

西北大学

硕士学位论文

基于运动背景下运动目标的检测与自动跟踪

姓名：贺琴

申请学位级别：硕士

专业：电路与系统

指导教师：潘建寿

20100610

摘要

考虑到现在一般的检测与跟踪系统都需要人为辅助操作的问题,本文介绍一种自动监控系统。本文主要涉及分两个部分:一、通过一些图像处理算法检测出运动背景下目标在图像中的位置;二、通过 PC 机控制云台运动,进而带动摄像头转动完成对运动目标自动跟踪。

基于运动背景下运动目标的检测主要完成以下两方面工作:一、全局运动补偿,为了解决摄像机运动导致背景发生大幅度,本文提出了一种具有鲁棒性全局运动背景匹配法完成动态背景的静态转化;二、比较几种常用的运动目标检测算法的优缺点,提出适合本系统的检测算法——三帧差分法。

在摄像机和运动物体同时运动情况下,一般的运动目标跟踪算法存在着被动跟踪的滞后问题。为了解决此问题,本文提出了一种结合 Kalman 滤波器的 Mean-Shift 自动跟踪算法。首先利用 Kalman 滤波器算法进行运动估计,预测物体在下一帧中的位置,根据预知的位置向云台发出相应的指令控制云台运动,进而带动摄像机转动相应角度,然后在此基础上利用 Mean-Shift 搜索算法精确的确定目标位置,与此同时本文又引入相似性系数来判定目标是否发生遮挡。如此循环。

最后自主搭建了单通道视频监控系统实验平台进行仿真实验,验证了系统的有效性。

关键词

全局运动估计、背景匹配、运动目标检测、运动目标自动跟踪、摄像机控制

Abstract

Considering the detection and tracking system generally need the artificially support, the paper researchers a new fully automated monitoring system. The paper mainly involves two parts: (1) detect the moving targets through some image processing algorithms; (2) control the camera to automatic track the moving target with the combination of the PC and the PTZ.

In order to detect moving targets in the moving background, two aspects are needed to be done: firstly, match the background. The paper propose a robust background matching algorithm to complete the background's static conversion; secondly, introduce a detection algorithm -- the three frames subtraction to detect the moving targets in the static background.

There is a problem of tracking lag in the usually passive tracking algorithms, when the camera and the mobile move at the same time. In order to solve this problem, the paper combines the kalman filter algorithms and the Mean-Shift algorithms to complete the auto-tracking. Firstly, the kalman filter algorithms, which can initiatively the position of the object in the next frame is introduced to estimate the motion. According to the location of the next frame, the PC issue corresponding instructions to the PTZ. The PTZ controls the camera rotation. Next, determine a precise target location of the current frame using the Mean-Shift algorithms. Simultaneously, the Bhattacharyya similarity coefficient is introduced to determine whether the mobile has been obscured. This cycle continues.

Based on the above ideas, a single-channel video surveillance system is set up under an experimental environment for programming and simulation after some algorithms are improved, integrated and optimized in this paper. Experiments show that the entire system achieves the desired results.

Key words

Global motion estimation, background matching, moving target detection, moving target auto-tracking, camera controlling

西北大学学位论文知识产权声明书

本人完全了解西北大学关于收集、保存、使用学位论文的规定。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版。本人允许论文被查阅和借阅。本人授权西北大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。同时授权中国科学技术信息研究所等机构将本学位论文收录到《中国学位论文全文数据库》或其它相关数据库。

保密论文待解密后适用本声明。

学位论文作者签名： 贺琴 指导教师签名： 潘建群

2010年6月10日 2010年6月10日

西北大学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。据我所知，除了文中特别加以标注和致谢的地方外，本论文不包含其他人已经发表或撰写过的研究成果，也不包含为获得西北大学或其它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： 贺琴

2010年6月10日

第一章 绪论

1.1 课题的研究背景和意义

为使人们的生活更加便利，人们不断发明、创造各类智能机器来辅助、甚至代替人类完成某些任务。例如：汽车的发明不仅代替了人们的脚力，也缩短了我们的行程距离；电脑的产生减轻了人们脑力劳动和提高了处理效率；起重机械强化了人的臂力等等。而人类通过视觉感知外界的信息量占有所有感觉器感知的信息量高达 80%^[1]。所以人们在不断地尝试着使人类的视觉系统功能赋予机器化的梦想。而计算机视觉就是人们开始尝试利用计算机与摄像机的结合代理人们天然的视觉系统的一门新兴学科。这门学科主要是利用摄像机将世界中的三维物体通过视觉系统成像为二维图像并转化为数字信号，再让计算机以数字图像的方法来处理这些数字信息，这就成功的实现了对人类视觉系统的模拟。

计算机视觉是一门边缘性学科，它涉及到很多学科的知识，如：图像处理、模式识别、自动控制、计算机、数学、人工智能、神经网络等等。这门学科的研究始于 19 世纪 60 年代，直至 1982 年，Marr 提出著名的视觉理论^[2]。Marr 比较具体阐述了有关人类视觉理论和计算机视觉研究的基本思想。他认为人类视觉对视觉信息或图像的处理过程中包括了许多有条理和严格的理论。这个较复杂处理过程可以使人们能够从视网膜恢复出物理世界。根据他的基本思想，Marr 也提出一些相应的解决方案。尽管最后 Marr 解决方案被否定，但是他的研究却为科学界开辟了一个崭新的科学研究天地——计算机视觉研究领域。近 30 年来，计算机视觉方面的研究取得较大进步。现在其很多研究成果已经广泛投入各行各业的应用中，例如：智能导航、智能监控、医疗诊断等。

视频监控就是计算机视觉的一个重要研究方向。现在它已经广泛的应用在工地、厂矿、小区、交通道路、银行、仓库、机场、车站等各类场所已完成自动监控、安全监控。视觉监视的广泛应用领域决定了它具有较强的研究价值。但是，现代人们对监控的要求越来越高，不再仅仅停留在人工辅助操作的可视化水平，本文旨在通过计算机自动控制摄像头识别跟踪运动目标从而减轻人的负担，实现监控的智能化。

对于一个视频监视系统，运动物体是研究的对象。因此一个完整的视频监控系统的主要包括四部分^[3]：(1) 运动物体检测；(2) 运动物体跟踪；(3) 运动物体分类；(4) 主动识别。而其中运动物体的检测与跟踪是整个监控系统的基本，其直接决定后续高级处理。

所以多年来,运动目标的检测和跟踪的研究备受关注,不断的探索与创新。而运动背景下运动物体的检测与跟踪的研究也成为计算机视觉领域中的研究热点之一。

本文所研究的基于单目视觉运动背景下的运动目标检测与自动跟踪就是视频智能监控系统的研究工作的基础部分。

1.2 课题的研究现状

运动目标检测与跟踪这两方面存在一种承接关系。运动目标的检测是运动目标跟踪的前基础与前提,它负责完成在被监控视场中运动目标是否出现的判断和在图像的具体坐标的提取。而运动目标跟踪一般是根据已检测出的目标特征,再采用适当的搜索算法找到目标在下一帧的位置。

一般,依照被监视的场景与摄像机之间是否存在相对运动,人们将视频处理分为静态背景下运动目标检测和运动背景下运动目标检测两大类。所谓静态背景运动目标检测是指运动场景中的运动只存在实体运动,而背景没有或者只有微小的变化。而运动背景下运动目标检测是指运动图像中的运动由摄像机的运动和实体运动共同运动产生的。自从 19 世纪 60 年代计算机视觉起源后,有关静态背景下运动目标检测下已经发展的比较成熟了。而运动背景下的视频处理研究的相对还是比较少。静态背景运动目标的检测现在已经有多种比较成熟的算法,典型的是:帧间差分法^[4]、光流法^[5]和背景差分法^[6]。这些方法均各自有其自身的特点和缺点^[7]。光流法可以在不知视场任何信息的条件下,成功检测出独立运动目标,可是大多数光流法的计算复杂,很难满足检测的实时性。帧间差分法用连续的二到三帧图像相减实现时域差分。帧间差分法尽管对动态变化的场景非常有效,但这种方法有赖于运动目标的运动速度,非常容易造成“多目标”或“目标分裂”等现象。背景差分法能完整地分割出运动目标,而且最容易实现,但是这种方法存在背景的获得与不断更新的问题。

上面提及的几种静态背景下的运动目标检测方法不能直接应用于对摄像机和运动目标同时发生运动的情况。为了解决这方面的需求,需要寻找新的解决办法。2005 年,美国中央佛罗里达大学计算机视觉实验室开发了基于 MATLAB 的 COCOA 系统,用于处理无人驾驶飞机低空航拍视频图像的目标检测与跟踪问题^[8]。该系统实现了运动背景下的目标检测,其中就采用了全局运动估计和运动补偿解决了运动背景的静态转化,最核心的是全局运动参数估计模块,但全局运动估计存在算法计算量大和构成视频场景的各个运动对象引入了参数估计偏差的两大主要困难。为了解决这些,很多学者做了很多

的研究。对于第一个问题，Konrad 提出的全局运动估计算法采用 Levenberg-Marquadt(LM)方法^[9]，用残差直方图方法去除噪声，但这种算法计算量大还是比较大，而且计算结果对噪声敏感，依旧难做到实时。Amir Averbuch 和 Yosi Keller 改进基于梯度的运动估计法^[10]，使计算量减少约二十几倍。还有 G Sorwar 基于运动前景通常分布在图像中间的假设^[11]，采用图像边缘的宏块运动矢量来估计全局运动参数。从而提高计算速度和排除前景中运动对象的干扰。

运动目标的跟踪近年来一直是人们备受关注的多学科交叉课题。国外研究比国内研究起步相对较早，并已经取得了一定成果。其过程就是选取运动目标的有效特征，在后续帧中搜索与该描述的特征进行最匹配的目标位置的过程。现在，比较常用的运动目标跟踪算法包括基于 3-D 模型的方法、基于特征的方法、基于活动轮廓模型的方法等^[12,13,14]。随着现代社会监控难度的增加，越来越多的领域对传统意义上的视频监控提出了更高的要求。因此，新一代的智能视觉监控技术得到了日益广泛的关注。即使其更加智能化，能够在无须人为干预的情况下，如：VSAM 视频监控系统^[15,16]。此系统由美国国防计划局支助的卡内基梅隆大学机器人研究所，另外还有联合一些著名企业于1997年至1999年间开发的一个典型智能监控系统。这个系统只需要一个操作员就可以完成较大范围场景的监控。其具有以下功能：

(1)较完整的视频分析技术：具有对运动目标检测（航拍或地面视频图像中人、动物和车辆）、分析和识别。

(2)运动目标的准确定位：通过光线交叉完成对目标的定位。

(3)场景的建模：可以利用无人直升机得到地形的高程数据、球形图像镶嵌和全景图像镶嵌、虚拟现实模型和 3D 可视化。

(4)运动目标的行为分析：通过运动目标的运动轨迹绘制完成单一目标建模，人车交互等。

(5)传感器协同工作：空中运动摄像机与地面静止摄像机的多传感器协同工作。

(6)人机交互：具有很好的图形用户界面。

目前，这个智能系统还处于试用阶段。除此之外，我国国内也有相关的研究。如：智能控制系统研究院和南加州大学机器人研究所也联合研发了一个对航拍图像序列的动态场景进行分析的智能系统。该系统主要可以完成两个功能：(1)对图像序列中的运动物体完成实时的检测和自动跟踪；(2)完成图像镶嵌，提取运动物体轨迹，分析运动物体间的运动关系。但是由于众多变化因素的存在，使得系统的自动跟踪方面在实际应

用中不具有广泛性。影响运动目标跟踪的因素主要因素有：目标种类繁多的问题（目标的大与小、目标是刚性与非刚性、独立目标与多个目标等）；运动目标存在冲突、遮挡现象；运动目标的先验知识难于获取的问题；匹配模板需要具有自适应性；目标运动位置预测的准确度等。这些问题仍需要国内外的学者做深入研究。

1.3 论文的结构安排及其主要贡献

1.3.1 论文的结构安排

本文将运动背景匹配、运动目标检测、运动目标跟踪作为既承接又独立的几部分来完成。本文的组织结构和各章内容如下：

第一章：绪论。主要阐述了课题的研究背景和意义，简单介绍了当前国内外关于基于单目视觉目标检测和跟踪的研究现状，具体介绍了组织结构和各章内容。

第二章：全局运动补偿。介绍了运动参数模型的建立、特征点的提取与匹配；然后将鲁棒统计分析法和 LMA 优化算法结合的应用到仿射变换模型的参数估计中，完成全局运动参量估计；最后根据估计出的全局运动参量，采用双线性内插法完成对背景匹配，进而成功实现了动态背景的静态转化。

第三章：运动目标的检测。研究了运动目标检测的一般过程，提出适合本系统的检查方法——三帧差分法。为了准确和完成的提取出运动目标，在本章中又对检测到目标进行了二值化、形态学滤波、连通性分析等后续处理，最后采用形心法完成对运动目标的定位。

第四章：运动目标的跟踪。重点研究了均值平移跟踪算法，对于经典的均值平移跟踪算法的问题（当背景运动较大时，如果使得当前的目标中心超出了前一帧的目标跟踪窗，则会造成目标丢失）进行了改进。接着又提出了 Kalman 算法完成对下一帧目标的位置估计，实现对目标运动估计。然后进行遮挡情况的分析处理。

第五章：系统硬件的搭建和软件的实现。文根据本实验的需要，组建了一个单通道视频监控系统。

最后是结论：总结了全文的所做的工作与研究成果，指出研究中不足之处和展望继续的研究工作。

1.3.2 主要贡献

(1) 为了解决摄像机运动造成的背景发生大幅度的全局运动现象，本文采用一种具

有鲁棒性的基于特征点的全局运动背景匹配算法。利用 Harris 算法提取出特征点后,提出一种改进型的、快速的圆形模板匹配算法(SSAD),完成特征点的匹配。这种方法不仅可以处理相邻帧间除了平移以外更大幅度旋转、缩放的全局运动,而且相对于传统的穷尽搜索法降低了计算量。考虑外点的存在对全局运动估计的影响,本文将鲁棒统计分析和 LMA 优化算法结合的应用到仿射变换模型的参数估计中,成功的去除了外点,进而提高全局运动参量的估计精度。

(2) 在运动目标的检测部分,比较了几种常见的运动目标的检测算法,并重点研究了三帧差分算法。

(3) 提出了一种结合 Kalman 滤波器的 MeanShift 自动跟踪算法。由于摄像机与运动目标同时运动,仅仅使用 MeanShift 跟踪算法,会造成跟踪滞后问题。因此本文引入 Kalman 滤波器算法,根据目标的速度信息主动预测物体在下一帧的位置,以便于摄像头及时调整角度,快速跟踪目标,消除了跟踪滞后现象。

(4) 自主搭建了一个基于 Marox 图像采集卡的视频图像采集的试验平台,验证了本文所提出的算法的可行性。

第二章 全局运动背景补偿

摄像系统目前被广泛应用于自动导航、运动目标的检测与跟踪、视频压缩和全景图像拼接,要求输出图像具有较好的稳定性和较高的实时性。但往往由于摄像机本身的运动导致输出的图像序列不稳定,会极大地影响后续处理。针对此问题国内外开始研究由于摄像机位置或参数的变化而引起的整个图像变化的补偿。绝大部分方法都遵循如下两个步骤:首先是全局运动参量估计,然后是背景匹配。这两部分存在承接关系。其中全局运动估计是背景匹配的关键,它主要涉及运动参数模型的建立和全局运动参量的估计。其框架如图 2.1。

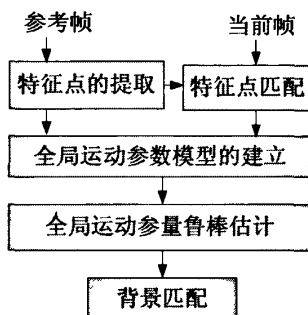


图 2.1 使用参数模型的全局运动估计功能框图

鉴于框图结构,本章的内容安排如下:在 2.1 节介绍全局运动参数模型的建立;在 2.2 节讨论全局运动参量的鲁棒估计;在 2.3 节阐述了特征点的提取与匹配;在 2.4 节介绍运动背景的匹配;在 2.5 节通过两组实验室证明本章所提及算法的可行性;在 2.6 节总结本章的工作及其在全文中的作用。

