles with the Out-of-Sequence Measurements Solution[J]. Robotics and Autonomous Systems, 2008, 56(2): 157-176.
- [18] 张峰,史利民,孙凤梅,胡占义,一种基于图像的室内大场景自动三维重建系统.自动化学报,2010,5(36):625-632.
- [19] 蒋焕煜, 彭永石, 应义斌. 双目立体视觉关键技术在果蔬采摘机器人中的应用[J]. 江苏大学学报(自然科学版), 2008, 29(5): 58-66.

装  
订  
线

装  
订  
线

- [20] Suolin Duan, Yunfeng Li, Shuyue Chen, et al. Research on Obstacle Avoidance for Mobile Robot Based on Binocular Stereo Vision and Infrared Ranging[C]. Proceedings of 9th World Congress on Intelligent Control and Automation (WCICA), 2011: 1024-1028.
- [21] 杜宇. 三维重建中双目立体视觉关键技术的研究[D]. 哈尔滨理工大学, 2014.
- [22] Liping Zheng, Guangyao Li, Jing Sha. The survey of medical image 3D reconstruction [C]. Fifth International Conference on Photonics and Imaging in Biology and Medicine, 2007.
- [23] John C. Carr, R. K. Beatson. Reconstruction and representation of 3d objects with radial basis functions [C]. 28th Annual Conference on Computer Graphics and Interactive Techniques, 2001: 67-76.
- [24] Markus Hehn, Raffaello D'Andrea. A Flying Inverted Pendulum [J]. IEEE International Conference on Robotics and Automation, 2011: 763-77.
- [25] Guillaume Ducard, Raffaello D'Andrea. Autonomous Quadrotor Flight Using a Vision System and Accommodating Frames Misalignment [J], IEEE International Symposium on Industrial Embedded Systems, 2009: 261-264.
- [26] 陈若珠, 武艺玄. 基于车牌识别的图像预处理研究[J]. 工业仪表与自动化装置, 2015 (4): 40-43.
- [27] 谭振坤, 冯登超, 陈刚, 等. 医学超声病灶图像预处理[J]. 国外电子测量技术, 2014 (3): 89-91.
- [28] R. W. G. Hunt. The Reproduction of Colour (6th ed.) [M]. Chichester UK: Wiley-IS&T Series in Imaging Science and Technology, 2004
- [29] Wikipedia. Grayscale[DB/OL]. (2015-5-8)[2017-6-10]. <https://en.wikipedia.org/wiki/Grayscale>
- [30] Wikipedia. CIE 1931 color space[DB/OL]. (2017-6-1)[2017-6-10].  
[https://en.wikipedia.org/wiki/CIE\\_1931\\_color\\_space](https://en.wikipedia.org/wiki/CIE_1931_color_space)
- [31] Wikipedia. Gamma correction[DB/OL]. (2017-6-6)[2017-6-10].  
[https://en.wikipedia.org/wiki/Gamma\\_correction](https://en.wikipedia.org/wiki/Gamma_correction)
- [32] 梁一江. 图像平滑处理方法初探及简单的算法介绍[J]. 才智, 2009, 4: 113.
- [33] 韩宝立, 刘杨涛. 图像平滑算法的研究与改进[J]. 电脑知识与技术, 2009, 30: 082.
- [34] 魏有法. 基于 Matlab 的图像平滑算法浅析[J]. 机电技术, 2013, 36(3): 50-52.
- [35] 李慧霞, 高梓豪. 室内智能移动机器人规则物体识别与抓取[J]. 软件, 2016 (2): 89-92.
- [36] 王艳芳. 类圆识别技术的研究[D]. 北方工业大学, 2006.
- [37] H.K Yuen, J. Prince, J. Illingworth, and J. Kittler. "Comparative study of Hough transform methods for circle finding[J]." Image and Vision Computing. Volume 8, Number 1, 1990, pp. 71-77.
- [38] 杜宇. 三维重建中双目立体视觉关键技术的研究[D]. 哈尔滨理工大学, 2014.
- [39] 李慧霞, 高梓豪. 室内智能移动机器人规则物体识别与抓取[J]. 软件, 2016 (2): 89-92.

- [40] Weng J, Cohen P, Herniou M. Camera calibration with distortion models and accuracy evaluation[J]. IEEE Transactions on pattern analysis and machine intelligence, 1992, 14(10): 965-980.
- [41] Tsai R. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses[J]. IEEE Journal on Robotics and Automation, 1987, 3(4): 323-344.
- [42] Faugeras O D, Luong Q T, Maybank S J. Camera self-calibration: Theory and experiments[C]//European conference on computer vision. Springer Berlin Heidelberg, 1992: 321-334.
- [43] Logitech. LOGITECH HD WEBCAM C270 TECHNICAL SPECIFICATIONS[EB/OL]. [2017-6-10]. [http://support.logitech.com/en\\_us/article/17556](http://support.logitech.com/en_us/article/17556)

装

订

线

## 辞 谢

四年的本科生涯转眼就来到了尾声。尽管大四学年作者在位于意大利米兰的 Pelitecnico di Milano 交流学习，且在临近毕业前学业生活非常繁忙，但仍然在前后两个月这样有限的时间中完成了同济的这篇毕业设计。在此要感谢朱传敏教授远在万里之外的沟通与指导。同样，要感谢朱传敏教授开放式的选题过程，让本人得以对自己感兴趣的课题做研究。通过本课题的研究，作者对于计算机三维视觉这样一个原本比较陌生的领域有了一个较为粗浅的了解，锻炼了自主学习并运用新知识的能力，也为未来在“系统控制和自动化工程”专业研究生的学习之路，和机器人算法领域的研究上开了个好头。

装  
订  
线