2.1 全局运动参数模型的建立

在场景成像时,摄像机的运动会使图像序列中像素点的坐标从 $k-1$ 帧的 (x_{k-1}, y_{k-1}) 处移动到 k 帧的 (x_k, y_k) 处。这种坐标变换量满足一定运动变换模型。即, $(x_{k-1}, y_{k-1}) = f(x_k, y_k)$ 。不同的运动会涉及不同个数的参数模型。因此寻找前后两帧中对应像素点的坐标变换量,首要解决问题的是确定运动参数模型。下面将介绍几种最常见的运动参数模型:二参数模型法^[17]、四参数模型法^[18]、八参数模型法^[19]等。

1、平移模型

平移模型涉及两个参数,如式(2.1):

$$\begin{cases} x_{k-1} = f(x_k, y_k) = x_k + c \\ y_{k-1} = f(x_k, y_k) = y_k + d \end{cases} \quad (2.1)$$

其中, 参量 (c, d) 表示图像沿着 x 轴和 y 轴方向的平移量, 如图 2.2 所示。这种简单的运动参数模型在 MPEG-1、MPEG-4 和 H.263^[20] 等几种国际图像标准中已经被应用。但由于这种运动模型的运动矢量与实际运动存在着差异, 所以利用此模型完成全局运动参量的估计会带来比较大的误差。

2、仿射运动参数模型

仿射运动模型涉及六个参数, 如式(2.2):

$$\begin{aligned} \begin{pmatrix} x_{k-1} \\ y_{k-1} \end{pmatrix} &= \gamma \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \begin{pmatrix} c \\ d \end{pmatrix} \\ &= \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \begin{pmatrix} c \\ d \end{pmatrix} \end{aligned} \quad (2.2)$$

其中, 参数 γ 为缩放运动参数, 参数 θ 为旋转角度, 参数 (c, d) 为沿 x 轴和 y 轴方向的平移运动参数。仿射运动参数模型不仅可以处理二维图像中三角形任意变形到另外一种形状的三角形, 也可处理长方形二维变形到平行四边形, 还可处理平移、旋转、伸缩运动。如图 2.2 所示。

3、投影运动参数模型

投影运动参数模型是一种常用的变换模型, 其涉及八个参数, 如式(2.3)。

$$\begin{cases} x_{k-1} = \frac{a_1 \cdot x_k + b_1 \cdot y_k + c_1}{\alpha \cdot x_k + \beta \cdot y_k + 1} \\ y_{k-1} = \frac{a_2 \cdot x_k + b_2 \cdot y_k + c_2}{\alpha \cdot x_k + \beta \cdot y_k + 1} \end{cases} \quad (2.3)$$

投影运动参数模型可处理矩形到任意四边形的形变, 也可以处理旋转, 如图 2.2 所示。

4、双线性运动参数模型

双线性运动参数模型共有十参数, 如式(2.4)。

$$\begin{aligned} x_{k-1} &= a_1 \cdot x_k + b_1 \cdot y_k + c_1 \cdot x_k \cdot y_k + d_1 \\ y_{k-1} &= a_2 \cdot x_k + b_2 \cdot y_k + c_2 \cdot x_k \cdot y_k + d_2 \end{aligned} \quad (2.4)$$

这种运动参数模型可以处理平行四边形任意形变。如图 2.2 所示。

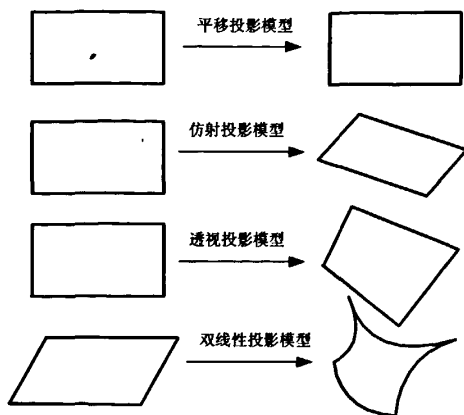


图 2.2 不同运动模型可描述的形变

摄像机运动引起的相邻两帧之间背景变化是一种慢速运动。这种运动主要是由于背景围绕轴的旋转或沿轴的平移引起的。因此在建立运动参数模型只需要考虑图像的旋转和平移即可。通过对上面那几种不同运动模型介绍，知道仿射运动参数模型可以处理长方形二维变形到平行四边形，也可处理二维图像中三角形任意变形到另外一种形状的三角形，还可处理旋转、平移和伸缩运动。因此，利用仿射运动参数模型就可以很好的表示由摄像机运动引起的背景发生的各种的运动。同样，投影运动参数模型和双线性运动参数模型也能很好地描述背景运动，得到比较好的补偿效果。但是涉及的参数越多，就会加大计算量。例如：利用投影运动参数模型在完成全局运动估计时至少需要 4 个像素点，且其中任何 3 个像素点不共线。而仿射模型则只需要 3 个不共线的像素点就可以完成。所以根据本文摄像机运动的实际情况、全局运动估计效果以及算法的精度三方面考虑，仿射模型是一种“折中”模型。

关于如何可以准确的估计出 γ 、 θ 、 c 、 d 这四个全局运动参数，人们研究了很多种算法，而使式(2.5)中的目标能量函数值 E 能否取得一个最小值是判断准则^[21,22,23]。

$$E = \sum_{x_{k-1}, y_{k-1}} e(x_{k-1}, y_{k-1})^2 = \sum_{x_{k-1}, y_{k-1}} [I_{k-1}(x_{k-1}, y_{k-1}) - I_k(x_k, y_k)]^2 \quad (2.5)$$

其中， I_{k-1} 、 I_k 分别为参考帧、补偿后的当前帧的灰度值。

基于此思想，本文将接下来几节具体阐述如何实现对全局运动参数的估计。

2.2 特征点提取与匹配

通常，全局运动参数估计有两种估计方法^[24,25]：第一种是先得到确定像素、特征点或块的运动矢量，然后用一种回归方法寻找与估计最优的全局运动参数；第二种是在一

个给定的参数集下，通过最小化预测误差来直接估计全局运动参数^[26]。本文是采用第一种方法—基于特征点的全局运动参量估计。

2.2.1 提取特征点

图像特征点的方法不仅易于用计算机处理，而且容易实现人机交互。有关特征点提取方法主要分为两类：基于图像边缘的方法和基于图像灰度的方法。基于图像边缘提取方法依赖于边缘检测和轮廓提取。这种方法易受到超声的干扰；而基于图像灰度的方法主要通过计算点的曲率及梯度来检测角点，考虑的是像素点邻域的灰度变化，不是整个目标的边缘轮廓，因此其受噪声干扰相对较小。目前相关的算法已有很多种被提出。如：由英国牛津大学的 Smith 和 Brady 于 1997 年提出的一种 SUSAN 算法^[27]；还有 C.Harris 和 M.Stephen 于 1988 年提出的角点检测方法—Harris 算子^[28]等。因为 Harris 算子的角点检测公式只涉及图像的一阶导数，所以它的计算简单有效。同时在图像旋转、灰度、噪声影响和视点变换的条件下，Harris 算子与其他算子相比是最稳定的一种特征点提取算子。因此本文采用的 Harris 算子。

Harris 算子图像的一阶导数：

$$M = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \otimes \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (2.6)$$

其中， I_x 代表图像中该点在水平方向的导数； I_y 代表图像中该点在竖直方向的导数； σ 为高斯平滑滤波函数标准方差；符号“ \otimes ”的含义是卷积。

对式(2.6)经相似对角化处理后得：

$$\begin{aligned} M &= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \otimes \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \\ &\Rightarrow R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R \end{aligned} \quad (2.7)$$

其中， λ_1, λ_2 是四元矩阵 M 的特征值。这样一个四元矩阵对应着一个像素。这里我们把 R 看成旋转因子，亮度变化量仅由 λ_1, λ_2 分析得到^[29]：当 λ_1, λ_2 二者都较大时，定义为角点；当二者只有一个较大，而另一个较小时，定义为边缘；当二者都较小时，定义为平坦区域，用 R 来表示。如图 2.3 是理想的角点、边缘和平坦地区示例图。

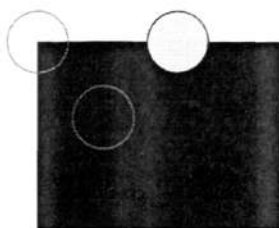


图 2.3 理想的平坦、边缘和角点区域示意图

Harris 指出 λ_1, λ_2 是 M 的自相关函数的主曲率, 因此很容易得到自相关函数矩阵的行列式 \det 和矩阵的迹 Tr :

$$\begin{aligned}\det(M) &= \lambda_1 \lambda_2 \\ Tr(M) &= \lambda_1 + \lambda_2\end{aligned}\quad (2.8)$$

则我们定义兴趣点检测 Harris 算子为:

$$R = \det(M) - k Tr^2(M) \quad (2.9)$$

其中, k 为默认常数 (一般取为 0.02 到 0.04)。当 R 的值大于某一阈值, 并且某领域取得局部极值时, 我们把该点标定为候选角点。根据上面的理论, 得出角点提取的步骤如下:

(1) 采用水平和竖直差分算子对每个像素进行滤波得到 I_x 、 I_y 。

(2) 利用式(2.6)进而求得 M 。

(3) 利用式(2.8)、(2.9)得出 R 。

(4) 判断选择, 若同时满足 R 大于某一阈值和 R 是某邻域的局部极大值的这两个条件的点就被标定为角点。否则淘汰。

2.2.2 匹配特征点

在拍摄事物时, 摄像机会发生平移、旋转和缩放等各种不同的运动。而基于传统的矩形模板匹配只适合处理视频图像间发生平移运动或只有较小幅度的旋转和缩放运动的情况。这大大限制了它使用范围。考虑到圆具有中心对称和旋转不变的特点, 本文采用基于圆形模板匹配搜索法完成特征点的匹配。

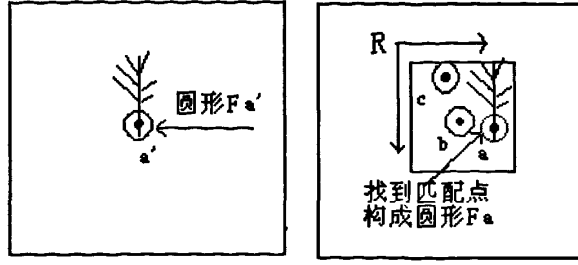


图 2.4 圆形块匹配示意图

本文采用以参考帧特征点为圆心，构造半径为 r 的圆形模板图像作为特征窗，在当前帧中进行搜索匹配的圆形模板。匹配搜索过程如图 2.4 描述所示： $F_{a'}$ 为参考帧中以特征点 a' 为圆心构造的一个圆形模板。 F_b 是在当前帧与 $F_{a'}$ 位置对应的一个圆形模板，以此圆形模板为中心，设置一定大小矩形搜索区域 R 。在 R 内以每个像素点为圆心构造 N 个与 $F_{a'}$ 同样大小的待匹配圆形模板，从中找出与 $F_{a'}$ 最相似的模块 F_a ，那么圆形模块 F_a 的圆心就是要寻找的匹配点。在搜索匹配的圆形模板时，本文采用的是全搜索算法(Full Search Method, FS)，即穷尽搜索法匹配搜索法。这种算法是所有经典搜索策略中最简单、可靠的算法，它可以找到全局最优模块。其基本思想就是计算出整个搜索区域内所有候选模板与待匹配模板的相似性度量 SAD 值，选择最小的 SAD 所对应的模块就为所求的待匹配模板。定义相似性度量 SAD 的准则为^[30]：

$$SAD = \min \left\{ \sum_{\substack{(x,y) \in F_c \\ (x',y') \in F'}} |I(x,y) - I'(x',y')| \right\} \quad (2.10)$$

传统的穷尽搜索法的相似性判断都是将候选模块与待匹配模块内所有像素点灰度差的绝对值都累加完后才比较 SAD 的大小，其计算量很大。为降低时间消耗，本文采用了一种基于设定阈值的动态累加的改进型相似性判断基准($SSAD$)。即，通过预先设定一个阈值，然后边累加边判断，提前结束那些不满足条件的圆形模板的累加计算。具体实现步骤如下：

Step 1、计算以搜索域 R 中心点构成的圆形模板 F_b 与圆形模板 $F_{a'}$ 的 SAD 值，并设此值为初始阈值 T 。

Step 2、计算 R 中下一个待匹配点(从 R 中心位置相邻处展开)构造的圆形模板与 $F_{a'}$ 的 $SAD(x,y)$ 。在累加 $SAD(x,y)$ 过程中，每累加一个灰度差值就与初始阈值 T 做一次

判断。当累加到某一点 (x_i, y_i) 时, 此时的 $SAD(x_i, y_i)$ 已经超过 T , 停止累加转向执行第三步, 否则继续累加, 继续判断。若直到累加完最后一个灰度差值仍没有超过 T , 初始阈值不变, 返回第二步继续执行。

Step 3、更新初始阈值 T , 即 $T = SAD(x_i, y_i)$ 。返回执行第二步。直到计算完 R 中所有像素点构造的圆形模板。其中, 最小的 SAD 所对应的圆形模板的圆心就是要寻找的匹配特征点, 完成匹配。

在采用块匹配搜索法时搜索区域和模板的大小直接影响着计算量和计算精度。如果搜索区域和模板大则会导致计算量的加大和像素点错位; 如果过小, 则包含的信息量又过少, 找不到最优的匹配模板^[31]。本算法圆形模板的半径 r 占8个像素点, 搜索区域为 25×25 的特征窗。图 2.5(b)给出了利用本方法对图 2.5(a)上的特征点集在当前帧中搜索到的匹配点集的结果。



图 2.5 特征点集和匹配特征点集

2.3 全局运动参量的鲁棒估计

根据式(2.2)中的运动模型, 可以得到参考帧中像素点与当前帧中对应的匹配像素点满足线性参数系统, 见式(2.11)所示:

$$\begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & -y_n & 1 & 0 \\ y_n & x_n & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} x_1' \\ y_1' \\ \vdots \\ x_n' \\ y_n' \end{pmatrix} \quad (2.11)$$

其中, (x_i', y_i') ($i=1, 2, 3, \dots, n$) 为参考帧上 N 个特征点坐标; (x_i, y_i) ($i=1, 2, 3, \dots, n$) 为当前帧上 N 个对应的匹配特征点坐标; (a, b, c, d) 为要求解的相邻帧之间的全局运动参量。该线性参数系统一般情况下是无解的。但在工程上, 可以利用最小二乘法得到运动

参量的近似解。

对于在获得的匹配特征点对中，我们把特征点选在了运动目标上、被遮挡、误匹配和移出视场的特征点定义为局部特征点，也叫外点；而把那些位于背景区域并且正确匹配的点才称为全局特征点。如图 2.6 所示，标注“○”的这类特征点是位于纹理相同的区域，这些特征点会导致匹配错误；标注“□”的这类特征点位于运动物体上。这两类特征点都是外点。

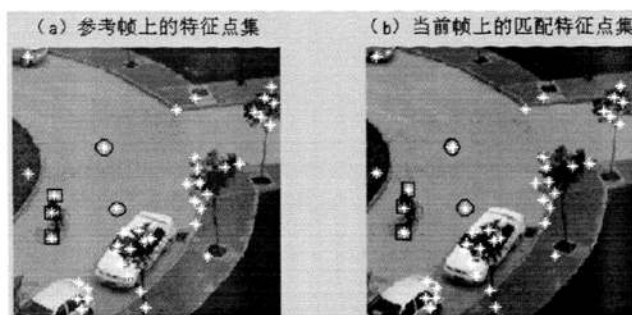


图 2.6 标有外点集的特征点集和匹配特征点集

最小二乘法对这些外点非常敏感，会对全局运动参量精度带来影响。因此在估计全局运动参数之前，必须先对去除这些不可靠的外点。基于以上考虑，本文采用一种统计鲁棒分析法^[32]去除这些外点，其算法如下：

Step1、随机选取 3 对匹配特征点对，利用式(2.11)得到运动参数 (a_1, b_1, c_1, d_1) 。

Step2、将 (a_1, b_1, c_1, d_1) 代入式(2.2)得到 E_1 。

Step3、重复执行第一步与第二步 M 次，得到 $E_1, E_2 \dots E_M$ ；

Step4、取 $E_0 = \min(E_1, E_2 \dots E_M)$ 。 E_0 所对应的全局运动参量就是无外点参与估计的参量。

第三步中的迭代次数 M 与无外点参与估计的可能性 p 满足一定关系，见式(2.12)：

$$p(\varepsilon, q, M) = 1 - (1 - ((1 - \varepsilon)q)^3)^M \quad (2.12)$$

其中， q 是随机选取的特征点中无外点的可能性； ε 是外点占有所有特征点中的比例。一般， $\varepsilon < 0.5$ 。如：当 $q=0.7$ ， $\varepsilon=0.3$ ，经过 25 次就可以得到 $p=0.993$ 。如图 2.7 是经过统计鲁棒分析法处理后的结果。



图 2.7 经鲁棒分析法处理后的特征点集和匹配特征点集

在统计鲁棒分析中,随机选取特征点的个数直接影响着估计的精度:如果选取的特征点过少,不能完成反映全局运动信息,则得不到全局最优参量,也就是 E 达不到实际最小;而选取的特征点过多,在利用最小二乘法估计时,外点对其影响不十分明显,就不能彻底的删除外点。因此,本文又引入了 LMA 算法(Levenberg-Marquardt Algorithm)对全局运动参量做进一步的优化处理^[33]。

定义 $m = (m_1, m_2, m_3, m_4)^T = (a, b, c, d)^T$ 。LMA 迭代公式为:

$$m^{(n+1)} = m^{(n)} + s^{(n)} \quad (2.13)$$

其中, $s^{(n)}$ 是每次迭代时一个更新量。其值可以通过式(2.14)求得:

$$[J^T(m^{(n)})J(m^{(n)}) + \mu^{(n)}\beta]S^{(n)} = -J^T(m^{(n)})r(m^{(n)}) \quad (2.14)$$

其中, β 表示单位矩阵, μ 表示稳定因子。 μ 是随时间变化的,初始值为 1,它以 10 幂次递减或递增来控制对 m 的估计。当 m 更新后,若 E 减少,则增加 μ 以跟随梯度方向;反之减少 μ 。 J 为 Hessian 矩阵, r 为梯度向量。 r 和 J 可分别通过式(2.15)和式(2.16)得到:

$$r = [e(1,1), e(1,2), \dots, e(2,1), e(2,2), \dots, \text{all } x, y]^T \quad (2.15)$$

$$J(i, j) = \sum_{x, y} \frac{\partial e(x, y)}{\partial m_i} \frac{\partial e(x, y)}{\partial m_j} \quad (2.16)$$

最小二乘法估计出的全局运动参量为 LMA 最初迭代值。因为初值已经很接近全局最优解,所以整个迭代过程可以很快的收敛。LMA 迭代过程实现步骤如下:

Step1、初值 $m^{(n)}$ 代入式(2.5)得到 $e(x, y)$ 。

Step2、分别通过式(2.16)、式(2.15)、式(2.14)算出 J , r , $s^{(n)}$ 。

Step3、利用式(2.13)更新全局运动参量，得到 $m^{(n+1)}$ 。

Step4、将 $m^{(n+1)}$ 代入式(2.5)算出 E ，判断 E 是否小于指定阈值(一般根据实际情况而定)：若小于指定阈值，停止迭代，即得到了全局最优解；否则返回第一步，继续执行直到 E 满足条件。

2.4 背景匹配

根据得到的全局运动参量，以参考帧为基准，对当前帧背景进行匹配。在匹配过程中，由于很多时候计算出来的坐标值不是整数，没办法得到灰度值为多少，因此需要对其进行调整。本文采用双线性内插法实现对当前帧背景的匹配变换。把全局运动参量和当前帧像素点的坐标 (x, y) 代入式(2.2)得：

$$Z(x, y) = ((x_0 + \alpha_x), (y_0 + \alpha_y)) \quad (2.17)$$

其中， $Z(\cdot)$ 表示仿射变换， x_0 ， α_x 分别表示计算得到的横坐标值的整数部分与小数部分； y_0 ， α_y 分别表示计算得到的纵坐标值的整数部分与小数部分。利用双线性内插变换后的背景为：

$$\begin{aligned} I(x, y) = & (1 - \alpha_x)(1 - \alpha_y)I(x_0, y_0) + \alpha_x(1 - \alpha_y)I(x_0 + 1, y_0) + \\ & (1 - \alpha_x)\alpha_y I(x_0, y_0 + 1) + \alpha_x\alpha_y I(x_0 + 1, y_0 + 1) \end{aligned} \quad (2.18)$$

2.5 实验数据分析

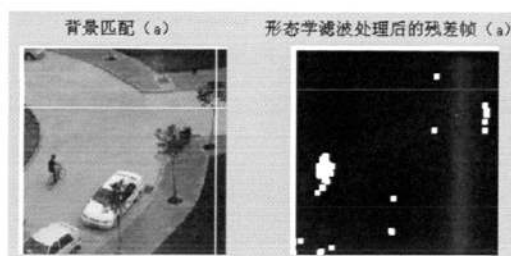
为了验证本章所提及算法的可行性，本文进行了两组实验。实验数据：256×256 大小的图像序列；实验目标：骑自行车的人；摄像机在拍摄人的过程中发生水平和旋转运动，导致使背景中的汽车、树等发生了全局运动。算法在 PC 机上利用 Matlab7.1 进行验证。搜索域为 25×25，圆形模板块半径取值为 8 个像素点。

第一组实验：摄像头发生水平位移。取图像序列中第 80 帧（参考帧）、第 82 帧（当前帧），如图 2.8 所示。从相邻两帧上那两条直线的交点可以体现全局运动幅度，交点坐标值为(244, 77)。

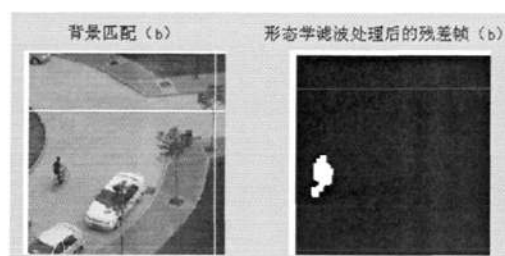
第二组实验：摄像头发生水平和旋转运动。取图像序列中第 70 帧（参考帧）、第 74 帧（当前帧），如图 2.9 所示。从图片上那条直线中可以体现相邻两帧旋转幅度。



(a) 参考帧与当前帧



(b) 未经过鲁棒分析的处理结果

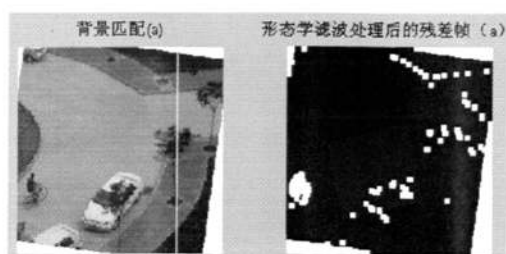


(c) 经过鲁棒分析的处理结果

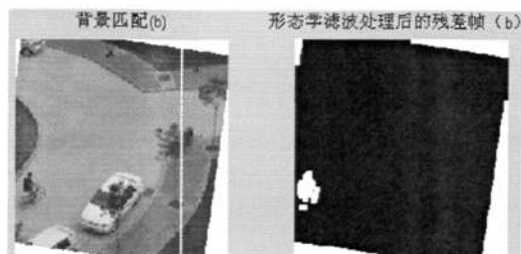
图 2.8 摄像机发生水平运动



(a) 参考帧与当前帧



(b) 未经过鲁棒分析的处理结果



(c) 经过鲁棒分析的处理结果

图 2.9 摄像机发生水平运动+旋转运动

图 2.8(b)、图 2.9(b)分别表示对图 2.8(a)、图 2.9(a)在全局运动估计过程中未经过鲁棒分析和 LMA 优化处理的残差帧。图 2.8(c)、图 2.9(c)表示经过鲁棒分析和 LMA 优化处理的残差帧。从这两组残差帧的对比中看出图 2.8(b)、图 2.9(b)中残差部分除了是由

于运动目标的运动造成的，还有背景的残差，所以没有达到理想效果。而图 2.8(c)、图 2.9(c)中的残差全部是由运动物体的运动造成的。因此可以说明在全局运动估计时经过鲁棒分析处理和 LMA 对参量优化提高了的精确。在完成特征点匹配过程中，本文采用了基于改进型的圆形模板匹配搜索法(SSAD)。这样在很大程度上降低了采用传统穷尽搜索法的计算量。使得本算法不仅可以处理较大幅度的旋转全局运动，而且在时效性上也得到了提高（多次计算平均时间为 140ms 左右）。为了验证此性质，本文将文献[5]中的基于光流法（简称为 GME1）与本算法（简称为 GME2）从处理的运动范围和估计的全局运动参量精度（可达到的最小误差值）两方面做了比较，见表 2.1。从表可以看出本算法对平移、旋转、缩放这三种全局运动处理范围分别可达大于 35 像素点，大于 25° 和 0.85~1.14。三种全局运动参数估计的精度又分别达到 0.1 像素， 0.1° 及 0.001。由此得出，与基于光流法相比，本文算法在处理全局运动幅度和全局运动参量估计精确这两方面都得到改善。

表 2.1 两种算法性能比较表

算法	γ	$\theta / (^\circ)$	$c, d / (\text{像素})$
GME1 可处理的运动范围	0.92~1.08	>7	>10
GME2 可处理的运动范围	0.85~1.14	>25	>35
GME1 全局运动参数估计精度	0.01	0.6	0.3
GME2 全局运动参数估计精度	0.001	0.1	0.1

2.6 本章小结

考虑到目前全局运动背景补偿中存在的问题，本文以此为出发点对一些算法做了改进与组合。首先介绍完几种摄像机运动的参数模型后，使用 Harris 算法完成特征点提取；然后提出一种改进型的圆形模板匹配法完成特征点的匹配。这种改进的方法不仅提高了时效性，而且可以处理相邻帧间发生的较大幅度平移、缩放、旋转等全局运动。考虑外点的存在对全局运动估计的影响，本文又将统计鲁棒分析法和 LMA 优化算法结合的应用到仿射变换模型的参数估计中，成功的去除了外点，进而提高全局运动参量的精度。最后利用双线性内插法实现对当前帧背景匹配，即实现了运动背景静态转化。从实验结果与实验数据的分析得出本章所提及算法的可行性和与其他传统算法相比的优越性。

第三章 运动目标的检测

在对视场内的运动目标进行实时的观测，首先要从图像序列中，将感兴趣的运动区域从背景图像中提取出来。这就是运动目标检测要完成的主要任务。如何能快速、准确的将目标检测直接影响目标的分类、跟踪和行为理解等后期处理。因此运动目标检测是智能监控的关键技术之一。判断一个检测算法好坏，一般依靠以下三个基准：

(1) 准确性高。这是整个检测算法的关键，能够准确地将目标信息从图像序列中分离出来。

(2) 鲁棒性强。在对视场进行实时监控时，图像序列很容易受光照、气流的影响以及视频信号受采集噪声等因素的影响，这使得目标的检测变得更加困难，因此这就要求检测算法的鲁棒性强。

(3) 实时性好。要实现运动目标的正常检测，至少要比运动目标移出视场范围的速率快。但一般运动目标的检测只做视频处理的基础部分，所以目标检测之后还要预留较多的时间给其他环节的图像处理，所以实时性也是非常重要的。

上一章已经介绍了如何将把运动背景静态化。在这章将对几种典型的静态背景运动目标了进行探索和研究，提出了一种更为有效的、适合本系统的运动目标检测算法。

3.1 运动目标检测的一般过程

相对于静态背景下的运动目标检测，运动背景下的运动目标检测需要做全局运动背景补偿处理的前续工作，使其静态化。因此我们将运动背景下运动目标检测的一般过程可以表示如图 3.1 所示：

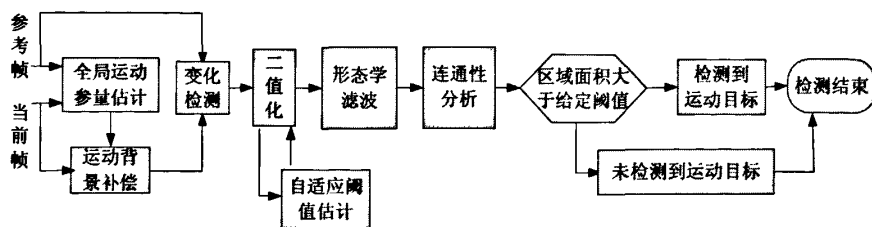


图 3.1 基于单目视觉的运动背景的目标检测框图

通过全局运动补偿后，动态背景下运动目标的检测已转换为静态背景下运动目标检测。从框图可以看出静态背景下运动目标的检测一般主要包括两大部分：变化检测与后续处理工作，即二值化、形态学滤波、连通性分析等。

变化检测的作用主要是利用背景与目标运动状态的显著差异来得到目标区别于背景的信息。它是整个过程的重点和难点部分。相关的算法已有很多，这部分内容将在 3.2 节具体介绍。二值化，形态学滤波、连通性分析这几部分是对经变化检测后的差分帧做后续处理工作，以达到提取出准确和完成的运动目标，这部分在 3.3 节详细阐述。

3.2 静态背景下的变化检测

对于一般应用来说，图像序列的变化检测是检测相邻两帧图像的差异。视频图像中任何细小的运动都会体现在图像序列的变化上，若能检测这种变化，就可以分析其运动特性。最广泛的独立变化检测算法有：光流法^[5]、背景差分法（背景与当前帧）^[6]和连续帧差分法（包括相邻两帧、三帧、多帧差分或隔帧差分）^[4,34]。由于图像中有多种噪声（环境噪声：如光照变化；图像获取设备引入的扰动；图像预处理中引入的噪声）的影响使的这几类算法在处理实时性、有效性、抗干扰性各具有差异。

3.2.1 光流法

光流(Optical Flow)是 Gibson 在 1950 年提出的，它是表示物体带光学特征部位的移动投影到图像平面上所形成的。当物体运动时，图像上对应物体的亮度模式也在运动，因此光流可看成是图像亮度模式的一种运动表现。光流场便是由于时变图像亮度或灰度特性变化所得到的运动场。一般情况下，可以认为光流与运动场没有太大区别，因此就可以根据图像运动来估计相对运动。

光流场的最初计算方法是由美国学者 Horn 和 Schunck 提出的^[35]。Horn-Schunck 是建立在整个图像上光滑变化的假设来求解光流，即运动场同时满足光流约束方程和全局平滑性。这种方法在只考虑离散情况下，根据光流约束方程，在一点 (i, j) 及其 4 个邻域上，光流误差的离散量表示为：

$$e^2(i, j) = I_x u(i, j) + I_y v(i, j) + I_t \quad (3.1)$$

其中， $u(i, j)$ ， $v(i, j)$ 分别为这点光流空间 x 和 y 方向的分量； I_x ， I_y ， I_t 分别为图像 $I(x, y, t)$ 在空间 x 、 y 方向上和时间 t 方向上的梯度。

由该点与其 4 个邻域上的光流值差分计算的光流平滑量为：

$$\begin{aligned}
 s^2(i, j) = \frac{1}{4} & \left[(u(i, j) - u(i-1, j))^2 + (u(i+1, j) - u(i, j))^2 \right. \\
 & + (u(i, j+1) - u(i, j))^2 + (u(i, j) - u(i, j-1))^2 \\
 & + (v(i, j) - v(i-1, j))^2 + (v(i+1, j) - v(i, j))^2 \\
 & \left. + (v(i, j+1) - v(i, j))^2 + (v(i, j) - v(i, j-1))^2 \right]
 \end{aligned} \quad (3.2)$$

则极小化函数为：

$$E = \sum \sum (e^2(i, j) + \alpha s^2(i, j)) \quad (3.3)$$

光流的迭代方程为：

$$u^{n+1} = \bar{u}^n - I_x \frac{I_x \bar{u}^n + I_y \bar{v}^n + I_t}{\alpha + I_x^2 + I_y^2} \quad (3.4)$$

$$v^{n+1} = \bar{v}^n - I_y \frac{I_x \bar{u}^n + I_y \bar{v}^n + I_t}{\alpha + I_x^2 + I_y^2} \quad (3.5)$$

其中， n 为迭代次数（一般初始值为 0），当迭代的结果小于预定值时，迭代终止； \bar{u} ， \bar{v} 分别为 u ， v 在点 (i, j) 的领域处的平均值。

光流法的缺点是由于存在孔径和遮挡问题。用光流法估算二维运动场的结果是不确定的、抗噪性能差，且计算方法相当复杂，无法在视频处理中满足实时性。其优点是在摄像机运动存在下，也可以检测出独立的运动目标。

3.2.2 基于背景差分的方法

基于背景差分方法是利用当前图像与背景图像的差分，通过阈值来检测出运动区域的一种技术，如图 3.2 所示。

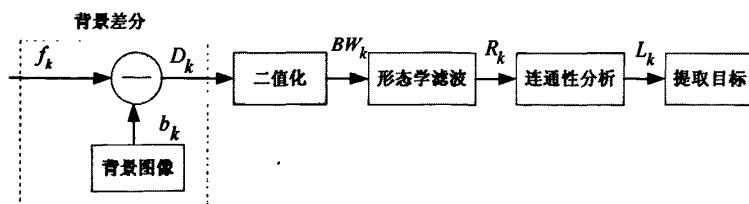


图 3.2 基于背景相减的变化检测流程

这种方法是最理想的一种方法，因为它最完全的特征数据，可以更好的解决帧间差分法中目标区域偏大、目标内部出现孔洞的问题，很多学者都致力于研发。但是这种方法存在以下两个问题：

(1)背景的提取。现在，有关背景的提取方法大致有渐消记忆递归最小二乘背景估

计法和 W4 方法两大类^[36]。渐消记忆递归最小二乘背景估计法是在有目标存时的一种背景提取方法，可是当目标所占面积较大时，就得不到理想的背景帧。而 W4 方法是一种无运动目标存在时的背景提取方法。这种方法不仅应用面狭窄，而且必须需要有人参与。

(2)背景的更新。在背景更新的困难之处主要是运动目标对应像素点的更新问题很难解决。如果更新率太小，会使静止目标被认为运动目标；如果更新率太大，又会把运动目标更新到背景中，进而产生空洞。目前大部分算法对更新率采取“折中”处理的方式。

3.2.3 连续帧差分法

检测相邻二帧图像序列之间的变化的最简单方法是帧差法，即直接利用相邻的两帧图像的灰度差来检查运动目标是否存在及其分析它的运动特性^[37]。在运动背景匹配后，可以与参考帧进行直接的像素差值运算，如下：

$$\Delta I = |I_t - I_{t-1}| \quad (3.6)$$

然后选取适当的阈值 T ，可以得到二值图像为：

$$M(x, y) = \begin{cases} 1, & \Delta I(x, y) \geq T \\ 0, & \Delta I(x, y) < T \end{cases} \quad (3.7)$$

如果差分图某一点的灰度变化大于一定的阈值 T ，则认为该点为目标上的点，否则为背景点。

基于两帧图像来提取运动目标的确简单，运算速度也快。但是它也具有几方面较大的缺陷：(1)对背景的匹配精度要求高。(2)如果两帧图像的改变是由光照等其它因素引起的非独立运动目标，基于两帧图像差分法就检测不到正确的结果。(3)由于独立目标不同的运动特点可能产生误测。可能导致连续两帧几乎完全一样，也有可能产生“多目标”等问题。例如，如图 3.3 中的利用两帧差分法得到的前后两幅图像差。从图中看出图像差出现两个不同的运动区域：一个对应前一帧的运动目标，而另一个则对应当前帧中的运动目标，因此难确定运动目标的位置。

针对以上存在的问题，可以采用基于三帧图像的检测方法。

$$M(x, y) = M(x, y, t, t-1) \cdot M(x, y, t, t-2) \quad (3.8)$$

利用该方法不仅可以很好的解决“多目标”，而且可以减小外界噪声对其的影响。如图 3.3 就是经式(3.3)处理后得到正确的结果。

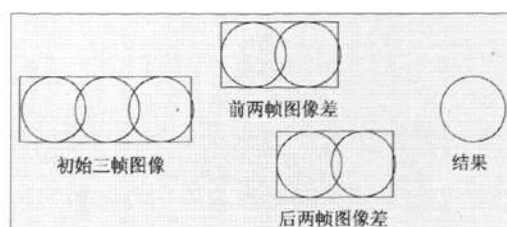


图 3.3 基于三帧图像的运动变化检测示意图

然而，基于三帧图像的检测方法仍然存在一个现象，即“目标的分裂现象”，这种现象同样会导致检测失效，如图 3.4 所示。

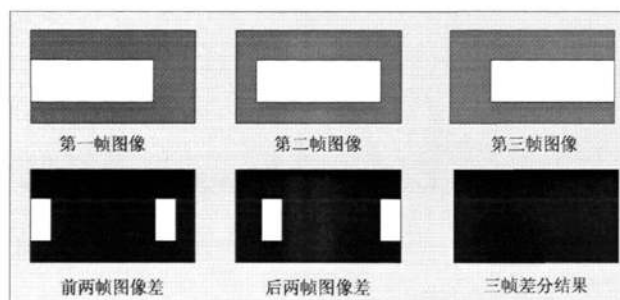


图 3.4 运动目标分裂时三帧变化检测失效图

这种现象是由运动目标的运动过慢或尺度过长引起。针对这种现象，很多学者也提出解决办法。其中一种较理想的解决办法是如果目标比背景亮，在图像差分后可以将差值为负的像素归零，即只取差值为正的像素值，这样的话差分剩余运动图像中就只会出现当前帧中的目标区域；如果运动目标相对背景较暗，则差分剩余运动图像中只会出现前一帧中的目标区域。这样可以消除“多目标”或“目标分裂”现象引起的含混问题

经过对上面三种经典的变化检测算法的对比，本文就是采用第三种方法。

3.3 差分图像的后续处理

3.3.1 二值化

二值化是在图像分割算法中将图像中运动目标区域从差分图像中分离出来的方法。它是目标检测中不可缺少的一个步骤。二值化方法归结起来有两种：固定阈值二值化和自适应阈值二值化。

固定阈值二值化是设定固定的阈值 T ，当差分值大于阈值 T 时，将其判为运动目标像素，反之就为背景像素，见式(3.9)。

$$f_i(x, y) = \begin{cases} \text{background} & f(x, y) < T \\ \text{motion} & f(x, y) \geq T \end{cases} \quad (3.9)$$

其中, T 的取值是由经验值确定 (环境不同, T 的大小也不同), 所以这就缩小了固定阈值二值化使用范围。后来, 人们成功的将分割算法引入差分图像的二值化处理问题中, 分割算法的引入可以自动选取合适的阈值将图像中运动目标区域从差分图像中分离出来。例如, Kim J.B 和 Park H.S 等就采用自适应阈值分割法^[38]。其基本思想是: 设定差分图像的直方图是由三个加权单峰高斯模型混合而成, 则差分图像的混合概率密度函数为下式:

$$P(d) = \frac{\varpi_1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(d-\mu_1)^2}{2\sigma_1^2}\right] + \frac{\varpi_2}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(d-\mu_2)^2}{2\sigma_2^2}\right] + \frac{\varpi_3}{\sqrt{2\pi}\sigma_3} \exp\left[-\frac{(d-\mu_3)^2}{2\sigma_3^2}\right] \quad (3.10)$$

其中, σ_i 为均值的标准差, μ_i 为单高斯模型的均值, ϖ_i 为差值数量。则 T 的取值问题就转为寻找划分三个高斯模型问题, 即阈值出现在两个模型两峰之间的谷底。如图 3.5 所示。

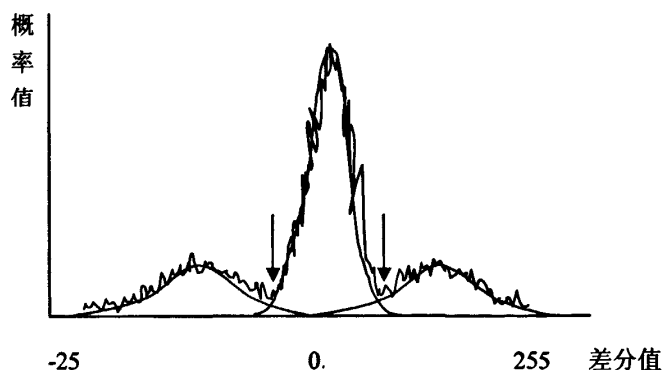


图 3.5 概率分布图

我们定义差分值的概率为:

$$m(d) = \frac{1}{n} \text{num}(d) \quad (3.11)$$

其中, d 为差分值, n 为整个差分图像的像素的个数。

则 $P(d)$ 与 $m(d)$ 的最小均方误差为:

$$e = \frac{1}{n} \sum_{d=-255}^{255} [P(d) - m(d)]^2 \quad (3.12)$$

则使均方误差最小的值就是分割三高斯模型的最佳阈值。

这是一种有效的阈值分割方法, 但是这种方法需要建立高斯模型, 计算量非常大, 不利于实时检测。本文采用基于最大熵的方法确定阈值^[39,40]。用信息熵来表示运动目标

和背景的灰度分布。阈值选取的不同，取得的背景区域和目标区域总信息量就不同。最佳的阈值就是使背景区域和目标区域总信息量最大化的值。

假设图像灰度的后验概率为：

$$p_i = \frac{n_i}{N^2}, \quad i = 1, \dots, M \quad (3.13)$$

$$N^2 = \sum_{i=1}^M n_i \quad (3.14)$$

其中， n_i 为灰度值为 i 的像素的总数； N^2 为图像中总像素数； p_i 为相应的概率。如果我们设阈值为 s ，则把整个灰度值分为两种，即背景灰度值与目标灰度值。这两种灰度值的分布式分别如下：

$$\begin{aligned} & \frac{p_{i+1}}{1-Z_s}, \frac{p_{i+2}}{1-Z_s}, \dots, \frac{p_M}{Z_s} \\ & \frac{p_1}{Z_s}, \frac{p_2}{Z_s}, \dots, \frac{p_i}{Z_s} \end{aligned} \quad (3.15)$$

$$\text{其中 } Z_s = \sum_{i=1}^s p_i$$

而背景与目标的熵分别如下：

$$H(A) = \sum_{j=s+1}^M \frac{p_j}{1-Z_s} \ln \frac{p_j}{1-Z_s} \quad (3.16)$$

$$H(B) = \sum_{j=1}^s \frac{p_j}{Z_s} \ln \frac{p_j}{Z_s} \quad (3.17)$$

直方图的熵为：

$$\Phi(s) = H(A) + H(B) \quad (3.18)$$

当使 $\Phi(s)$ 最大化时的 s 为最佳阈值，即能把差分图象分割为目标区域和背景区域两部分的阈值。

3.3.2 形态学滤波

二值化后的差分图像，在目标区域会有许多小孔洞和孤立的小点，还有一些断裂的轮廓线条。像这类问题可以用形态学滤波法来解决。其基本思想是：先定一个形态结构元素，通过腐蚀模板过滤消除孤立噪声点；然后再利用同样大小的膨胀模板过滤，对断裂或者孔洞部分进行处理，进而实现对图像分析和识别的目的。即就是对图像进行两种

基本操作：膨胀操作和腐蚀操作^[41]。膨胀的主要作用是使目标的像素数增多，来填补孔洞与桥接裂缝；腐蚀(Erosion)是膨胀的逆运算，它的作用是消除目标图像中孤立的小点和不必要的细小连线。其具体操作如下：

常用的结构元素有圆形、正方形和菱形，本文采用了正方形的，其定义如下：

$$B = \begin{matrix} & \bullet & \bullet & \bullet \\ \bullet & & * & \bullet \\ & \bullet & \bullet & \bullet \end{matrix} \quad (3.19)$$

结构元素的反转为：

$$B^s = \{-b | b \in B\} \quad (3.20)$$

膨胀：

$$\delta_B(X) = X \oplus B^s = \bigcup_{b \in B} X_{-b} = \{z \in E | B_z \cap X \neq \emptyset\} \quad (3.21)$$

腐蚀：

$$\varepsilon_B(X) = X \ominus B^s = \bigcap_{b \in B} X_{-b} = \{z \in E | B_z \subset X\} \quad (3.22)$$

其中 X 为二值图像， B_z 为 B 中心（“*” 的位置）平移到位置 z 得到的图像， E 为整幅图像的栅格。膨胀和腐蚀这两个过程的描述，如图 3.6 所示。

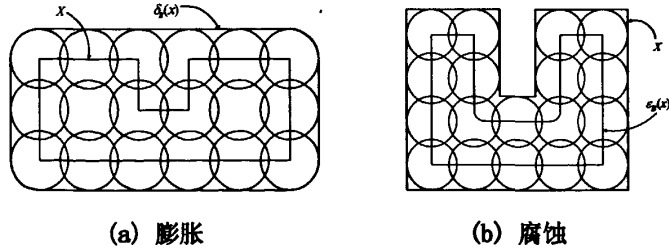


图 3.6 膨胀和腐蚀算子

从图 3.6 看出，膨胀的结果使 X 原有的凹槽被填平并且扩张成为一个具有圆角的长方形。腐蚀的结果使原来 X 中的凹槽变得更深更宽，并且底部变成圆滑的转角。由膨胀与腐蚀还可以衍生出两种基本形态学滤波运算：开运算(Opening)和闭运算(Closing)。

开运算操作是对图 3.6 (b)中的 $\varepsilon_B(X)$ 再作一次膨胀，如下式：

$$O_B(X) := \delta_B[\varepsilon_B(X)] \quad (3.23)$$

闭运算操作是对图 3.6(a)中的 $\delta_B(X)$ 再作一次腐蚀，如下式：

$$C_B(X) \doteq \varepsilon_B[\delta_B(X)] \quad (3.24)$$

对同一图形进行开运算与闭运算操作的结果区别很大：开运算的结果可以消除孤立点和细小目标，在边界平直的部分将与原来的 X 相一致，面积改变不明显，但在转角处都变成圆滑的，如图 3.7(a)所示。闭运算的结果是填充目标内部细小孔洞并连接邻近目标，也是在不明显改变面积的情况下平滑其边界，得到的是一个四角圆滑的长方形，如图 3.7(b)所示。

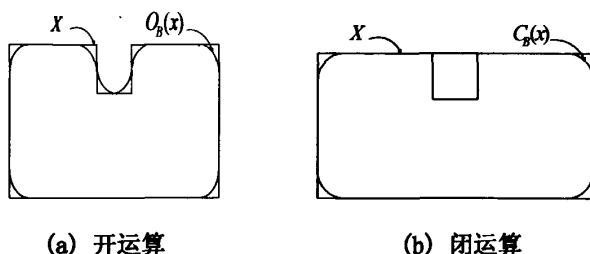


图 3.7 开运算和闭运算的操作

如何合理的选择膨胀、腐蚀、开操作和闭操作这四个运算，应根据实际情况而选择。本文是先采用开操作，后利用闭操作。

3.3.3 连通性分析

再经过数学形态学滤波后，图中的大部分小孔洞和孤立的小点被去除了，还有小的间隙也被连接上了，但是依旧存在一些大的孔洞，在对图像进行处理时会把背景误判为目标的，因此如果是多目标检测和标定则必须确定连通成分，所以需要进行连通性分析。

像素间的连通性是建立目标边界或确定区域元素过程中的一个重要概念。如果一个像素集合内的每一个像素与集合内其它像素连通，则称该集合为一个连通成分。其基本思想就是在形态学滤波处理后的图像中找出所有连通成分，并对同一连通成分中所有点分配同一标记。通常有两种用来进行连通性分析的判据，即 4 连通和 8 连通，如图 3.8 所示。本文是采用 8 连通来进行连通性分析判据的。

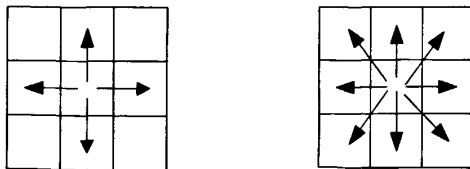


图 3.8 左图为 4 连通，右图为 8 连通

连通性分析的含义是实时地识别出每一目标的位置。序贯算法和递归算法是两种典型的标记方法。序贯算法是查看某一点的邻点，给像素值为 1 的邻点分配一个已经使用

过的标记。若邻点有两个不相等的标记，就利用等价表记录等价标记。通常，这种算法需要对图像进行二次处理，在二次处理过程中，使用其给某一连通域中所有的像素分配唯一的标记。递归算法相对序贯算法计算效率是比较低的^[42]。因此人们一般采用序贯算法。本文采用基于 8 连通的序贯算法。其具体操作如下：

(1)从上到下、从左到右各个方向扫描图像。

(2)若像素点的灰度值为 1，则：如果右点、右上点、左点、左上点中只有一个有标记，就复制这个标记；如果这四点是一样的标记，就复制这个标记；如果这四点有不一样的标记，就复制其中任何一个标记，并且把这个标记以及其它不同的标记一起输入等价表中作为等价标记；否则就给这一个像素点分配一个新标记，且把这一新标记输入等价表中作为等价标记。

(3)重复执行第二步，直到判断完所有点。

(4)从等价表中每一等价集里找出最低标记。

(5)扫描整个图像，用最低标记替换每一标记。

经过基于 8 连通的序贯算法对图像分析后，我们就知道每个标签对应的所有像素点占整个图像大小的比例值。再将此值与预先规定的运动物体的大小值作比较判断：若大于或等于预先规定的值我们就认为检测到的运动物体，否则属于误判。

3.4 运动目标的定位

在检测到运动目标后，就需要准确的运动目标坐标的做出定位。模板匹配^[43]与形心定位是两种通用的方法。基于模板匹配的定位方法的优点是能够较准确的识别同一类别的物体。但是这种方法需要事先知道模板样本，且这个模块的选择能够将所要识别类物体与其他类别物体区分开。因为本文不是局限于识别某一类特定的物体，所以采用形心定位方法计算运动区域的几何形心来作为运动目标的中心坐标。计算过程见式 (3.25)。

$$(x_0, y_0) = \sum_{i=1}^N (x_i, y_i) / N \quad (3.25)$$

其中， (x_i, y_i) 为同一连通域像素点的坐标， N 为同一连通域的总像素个数， (x_0, y_0) 为几何中心的坐标。

3.5 本章小结

本章主要介绍运动目标检测，本章的一开始先归纳的介绍了几种常用变化检测算

法，涉及到三种方法：光流法、背景减除法、帧间差分法。经过对各算法的优缺点的对比，再结合本文的实际情况，本文采用三帧差分算法。三帧差分法不仅运算速度快、节省内存空间、而且可以减小外界噪声对其影响。除此之外，这种方法相对于两帧差分法可以很好的解决“多目标”、“目标的分裂现象”等问题。然后介绍了二值化，形态学滤波、连通性分析这几部分来完成对变化检测后的差分图像做后续处理工作，以达到提取出准确和完成的运动目标。最后利用形心定位的方法对运动目标坐标的做出准确定位。

第四章 运动目标的自动跟踪

4.1 概述

有关运动目标的跟踪算法已有很多,但是相关算法实现的系统大多是基于可视化方面的应用,即摄像机固定监控系统(监控视场小、目标易走失)或是人工辅助监控系统。本章将介绍一种自动跟踪的实现算法,使监控系统具有一定的智能化水平。

自动跟踪算法的思想是:在摄像机循环转动扫描时,利用上章介绍的算法检测是否有运动物体,当发现有运动物体,则使摄像机暂先停止转动,采用均值平移目标跟踪算对目标进行跟踪,通过对几帧的跟踪后确定出目标的中心点坐标位置及图像运动物体的运动轨迹,对运动物体的运动轨迹的分析,再使用相应的预测算法预测运动物体在图像下一帧可能出现的位置,然后控制云台摄像机以相应的速度转动到相应位置。此思想的系统流程框图如图 4.1 所示:

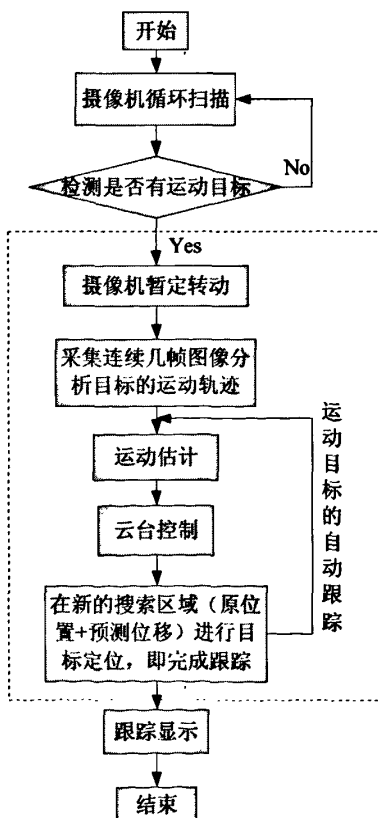


图 4.1 运动目标的智能跟踪系统流程

通过上述对自动跟踪算法分析过程,可知其关键技术是对目标的搜索定位、运动预

测和摄像机的控制，本章将在接下来的内容重点介绍这几部分。最后通过两组实验数据对本算法进行可行性分析。

4.2 运动目标特征选取

运动目标的跟踪技术常用的两种算法，即运动检测和目标识别来进行跟踪^[44]。运动检测来实现跟踪的算法是通过不断的目标检测方法来确定运动目标的位置完成跟踪。而目标识别来进行跟踪是通过研究物体目标的形状、尺寸等特征匹配来确定物体在下一帧图像中的位置。这种方法包含了目标识别和目标匹配两部分。本文采用后一种。这种方法基本思想主要包括两大步：首先提取目标的特征，然后对运动目标进行搜索匹配，实现目标在每帧图像中的定位。所以如何选择合适的目标特征（能使其在环境变化、目标自身变化或运动改变时都具有很好的鲁棒性）是非常关键的。

通常，选择的运动目标特征是从统计特征（直方图、各种矩特征）、视觉特征（图像边缘、轮廓、形状、纹理）、代数特征（图像矩阵的奇异值分解）等方面来选取^[34,48]。广泛选用的特征有颜色，直方图均值等。

4.2.1 颜色特征

每一种颜色在计算机中有不同的表达方式，主要有：RGB 格式（红、绿、蓝三种基色模型）、HVS 格式（色度、饱和度、亮度模型）、YUV 格式^[49]这几种方式。在色彩图像处理和计算机视觉的研究中，采用最多的是 RGB 格式和 HVS 格式。

光谱上的任何颜色都可以用红、绿、蓝三种单色混合而成。因此研究的色彩图像是研究光谱图像的一种情况。大部分图像采集设备都可以直接感知色彩的 R、G、B 三个分量，R，G，B 的比例关系决定了光的色度。RGB 模型如图 4.2 所示。

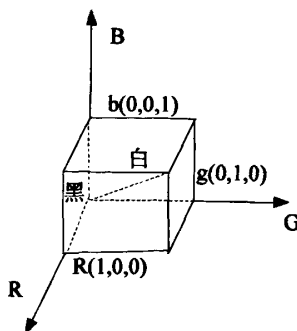


图 4.2 RGB 模型

其关系如下：

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \quad (4.1)$$

其中, r, g, b 称为色度坐标。由于 $r+g+b=1$, 所以只有两个色度坐标是相互独立的。

HVS 格式是由 Munseu 提出的色彩系统模型。这种格式也经常为艺术家所使用。HVS 格式是按照色度(hue)、饱和度(Saturation)和明亮值(value)建立起来的。

色度是色彩的重要属性, 是决定颜色本质的基本特性。它是由物体反射光线中占优势的波长来决定的, 不同的颜色取决于不同的波长。即七色(红、橙、黄、绿、青、蓝、紫)对应于七种波长。色彩信息在 HVS 颜色模型中绕 v 轴的旋转角给定的。如图 4.3 所示, 绿色对应于角度 120° , 蓝色对应于角度 240° ; 饱和度(S)是指一个颜色的鲜明程度, 颜色越深, 饱和度越高。例如: 深黄、深蓝等。饱和度为一比例值, 范围从 0 到 1 (所选颜色的纯度和该颜色最大的纯度之间的比率); 明亮度由物体的反射系数决定, 明度越大, 反射系数越大。表示色彩的明亮度范围也是从 0 到 1。

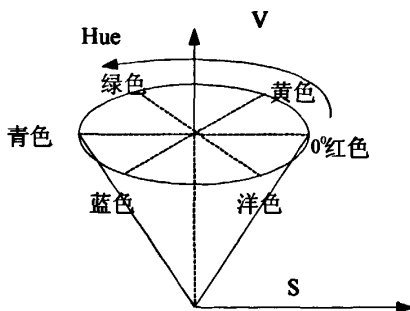


图 4.3 HVS 模型

由上图可以看出 HVS 模型的明度(V)与反映色彩本质特性两个参数色度(H)和饱和度(S)是独立的。

RGB 模型和 HVS 模型都是构造一个颜色空间的关系模型。HVS 模型是将 RGB 模型同过一个非线性坐标变换得到。它们之间的转换关系如下:

$$V = \max(R, G, B) \quad (4.2)$$

$$S = \begin{cases} (V - \min(R, G, B)) * 255 / V, & \text{if } (V \neq 0) \\ 0, & \text{otherwise} \end{cases} \quad (4.3)$$

$$H = \begin{cases} (G-B)*60/S, & \text{if } (V=R) \\ 180-(B-G)*60/S, & \text{if } (V=G) \\ 240+(R-G)*60/S, & \text{if } (V=R) \end{cases} \quad (4.4)$$

if ($H < 0$) then $H = H + 360$, (将 $H < 0$ 的色度转换到 $0 \sim 360$ 范围内。)

颜色特征是用反映色彩本质特性的色度分量来进行聚类分析。在 HVS 模型中明度分量(V)与反映色彩本质特性的色度(H)和饱和度(S)是独立的, 而 RGB 模型三个分量同时要考虑。因此 HVS 模型相对于 RGB 模型不仅计算量小, 而且受光照的影响也较小。

4.2.2 矩特征

矩特征是描述图像的一些统计特征。视频图像区域的矩是用所有属于区域内的点计算出来的。矩特征对运动目标平移、旋转和尺度变换都有很好的鲁棒性, 而且不易受噪声等的影响。如果一个数字图像 $f(x, y)$ 分段连续且只有 XY 平面上的有限个点不为零, 那么它的各阶矩存在, 则可以选择矩特征运动目标的不变特征来实现对其跟踪。

我们定义数字图像 $f(x, y)$ 的 $p+q$ 阶矩为:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad (4.5)$$

数字图像 $f(x, y)$ 的 $p+q$ 阶中心矩定义为:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (4.6)$$

其中, $(\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}})$ 是重心坐标。

$f(x, y)$ 的 $p+q$ 阶中心矩归一化为:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad (4.7)$$

其中, $\gamma = \frac{p+q}{2} + 1, p+q = 2, 3, \dots$

以下 7 个对平移、旋转和尺度变换不变的矩是由归一化的二阶和三阶中心矩得到的:

$$\phi_1 = \eta_{20} + \eta_{02} \quad (4.8)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (4.9)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (4.10)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (4.11)$$

$$\begin{aligned} \phi_5 = & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (4.12)$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (4.13)$$

$$\begin{aligned} \phi_7 = & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (4.14)$$

矩特征是一种全局的特征，如果有运动目标冲突、遮挡或信息损失等现象，就会造成不变矩的计算不准。所以矩特征应用面很狭窄。

4.2.3 直方图

直方图也是描述图像的一些统计特征的一种统计结果，反映了图像中像素的分布特性，如：方差、互相关值、均值等。这个统计特征是 Prewitt 于二十世纪六十年代中期提出，主要目的是完成对图像的分割^[51]。其基本思想是：当一幅图像中的背景与目标存在差异（例如背景是浅色的，目标是深色的），则这幅图像的灰度直方图中背景的大量灰度级产生了直方图上的右峰，目标像素产生了直方图上的左峰，如图 4.4。

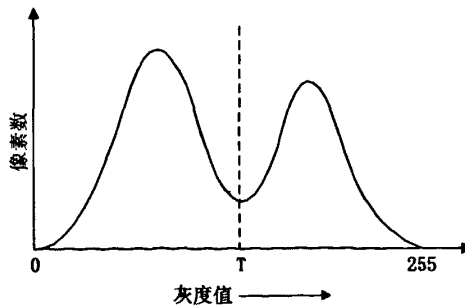


图 4.4 直方图

从图 4.4 可知道，在图像背景与运动物体的临界处灰度级的像素点数目不多，即在两峰之间产生了谷。这个谷就是将图像背景与运动物体的分割阈值。此外，若在跟踪算法中引用的话，就可以利用直方图作为判断两帧图片匹配程度。例如基于颜色直方图特性匹配跟踪算法的思想就是：首先选定运动目标的模板，再统计此模板的颜色直方图，

得到其概率,然后将其反投影到原来图像;再采用一些搜索算法准确的确定运动目标在下一帧中位置。这种方法针对运动目标发生各类形变都具有很强的鲁棒性。但这种方法缺点是:当图像背景中具有很多与运动目标的颜色相似的像素点或运动目标比较小时,该方法就不在适用了。但目前广泛的应用在基于颜色直方图统计特征的运动目标跟踪算法当中。本文这里利用图像的颜色统计直方图作为整个搜索匹配的特征。

4.3 均值偏移算法(MeanShift)搜索匹配算法

目前,比较常用的搜索匹配算法包括有归一化相关法 NC(Normalized Correlation)^[45]、绝对平衡搜索法 ABS(Absolute Balance Search)^[46]和均值偏移搜索法(MeanShift)^[47]等。NC 算法是在候选匹配图像中寻找与模板图像互相关值最大的窗口,即为搜索到的目标在一帧图像中的坐标。此算法对线性变换有很强的鲁棒性。可是当互相关值都非常大时,就很难准确的确定目标的新位置。此外 NC 的计算量很大,不易满足实时性。而 ABS 是一种运算最简单的搜索法。其基本思想是:先设定一个阈值,再计算候选匹配图像上搜索窗口与已选定目标模板之间的灰度差值,最后将差值与预先设定的阈值做比较来判决是否匹配。但此方法具有两个明显的缺点:一、阈值难以设定,因为每一幅候选匹配图像和图像模板都有不同的灰度值;二、使用面狭窄,此算法对线性变换很敏感。当候选匹配图像或图像模板中任何一个的灰度值发生线性变化时,就搜索不到匹配模板。MeanShift 是一种高效的模式匹配算法。它利用梯度优化方法实现快速目标定位,能够实时跟踪非刚性目标,对目标的变形、旋转等有较好的适用性。现在它已经被广泛地应用在很多目标跟踪领域。鉴于此,本文将着重介绍这种计算局部最优解的实用算法——均值偏移跟踪算法(MeanShift)。

4.3.1 MeanShift 的原理

均值偏移算法最早是由 Fukunaga 和 Hostetler^[52]于 1975 年提出的,其后经过 Chneg.Y^[53]、Bardks^[54]、Isard 和 Blake^[55]等学者不断研究和改进,推广了其应用范围。直到 Comaniciu 于 1999 年发表了一篇“MeanShift analysis and application”,使得 MeanShift 算法应用于对图像进行滤波和分割,也应用到了非刚性运动目标在视频序列的跟踪领域。随后 Comaniciu 又对 MeanShift 算法进行进一步的优化,增强了其搜索能力^[56]。使其优势在目标跟踪领域逐渐体现出来。近些年,国内也有一定研究^[57,58]。在本文中,又在此算法基础上结合了基于 Bhattacharyya 系数的相似性测度,作为对目标模板和候选目标之间的距离作为相似性判断,完成最优的特征匹配。

均值偏移算法是一种在特征空间中求解概率密度极值的优化算法。通过对运动目标和非运动目标赋予不同的权值，使运动目标区域成为密度极值区，从而将目标跟踪同均值漂移算法联系起来。具体原理如下：

(1)选择目标模板：在一个选定区域来确定目标模板。为了消除其尺寸不同的影响，对其做归一化处理来，即，归一化后模板的长为 h_x ，宽为 h_y 。由于远离目标模板中心的像素容易受到其他目标冲突或背景像素的影响，所以为了提高搜索跟踪的鲁棒性，还需要对目标模板区域的像素设置权值，即，目标模板中心的像素邻近处设置较大的权值，远离目标模板中心的像素设置较小的权值。

设目标模板区域的归一化像素位置为 $\{\bar{x}_i^*\}_{i=1 \dots n}$ 。定义 $k(x)$ 为核函数，即权函数。这个函数是一个凸型的、各向同性的和单调递减的函数。它的主要作用给靠近运动目标模板中心的像素设置较大的权值，反之设置较小的权值。同时定义 $b: R^L \rightarrow \{1 \dots m\}$ 为图像可处的灰度值索引函数，即 $b(\bar{x}_i^*)$ 表示 \bar{x}_i^* 处的像素灰度值。则基于图像灰度特征 $u=1, \dots, m$ 的目标模板灰度概率函数为：

$$\hat{q}_u = C \sum_{i=1}^n k\left(\left\|\bar{x}_i^*\right\|^2\right) \delta\left[b\left(\bar{x}_i^*\right)-u\right] \quad (4.15)$$

其中， $\delta(\bullet)$ 为 Kronecker Δ 函数， C 为归一化常数。它们定义分别为：

$$\delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad (4.16)$$

$$C = \frac{1}{\sum_{i=1}^n k\left(\left\|\bar{x}_i^*\right\|^2\right)} \quad (4.17)$$

(2)设定候选目标的表示：设检测算法得到位置 \bar{y} 为候选目标区域中心位置，候选目标区域归一化位置为 $\{\bar{x}_i\}_{i=1 \dots n}$ 。则可得基于图像灰度特征 $u=1, \dots, m$ 的候选目标灰度概率函数为：

$$\hat{p}_u = C_h \sum_{i=1}^n k\left(\left\|\frac{\bar{y}-\bar{x}_i}{h}\right\|^2\right) \delta[b(\bar{x}_i)-u] \quad (4.18)$$

其中， $k(x)$ 为引入的核函数，其尺度为 h 。 C_h 为归一化常数，可表示为：

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k \left(\left\| \frac{\bar{y} - \bar{x}_i^*}{h} \right\|^2 \right)} \quad (4.19)$$

(3)判断候选目标和目标模板之间的相似性：本文采用基于 Bhattacharyya 系数的相似性测度，其基本思想是：以候选目标和目标模板间的距离作为相似性判断依据。定义这两个模板离散分布之间的距离为：

$$d(\bar{y}) = \sqrt{1 - \rho[\hat{p}(\bar{y}), \hat{q}]} \quad (4.20)$$

则 Bhattacharyya 系数可由目标模板分布 \hat{q} 和候选目标分布 $\hat{p}(\bar{y})$ 表示为：

$$\hat{\rho}(\bar{y}) \equiv \rho[\hat{p}(\bar{y}), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\bar{y}) \hat{q}_u} \quad (4.21)$$

(4)目标在当前帧的定位。最佳匹配的目标就是候选目标中与目标模板 Bhattacharyya 系数最大的模板。而由式(4.20)可以看出 Bhattacharyya 系数最大化也是 $d(\bar{y})$ 的最小化。所以定位过程就是从上一帧的目标模板位置 \bar{y}_0 处开始，在当前帧中搜索与之匹配的新目标位置。通过式(4.15)和式(4.18)得到目标模板的概率密度 \hat{q} 以及当前帧 \bar{y}_0 处候选目标的概率密度 $\{\hat{p}_u(\hat{y}_0)\}_{u=1 \dots m}$ ，然后对 $\hat{\rho}(\bar{y})$ 在 $\hat{p}_u(\hat{y}_0)$ 处使用 Taylor 公式展开，得：

$$\rho[\hat{p}(\hat{y}), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} + \frac{1}{2} \sum_{u=1}^m \hat{p}_u(\bar{y}) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \quad (4.22)$$

代入 \hat{p}_u 得到变形公式为：

$$\rho[\hat{p}(\hat{y}), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} + \frac{C_h}{2} \sum_{u=1}^m \omega_i k \left(\left\| \frac{\bar{y} - \bar{x}_i^*}{h} \right\|^2 \right) \quad (4.23)$$

其中，

$$\omega_i = \sum_{i=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \delta[b(\bar{x}_i) - u] \quad (4.24)$$

式(4.23)中的第二项是在当前帧中利用核函数 $k(x)$ 和图像像素的加权值计算得到概率密度估计。从式可以看出第一项与 $d(\bar{y})$ 最小化无关。因此只需对第二项做最大化处理就可得最小距离。采用均值偏移过程检测，逐步从 \bar{y}_0 处不断递归的计算出新的目标

位置 \bar{y}_1 :

$$\bar{y}_1 = \frac{\sum_{i=1}^m \bar{x}_i \omega_i g\left(\left\|\frac{\hat{\bar{y}}_0 - \bar{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^m \omega_i g\left(\left\|\frac{\hat{\bar{y}}_0 - \bar{x}_i}{h}\right\|^2\right)} \quad (4.25)$$

4.3.2 MeanShift 在运动目标跟踪中的应用

基于均值偏移(MeanShift)算法应用到运动目标跟踪的实现流程框图如图 4.5 所示。

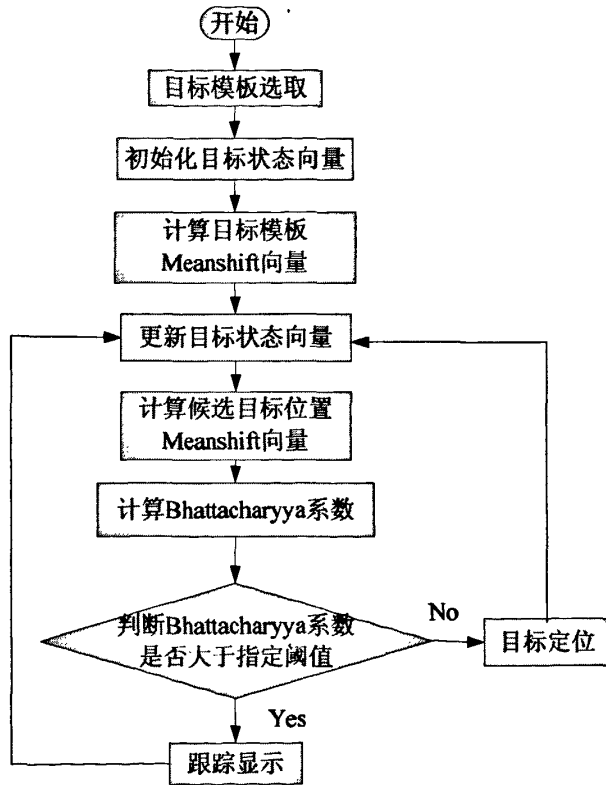


图 4.5 MeanShift 跟踪算法的流程图

具体实现步骤的描述如下:

Step1、对运动目标提取有效特征,并在前一帧建立目标模板,并利用公式(4.15)计算其在 \bar{y}_0 处的概率密度 $\{\hat{q}_u(\bar{y}_0)\}_{u=1 \dots m}$, 指定一个小的正常数 ε ;

Step2、确定目标在当前帧中的搜索范围或方向,利用公式(4.18)计算在当前帧 \bar{y}_0 处的候选目标的概率密度 $\{\hat{p}_u(\bar{y}_0)\}_{u=1 \dots m}$ 和 Bhattacharyya 系数, $\rho[\hat{p}(\bar{y}_0), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\bar{y}_0) \hat{q}_u}$ 。

Step3、通过公式(4.24)得到加权 $\{\omega_i\}_{i=1\dots m}$ 后，再利用(4.25)获得候选目标的下一个新位置 \bar{y}_1 。

Step4、再利用公式(4.18)和公式(4.23)计算新位置 \bar{y}_1 处的候选目标的概率密度，

$$\{\hat{p}_u(\hat{y})\}_{u=1\dots m} \text{ 和 Bhattacharyya 系数, } \rho[\hat{p}(\bar{y}_1), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\bar{y}_1) \hat{q}_u} ;$$

Step5、判断 $\rho[\hat{p}(\bar{y}_1), \hat{q}] < \rho[\hat{p}(\bar{y}_0), \hat{q}]$ ，若满足则 $\bar{y}_1 \leftarrow \frac{1}{2}(\bar{y}_0 + \bar{y}_1)$ ，然后判断 $\|\bar{y}_1 - \bar{y}_0\| < \varepsilon$ ，如满足条件则终止循环判断；否则 $\bar{y}_0 \leftarrow \bar{y}_1$ ，返回 step2 继续寻找满足条件的候选目标位置。

一般为了简化运算过程我们可以选择 Epanechnikov 函数为核函数，如下式：

$$k(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2), & x < 1 \\ 0, & \text{otherwise} \end{cases} \quad (4.26)$$

这样的话 $g(x)$ 就是一个常数，则公式(4.25)可以简化为：

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} \hat{x}_i \omega_i}{\sum_{i=1}^{n_h} \omega_i} \quad (4.27)$$

4.4 目标运动估计

现在很多监控系统都是仅仅停留在人工辅助操作的可视化水平的现象^[59,60]，但我们期望通过自动控制摄像头转动，使运动目标总是能处于摄像头图像的中心或在图像某个指定范围内运动。针对此要求，本文研究了视频序列中运动目标跟踪算法和自动控制云台带动摄像机转动的方法，实现了系统的全自动化来减轻人的负担。在运动目标跟踪方面，本文在上一节已经介绍了一种均值偏移跟踪算法，但这种算法是一种被动跟踪算法。若仅仅利用均值偏移跟踪算法，由于摄像头转动的过程中，运动目标也在移动，所以当摄像头中心转动了指定的角度后，并不能使运动目标始终处于图像的指定范围，会造成跟踪滞后^[61]。能否成功将运动目标一直控制在某一指定范围内的关键在于如何能够及时预估计出下一个时刻目标出现的最大可能位置，以便于摄像头及时调整角度，快速跟踪目标。为了解决此问题，本节将介绍一种卡尔曼滤波预测算法(Kalman)，主动预测物体在下一帧的位置。

4.4.1 Kalman 滤波器的原理

卡尔曼预测器是一种递推估计器，是一个对动态系统的状态序列进行线性最小方差误差估计的算法。它不仅计算量小，而且它可以以任意一点作为起点来观测和估计目标的运动^[62]。它的系统是用状态方程和观测方程来表示的，原理如下：

系统的状态方程：

$$X_{k+1} = Ax_k + GW_k \quad (4.28)$$

其中， X_k 是 $n \times 1$ 维状态向量， A 是 $n \times n$ 状态转移矩阵， G 为 $n \times m$ 输入转移矩阵， W_k 是 $m \times 1$ 维干扰随机向量，它服从 $N(0, Q)$ 的高斯白噪声， Q_k 为动态噪声的 W_k 协方差矩阵，其式如下：

$$Q_k = E\{W_k W_k^T\} \quad (4.29)$$

系统的观测方程：

$$Z_k = H_k x_k + V_k \quad (4.30)$$

其中， Z_k 是观测到的系统状态的 $m \times 1$ 维状态向量； H_k 是 $m \times n$ 观测矩阵； V_k 是 $m \times 1$ 维观测噪声向量，观测噪声 V_k 和动态噪声 W_k 为互不相关的零均值的正态白噪声序列。我们令 R_k 为观测噪声 V_k 的协方差矩阵，其式如下：

$$R_k = E\{V_k V_k^T\} \quad (4.31)$$

Kalman 滤波器算法的方程如下：

$$\hat{X}(k+1|k) = A\hat{X}(k|k) \quad (4.32)$$

$$\hat{Z}(k+1|k) = H\hat{Z}(k+1|k) \quad (4.33)$$

$$V(k+1) = Z(k+1) - \hat{Z}(k+1|k) \quad (4.34)$$

$$P(k+1|k) = AP(k|k)A' + GQ(k)G' \quad (4.35)$$

$$S(k+1|k) = HP(k+1|k)H' + R(k) \quad (4.36)$$

$$K(k+1) = P(k+1|k)H'S(k+1|k)^{-1} \quad (4.37)$$

$$\hat{X}(k+1|k+1) = A\hat{X}(k|k) + K(k+1)V(k+1) \quad (4.38)$$

$$P(k+1|k+1) = [1 - K(k+1)H]P(k+1|k) \quad (4.39)$$

其中，式 $\hat{X}(k+1|k)$ ， $\hat{Z}(k+1|k)$ 分别为动态系统状态预测值和观测预测值。 $V(k+1)$ 为观测预测值与真实值之间的差值。 $S(k+1|k)$ 为 $V(k+1)$ 的方差矩阵； $P(k|k)$ 为状态预测值的方差矩阵； $K(k+1)$ 为增益矩阵。

4.4.2 Kalman 滤波器在目标运动估计中的应用

上一小节我们介绍到 Kalman 滤波器可以利用观测噪声和系统噪声的统计特性，以系统观测值作为输入，以需要估计的值为输出，对运动目标的当前运动状态作出估计，且对未来运动状态作出预测。因此本文利用 kalman 滤波器，根据运动目标的速度信息主动预测物体在下一帧的位置。其基本思想是：通过分析前面的连续几帧中运动目标运动的轨迹来建立二维平面运动目标跟踪模型，并列出状态方程和观测方程。利用前一时刻对当前时刻的预测值和当前的观察值来更新对状态变量的估计，从而求出下一时刻的预测值，实现递归的预测。

在视频图像中只能观测到目标的位置，因此我们定义 Kalman 滤波器观测状态向量为 Y_k ，即 (xw_k, yw_k) ；定义系统状态为 x_k ，即四维向量 (xs_k, ys_k, xv_k, yv_k) 分别表示目标的在 x 轴和 y 轴上的位置和速度。由于连续两帧时间间隔比较短，则运动目标的运动的状态变化很小，所以我们可以设定运动目标在单位时间内是匀速运动。

定义状态转移矩阵 A 为：

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.40)$$

观测矩阵 H_k 为：

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (4.41)$$

动态噪声 w_k 的协方差矩阵为：

$$Q_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.42)$$

观测噪声 V_k 的协方差矩阵为：

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.43)$$

Kalman 算法完成运动估计的具体实现步骤如下：

- (1) 根据先验知识设定 Kalman 滤波器的状态初始向量。
- (2) 根据式(4.32) 进行预估 k 时刻的状态信号 $\hat{X}(k|k-1)$ 。
- (3) 根据当前的观察值和上一帧的预测值求运动目标的最优预测值 $\hat{X}(k|k)$ 。
- (4) 求修正后的最小预测均方误差矩阵 $P(k|k)$ 。
- (5) 根据当前帧的最优预测值预测下一帧位置 $\hat{X}(k+1|k)$ 。
- (6) 根据预测值和摄像头中心值求出摄像头所要转过的距离。若需要转动的距离过小，则暂不转动。否则摄像头根据所求的值转动相应的角度。
- (7) 求当前的摄像头相对于最初静止时移动的距离。
- (8) 通过式(4.35)求得最小预测均方误差矩阵 $P(k|k-1)$ 。
- (9) 利用式(4.37)得到滤波增益 $K(k)$ 。

如此这样循环执行。需要注意的是在摄像头转动的过程中，绝对坐标与相对坐标的转化。其转化过程为：如图 4.6 中对于 X 坐标，摄像头在静止时图像的最左边的绝对位置为绝对坐标的 0 点，即为 J_0 。在摄像头转动后图像的最左边的绝对位置为相对坐标的 0 点，即为 x_0 ，则摄像头图像相对于静止时移动的距离为 $dx = x_0 - J_0$ ，同时也是图像的最左边的绝对坐标，此时摄像头中心点的相对位置为： $x_c = \frac{WIDTH}{2}$ 。其中， $WIDTH$ 为图像的宽度；而绝对位置为 $x_r = x_c + dx$ 。摄像头经过多次运动后，最终的相对于静止时移动的距离为 $s_{dx} = s_{dx} + dx$ 。其中， dx 为每次移动的距离（向右转时为正值，向左转时为负值）。经过多次摄像头转动后绝对坐标是相对坐标和转移距离的和。

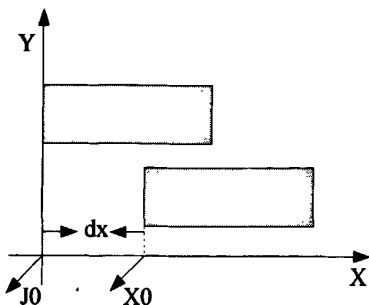


图 4.6 绝对坐标与相对坐标的转换

4.5 摄像机的控制

在整个监控系统中，目标是处于运动状态中的，若用固定摄像头式的监控系统，目标易走出视场范围，而造成丢失。为了能将目标控制在视场中心范围，需要人工辅助或摄像机被固定在云台之上，使摄像机随同云台的运动来完成。本系统就是通过控制云台的运动来实现。

4.5.1 云台控制参数的计算

在视频序列中选定锁定运动目标的合适位置。在系统跟踪中，假设运动目标 t 时刻在图像中的坐标为 (xw_t, yw_t) 。利用我们上面介绍的 Kalman 滤波器进行运动估计得到目标在 $t + \Delta t$ 时刻的质心位置 $(xw_{t+\Delta t}, yw_{t+\Delta t})$ ，即在下一帧图像中坐标。判断 $t + \Delta t$ 时刻目标的质心位置是否即将离开或已经离开图像上指定的视窗范围。若是，则通过计算机控制云台运动，进而带动摄像机沿目标运动方向转动及时把目标控制到图像上指定位置 $(xw'_{t+\Delta t}, yw'_{t+\Delta t})$ 。摄像机转动的幅度与目标的平移量存在如下关系：

从 $(xw_{t+\Delta t}, yw_{t+\Delta t})$ 到 $(xw'_{t+\Delta t}, yw'_{t+\Delta t})$ 的位移量为：

$$\Delta x = xw_{t+\Delta t} - xw'_{t+\Delta t} \quad (4.44)$$

$$\Delta y = yw_{t+\Delta t} - yw'_{t+\Delta t} \quad (4.45)$$

其中， Δx 是在水平方向上的平移量； Δy 是在垂直方向上的平移量。

假设摄像机视野水平视角为 θ ，垂视角为 ϕ ，摄像机焦距为 f （未知量），图像水平方向长度为 w ，垂直方向长度为 h 。

在水平方向，图像成像在焦平面上：

$$\begin{cases} f \times \text{tg}(\theta/2) = w/2 \\ f \times \text{tg}(\Delta\theta) = \Delta x \end{cases} \quad (4.46)$$

得：

$$\Delta\theta = \arctg\left(\frac{2\Delta x}{w} \text{tg}(\theta/2)\right) \quad (4.47)$$

在垂直方向，图像成像在焦平面上：

$$\begin{cases} f \times \text{tg}(\phi/2) = h/2 \\ f \times \text{tg}(\Delta\phi) = \Delta y \end{cases} \quad (4.48)$$

得：

$$\Delta\phi = \arctg\left(\frac{2\Delta y}{h} \text{tg}(\phi/2)\right) \quad (4.49)$$

由于连续两帧时间间隔比较短，运动目标的运动的状态变化很小，即转动的角度比较小。则可近似的认为：

$$\text{tg}(\theta/2) = \theta/2 \quad (4.50)$$

$$\text{tg}(\phi/2) = \phi/2 \quad (4.51)$$

则式(4.47)和式(4.49)可简化为如下式：

$$\begin{cases} \Delta\theta = \frac{\Delta x}{w} \theta \\ \Delta\phi = \frac{\Delta y}{h} \phi \end{cases} \quad (4.52)$$

通过上式就可以实现了运动目标平移量到摄像机旋转角度的转换。

4.5.2 云台控制实现

经计算机利用算法运算后，根据云台解码器的编码向云台发出指令进而实现了对云台的控制。系统所采用的云台编码表，见表 4.1：

表 4.1 云台编码表

指令码	说明
6AH	K2 断开
69H	K2 接通
68H	K1 断开
67H	K1 接通
4FH	索取报警状态
4EH	聚焦后移
4DH	聚焦前移
4CH	长焦
4BH	短焦
4AH	光圈减小
49H	光圈增大
45H	云台抬头
44H	云台低头
43H	云台逆时针转动
42H	云台顺时针转动
41H	云台或镜头的任何动作停止
40H	云台自动左右转动命令

云台控制编程的具体步骤如下：

Step1、先创建一个访问 COM 口的线程。

Step2、接着对 COM 的各个参数进行设置。

Step3、初始化一个云台控制命令字符串，用以标示发送命令的控制字。

Step4、将命令字符串的不是标示字的部分改成要发送的命令字向 COM 口发送控制。

4.6 实验结果与数据分析

为了验证本章所提及算法的可行性，采用本实验室提供的硬件设备做了两组实验。

第一组实验，运动目标是被旋转机带着运动的一个小物体。采用单目摄像机完成对运动目标的自动跟踪。运动目标与视频序列的参数见表 4.2。采用本章算法完成对其自动跟踪结果如图 4.7。从图中可以看出本算法不仅可以成功跟上目标，而且能较准确预测到目标在下一帧图像中的位置，进而完成对摄像机的自动控制，实现了把目标锁定到指定视窗范围。本组的实验结果数据分析，见表 4.3 。

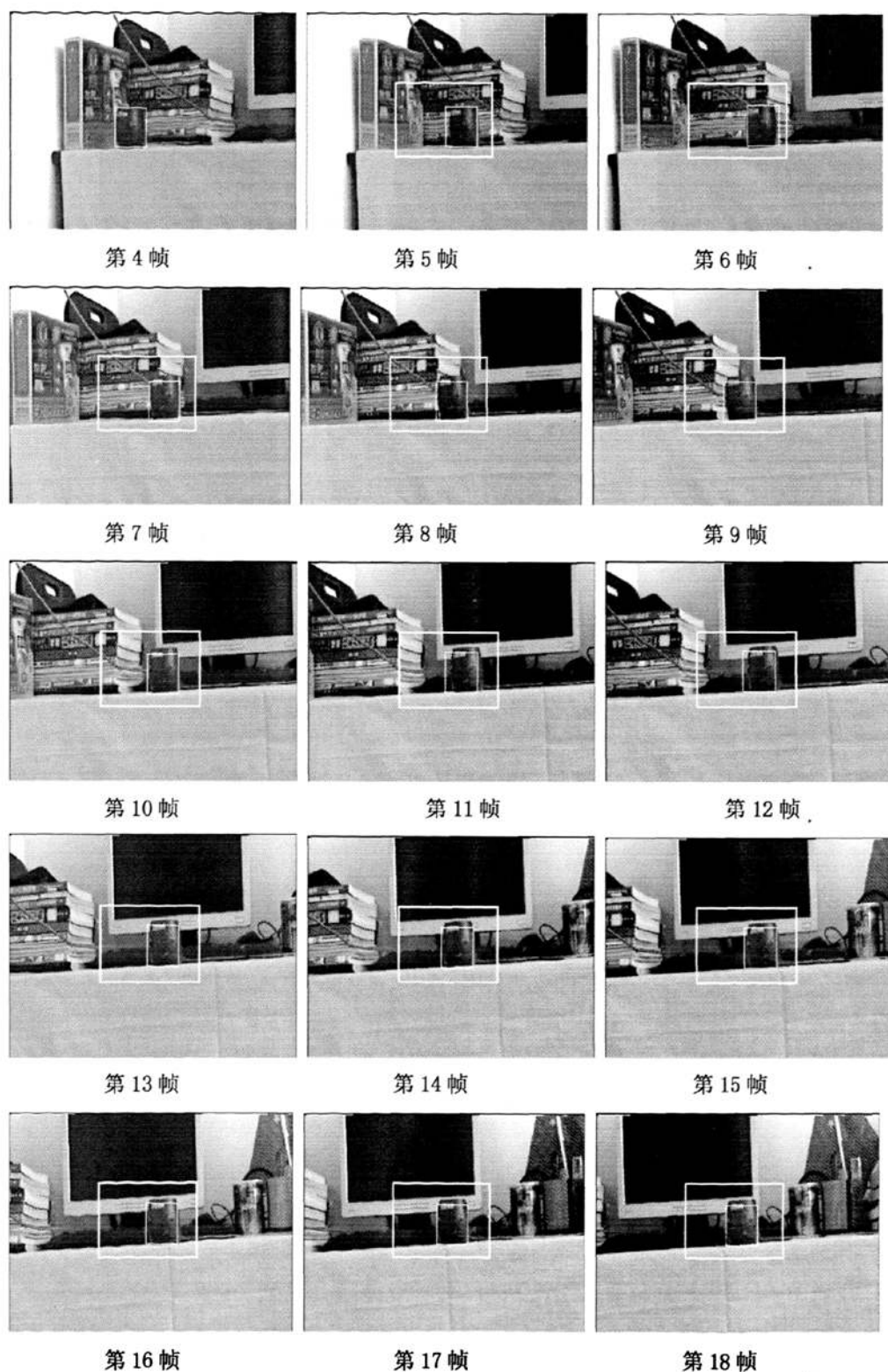


图 4.7 第一组运动目标自动跟踪的结果

表 4.2 运动目标与视频序列的参数表

目标个数	目标大小 (像素)	背景大小 (像素)	帧数 (帧)	算法总耗时间 (s)	目标初始速度 (像素/s)	摄像机离目标的 距离(米)
1	67×85	480×640	100	687	(0,23)	2

表 4.3 第一组实验结果的数据分析表

时刻(帧)	Kalman 预测世界位置 (像素)	摄像机转动后的目标几何 位置(像素)	运动目标偏离图像中心位置的 距离(像素)
4		(259, 294)	(19,-26)
5	(253.0088, 385.6260)	(250, 377)	(10,57)
6	(252.8461, 367.3748)	(256, 349)	(16,29)
7	(250.7248, 357.5387)	(255, 350)	(15,30)
8	(249.5136, 360.1077)	(253, 356)	(13,36)
9	(245.9949, 342.5240)	(254, 337)	(14,17)
10	(239.9552, 364.6606)	(248, 347)	(8,27)
11	(230.8105, 368.0047)	(244, 348)	(4,28)
12	(230.2872, 368.8391)	(247, 340)	(7,20)
13	(232.2674, 360.6210)	(243, 344)	(3,24)
14	(227.7464, 365.7488)	(242, 351)	(2,31)
15	(231.0946, 374.9808)	(249, 344)	(9,24)
16	(231.6727, 368.9535)	(249, 337)	(9,17)
17	(231.0200, 362.5783)	(246, 338)	(6,18)
18	(221.1957, 361.1686)	(244, 338)	(4,18)

第二组实验,运动目标仍然是被旋转机带着运动的小物体。在这组实验里运动物体的速度得到加倍。运动目标与视频序列的参数见表 4.4。采用本章算法完成对其自动跟踪结果如图 4.8。从结果图中可以看出前几帧运动目标虽然能较成功的跟上,但是从第 7 帧开始,由于摄像机转动出现滞后性,导致运动目标脱离了视窗指定的范围,直到最后运动目标从监控的视场中的走丢。这种现象是主要由两方面原因造成的:一方面是实验硬件设备的限制;另一方面,也是关键的原因,运动目标估计算法的时效性较差。随着目标的运动速度加快,本章的运动目标估计算法还是受到限制的。本组实验结果数据分析见表 4.5。

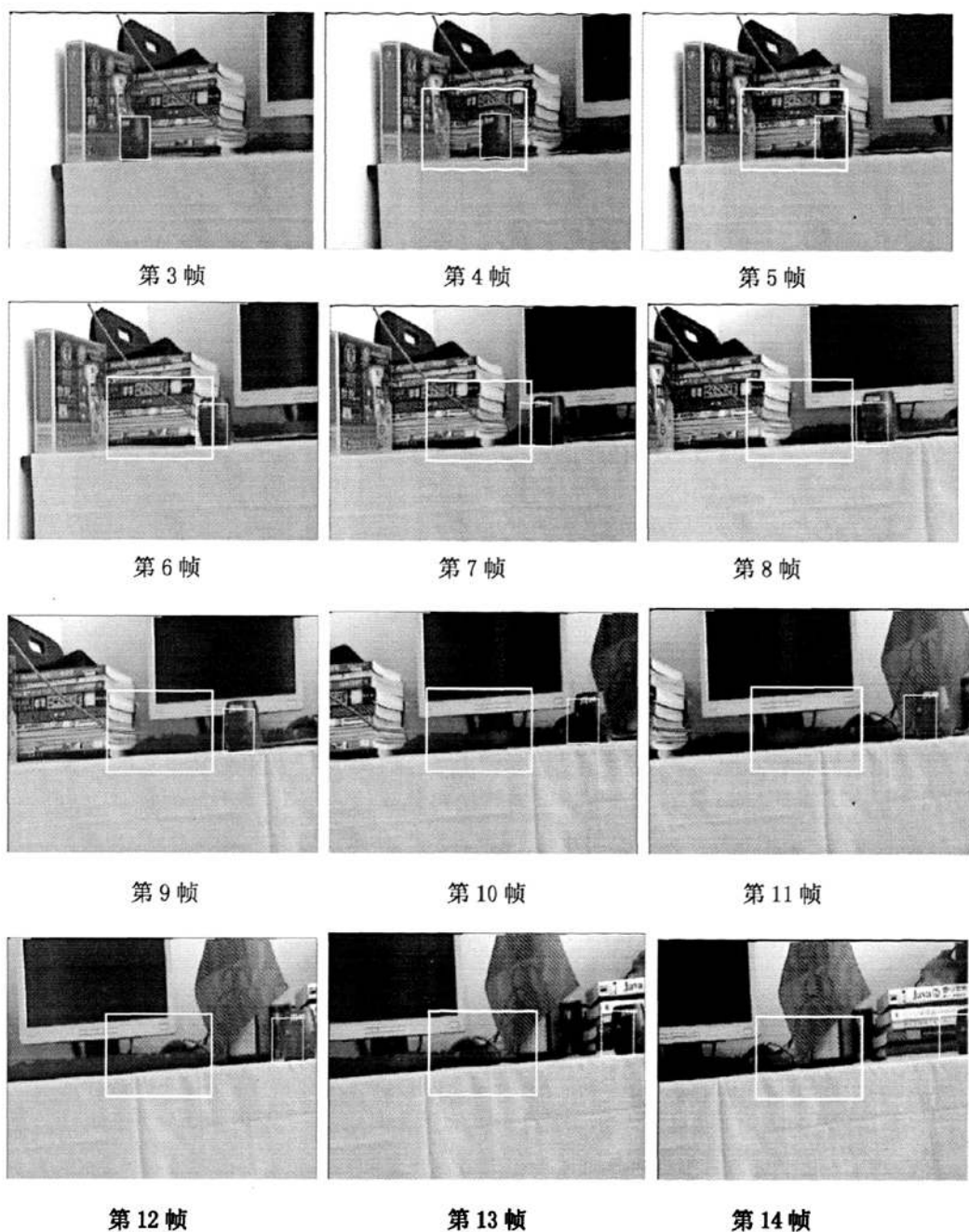


图 4.8 第二组运动目标自动跟踪的结果

表 4.4 运动目标与视频序列的参数表

目标个数	目标大小 (像素)	背景大小 (像素)	帧数 (帧)	算法总耗时间 (s)	目标初始速度 (像素/s)	摄像机离目标的距离 (米)
1	67×85	480×640	50	371	(0,65)	2

表 4.5 第一组实验结果的数据分析表

时刻(帧)	Kalman 预测世界位置 (像素)	摄像机转动后的目标几何 位置(像素)	运动目标偏离图像中心位置的 距离(像素)
3		(259, 282)	(19,-38)
4	(260.7310, 415.5851)	(255, 425)	(15,105)
5	(258.2841, 444.0109)	(253, 423)	(13,103)
6	(252.2199, 357.5387)	(247, 437)	(7,117)
7	(238.3616, 487.5130)	(248, 467)	(8,147)
8	(233.1989, 503.8846)	(236, 487)	(-4,167)
9	(228.6789, 510.5540)	(234, 517)	(-6,197)
10	(214.4104, 562.9082)	(218, 541)	(-22,221)
11	(198.0096, 598.1716)	(210, 581)	(-30,261)
12	(192.3885, 612.8028)	(202, 586)	(-38,266)
13	(181.9117, 643.3535)	(197, 615)	(-43,295)
14	(180.0131, 645.1260)	(196, 615)	(-44,295)

4.7 本章小结

针对当前很多监控系统中的运动目标跟踪模块都停留在人工辅助操作的可视化水平的现象, 本文在本章主要介绍了一种运动目标自动跟踪方法。此方法不仅无需人工参与, 而且也可以一直把目标控制到指定视频图像的中心范围, 不易丢失目标。

比较了几种典型的跟踪算法后, 本文采用均值偏移算法(MeanShift)实现运动目标的跟踪。但在摄像机与运动目标同时运动的情况下, 仅仅使用 MeanShift 跟踪算法, 会造成跟踪滞后问题, 因此本文引用 Kalman 滤波器算法, 来完成对运动目标的当前运动状态估计, 且利用运动方向的速度信息主动预测物体在下一帧的位置。根据预测值控制云台运动, 云台带动摄像头转动指向运动目标, 消除了跟踪滞后现象。在本章的最后又介绍了如何实现摄像机的自动转动。最后又利用一组实验数据说明此方法的可行性。

第五章 系统实验平台

系统平台的搭建结构都是依据所研究和应用的对象而言的。现在大体可分为单通道和多通道视频图像采集系统搭建。基于单通道视频图像采集系统获取单视场视频图像，一次只能监视一个场景，这种系统是最基本、最主要的视频图像采集。多通道视频图像采集系统可以同时获取和融合处理多个场景的图像序列数据，这样避免出现监视死角区。本文依据现有的实验条件组建了一个单通道视频图像采集系统，下面就对该系统硬件系统结构与应用系统软件开发作以分析。

5.1 系统硬件实验平台

本文硬件系统主要包括：计算机、云台、解码器、摄像头、图像采集卡与计算机接口(USB1.0 接口)。其框图如图 5.1 所示。

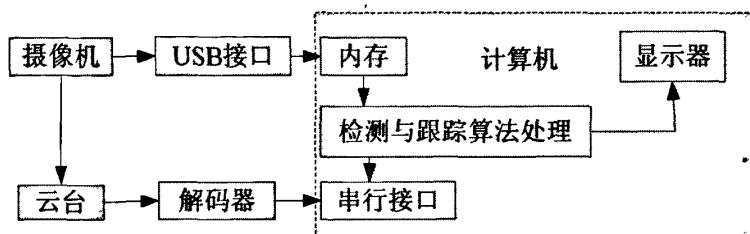


图 5.1 系统实验平台的硬件组成结构

系统以摄像机、图形采集卡为采集部分，完成运动目标视频图像的采集；计算机作为控制中枢，即中央控制单元，处理包含运动目标的视频图像序列，并分析目标的运动状况，向云台发出控制指令；云台为控制终端，云台接收到计算机发送的指令带动摄像机运动，跟踪运动目标，将其控制在视窗指定中心范围。下文将对组成实验平台的各个硬件的参数特征作以介绍与说明。

5.1.1 摄像机主要参数及选择

在本文，图像采集设备是采用市面上一种通用的摄像头——Unibit 公司的 U-T527 摄像头。这种摄像头不仅价格低、而且可以直接插入 USB 接口，拆装简单，降低了监控设备成本进而提高使用性。其技术规格表，见表 5.1。

表 5.1 U-T527 摄像机技术规格表

技术参数	说明
型式	美光新一代 35 万像素高速 CMOS 传感器
分辨率	640*480
速度	320*24015—35Pfs; 640*4805—15Pfs
信噪比	优于 48dB
动态范围	优于 72dB
视角	45 到 60 度
接口	USB1.1, 热插拔
成像距离	3cm—无限远(常规); <3cm(微距)
支持格式	Vrwr, Directshow

5.1.2 视频图像采集卡的选择

视频图像采集卡可被认为是 CCD 摄像机与计算机的接口,是将来自 CCD 摄像机的模拟信号转换成数字信号输入到计算机中进行处理。本文选择的视频图像采集系统是 Matrox 公司的 Matrox Meteor 系列产品中的一款性能优异的图像采集卡——Matrox Meteor II Standard, 它是一款标准型彩色/黑白实时图像采集卡。其结构如图 5.2。

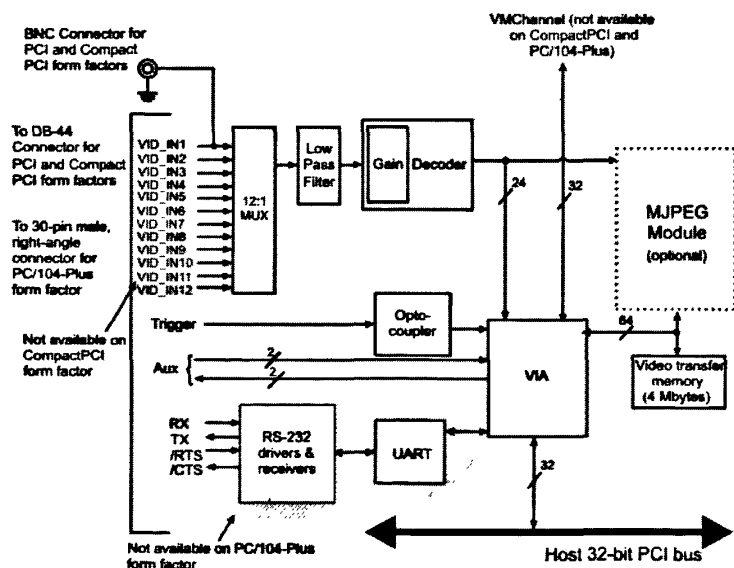


图 5.2 图像采集卡结构

Matrox Meteor II Standard 图像采集卡的工作过程是首先图像模拟信号输入采集卡；再经过多路开关、低通滤波器、视频解码器、转换器转换为数字信号；然后把数字信号通过 PCI 传送到计算机系统主内存，一般传输率可到 132Mbps；最后经过软件处理后显示该帧。这种采集卡的特点是：它支持 12 路软件可选通道单色 RS-170/CCIR 或混合

NTSC/PAL 视频制式，6 路 NTSC/PAL 视频制式的 Y/C 颜色模式视频；支持 PCI、PC/104 -plus 等多种总线。它也有一个 RS-232 串行接口，可以实现对摄像机的控制，可编程或自动增益控制；它拥有 4MB 的图像传输缓存，可以选择 MJPEG 实时图像压缩模块，以利于其的传输和存储；另外，它本身提供 5V、6V 直流电源输出，这样的话就不用外接电源来驱动摄像机的工作。

5.1.3 计算机系统的选择

由于 U-T527 型摄像头对计算机系统的要求比较高，特别是对计算机的处理速度和内存的要求要求比较高，因此本文采用的计算机系统配置如下：

- (1)CPU: Inter (R) Core (TM)2 6300
- (2)硬盘: 80G
- (3)内存: 2G
- (4)USB 接口
- (5)显卡: Intel(R) Q965/Q963 Express Chipset Family

5.1.4 云台技术参数及特点

本系统在自动跟踪部分是通过计算机控制云台运动，云台带动摄像头自动转动来实现。因此云台的选择也很重要。本系统中采用 Microvsiion 公司系列云台镜头解码器。

这系列的云台采用软硬多重防死机原理设计，抗干扰性能好；具有 R485 和 R232 的双接口，抗静电和防雷击；支持所有二、三可变焦镜头的动作；支持各种数字硬盘录像系统和矩形切换系统；除此之外，还可以提供摄像机电源及照明和继电器开关控制。其工作的主要技术参数表，见表 5.2。

表 5.2 云台主要技术参数表

技术参数	说明
工作电源	AC220V@50Hz
输入电压变动率	10%以内
Control 输出电压	AC220V/24V
PAN/TILT 输出电压	AC220V/24V
Lens 电压	DC12V+
PANT/ITL 控制	Up, Down, Left, Right
LENS 控制	Zoom-In/Out, Focus-Near, Far
传送方式	RS-422\232(非同期通信)
传送距离	1.2Km 以内
摄像机 ID 设定	00-99

5.2 系统软件开发平台

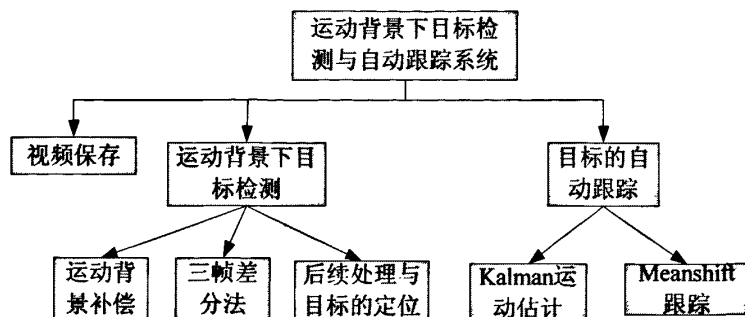


图 5.3 应用系统组成结构

本文的应用系统组成结构如图 5.3，应用系统主要包括：视频保存、目标检测、目标跟踪三个模块。在计算机软件开发过程中，本文考虑到 VC++6.0 能以高效的编码率，快速地开发出图形界面丰富、功能强大的软件系统，所以系统模块中的视频保存处理和整个应用系统的主操作界面在 VC++6.0 完成，如图 5.4。但在数值计算方面 VC++不如 Matlab7.1。特别在图像处理方面，Matlab 以强大的矩阵运算和绘图功能、丰富的图像处理函库、大量稳定的算法库和显示功能，成为图像处理首选仿真软件。因此本文对于目标的检测与跟踪算法处理在 MATLAB 中完成。这样可以节省系统资源、减小程序冗长和缩短程序的开发周期。

在实现 VC++ 调用 Matlab 实现混合编程方面，本文采用调用 Matlab 引擎的方法。

Matlab7.1 的计算引擎采用客户机/服务器的体系结构,它可以通过 Matlab 与 Windows 的 UNIX 的管道或 ActiveX 通道连接,实现对 Matlab 中的 C\C++函数和工具箱中的函数的调用。同时,这种方法可利用 Matlab 强大的图形功能。其基本思想就是: VC 编写的应用程序完成应用系统的前端;通过 Matlab 引擎向后台 Matlab 服务器传送数据;处理后的结果再通过 Matlab 计算引擎返回到应用系统的前端.实现过程如下:

首先，配置 VC 集成环境。配置步骤如下：

Step1、在 VC 中建立 MFC Appwizard (exe)(Dialog Based) 基于单文档的工程。

Step2、配置 VC 下选择菜单: Tools -> option -> Directory ,分别添加文件和库文件的搜索路径: Include file %本机 Matlab 的安装路径 %\extern\include , Library files %本机 Matlab 的安装路径 %\extern\lib\win32\Microsoft\Msvc6 。

Step3、制定引擎函数的库文件,选择菜单:Project -> Settings -> Link , 设置工程: “Object/ library modules” 中添加: libeng. lib libmx. lib 两个库文件。完成设置。

然后,完成 Matlab 引擎法的调用。调用步骤如下:

Step1、定义 Matlab 引擎指针、数组和结构体变量。

Step2、判定 Matlab 的引擎启动与否;如果是构造矩阵进行赋值,否则停止执行。

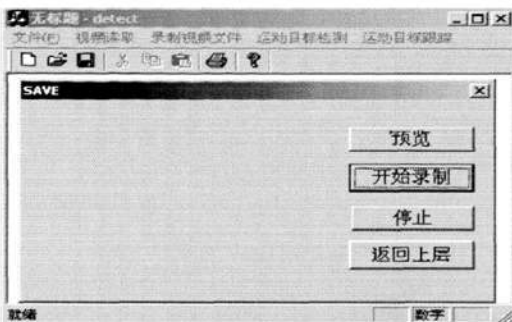
Step3、向 Matlab 工作区间传送构造矩阵,通过调用 Matlab 命令,完成数据处理。

Step4、释放内存,关闭引擎。

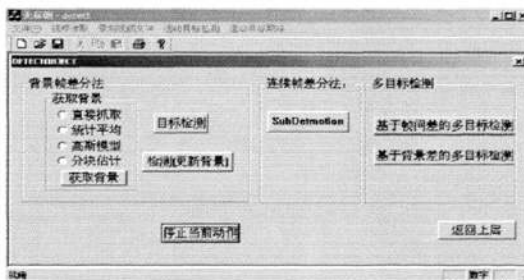
在整个图形处理过程中需要注意一点:VC++6.0 和 Matlab7.1 对图像的存储方式不同。Matlab7.1 对图像数据的存储采用列优先的原则;而 VC++6.0 是采用自下而上,从左到右,行优先的原则的存储方式。因此,在 Step3 中,在利用后台 Matlab 程序对数据完成处理前,先要进行数据的格式转换。



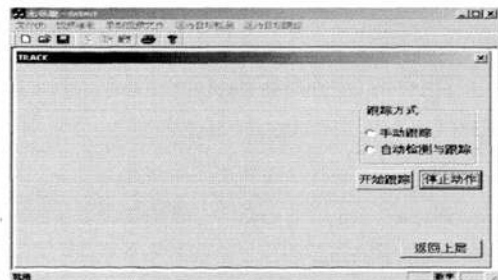
(a) 应用系统的主操作界面图



(b) 视频保存模块的主操作界面



(c) 运动目标检测模块的主操作界面



(d) 运动目标跟踪模块的主操作界面

图 5.4 应用系统与各个模块的主操作界面图

5.3 本章小结

为了验证本文所提及的算法,在本章设计我们设计了一个实验系统平台。在本章的开始先描述了它的硬件组成结构以及各部分硬件的要求和参数,然后根据应用系统平台介绍了选择的开发环境。对于实验结果在各章节中已给出。通过实验证明本文所介绍的算法的可行性。

总结与展望

本文讨论了基于单目视觉运动背景下的运动目标检测与自动跟踪。此系统可以在摄像机转动期间成功检测到运动目标,并且可以根据目标的运动速度主动预测运动目标在下一帧的位置,提前控制云台运动,进而带动摄像机转动,将运动目标锁定在指定的视窗范围,实现了自动跟踪。在研究中提出了一些新的算法,也取得了一定的研究成果。其总结如下:

(1)提出了一种具有鲁棒性的基于特征点匹配全局运动补偿技术。首先介绍完几种摄像机运动的参数模型后,使用 Harris 算法完成特征点提取;然后提出一种改进型的圆形模板匹配法完成特征点的匹配。这种改进的方法不仅提高了时效性,而且可以处理相邻帧间发生的较大幅度平移、缩放、旋转等全局运动。考虑外点的存在对全局运动估计的影响,本文又将鲁棒统计分析和 LMA 优化算法结合的应用到仿射变换模型的参数估计中,成功的去除了外点,进而提高全局运动参量的精度。最后利用双线性内插法完成对当前帧背景匹配,即实现了运动背景静态转化。

(2)基于单目视觉静态背景下帧间差分法的目标检测。本文对比性的研究了几种经典的静态背景下运动目标检测算法,并重点研究了三帧差分算法。

(3)在运动自动目标跟踪部分中,本文提出了一种结合 Kalman 滤波器和 MeanShift 跟踪算法。首先引入 Kalman 滤波器对运动目标的当前运动状态做出估计,且根据运动的速度主动预测物体在下一帧的位置,根据预测值指挥云台运动,进而带动摄像头及时的转动,快速跟踪目标;然后利用 MeanShift 算法完成对目标搜索定位。这样不仅可以解决自动跟踪的滞后问题,而且提高了系统跟踪的实时性和精确性。

再介绍完基于单目视觉运动背景下的运动目标检测与自动跟踪的算法后,本文自主搭建了一个系统实验平台。实验的结果达到预期效果,即证明了本文所提出算法的可行性。

本文虽然对基于单目视觉运动背景下的运动目标检测与自动跟踪做了一定的研究工作,可是由于实验室条件以及时间等因素的限制,在算法优化、理论学习等各方面的研究工作还有待于进一步拓展和深入。具体的内容如下:

(1)对基于全局运动估计的背景匹配方面,虽然本文做了很多改进,但是全局运动估计还是不仅加大了整个系统的运算时间,而且其估计的精度直接决定了后面独立运动目标检测的精度,从而影响到后面的目标跟踪。为了避免计算全局运动,可以考虑通过

一些算法直接分割出运动目标。

(2)对运动目标的检测方面，本文采用三帧差分算法，利用该方法可以较快的、较准确的检测出运动区域。然而，若运动目标运动速度较慢时或沿运动方向的尺度较长，就会出现“目标分裂”现象。这时利用三帧图像可能检测不出运动目标。因此对于运动目标的检测算法选择有待进一步探讨。

(3)对于运动目标自动跟踪方面，当摄像机与运动物体同时运动时，在图像采集与处理过程中有时需要暂停摄像机的转动，降低整个系统的实时性，因此在改进的方法中应该考虑到使用摄像机的标定和图像的定位以及运动补偿等技术来改善这方面的缺陷。另外，目标运动估计算法的时效性较差，还有待研究。

(4)对于远距离的图像传输，我们可以考虑使用无线传输方法，进而减少布线的人力和物力。

参考文献

- [1] 贾云得. 机器视觉[M]. 北京:科学出版社, 2002,4
- [2] Yilmaz. Object Tracking and Activity Recognition in Video Acquired Using Mobile Cameras[D].University of Central Florida, 2004
- [3] 丁家锐,安锦文. 基于运动平台的多目标检测与跟踪[D]. 西北工业大学, 2007
- [4] Meier T, Ngun K N. Video Segmentation for Content-Based Coding[J]. IEEE Trans on Circuits and Systems for Video Technology, 1999,9(8):1190-1203
- [5] Barron J, Fleett D. Performance of Optical Flow Techniques[J]. International Journal of Computer Vision, 1994,12(1):42-47.
- [6] Gupt S, Masound O. Detection and Classification for Vechicles[J]. IEEE Trans on Intelligent Transportation Systems, 2002,3(1):37-47
- [7] 邱德润,朱明早,伍宗富. 一种基于帧间差分与背景差分的运动目标检测新方法[J].湖南文理学院学报(自然科学版), 2007,19(1):80-83
- [8] 张珍明,朱光喜,王锋. 一种新的全局运动估计算法[J]. 计算机工程与用, 2005,26:5-7
- [9] Nam-Joon Kim, Hyuk-Jae Lee. Probabilistic Global Motion Estimation Based on Laplacian Two-Bit Plane Matching for Fast Digital Image Stabilization[J]. EURASIP Journal on Advances in Signal Processing, 2008, 1155:1-10
- [10] Yosi Keller, Amir Averbuch. Fast Gradient Methods Based on Global Motion Estimation for Video Compression[J]. IEEE Trans on Circuits Syst Video Technol, 2003,13(4):300-309
- [11] MJD Powell. Radial Basis Function for Multivariable Ineterpolation[J]. Algorithms for Approximation, Oxford, 1987:143-167
- [12] 王素玉,沈兰荪. 智能视觉监控技术研究进展[J]. 中国图象图形学报, 2007,12(9): 1505-1514.
- [13] K. Mehmood, M. Mrak, J. Calic. Object Tracking in Surveillance Videos Using Compressed Domain Features from Scalable Bit-streams Signal Processing[J]. Image Communication, 2009,24(10):814-824
- [14] Mohand Saïd, Allili, Djemel Ziou. Object Tracking in Videos Using Adaptive Mixture Models and Active Contours[J]. Advances in Blind Signal Processing, 2008,71(10): 2001-2011
- [15] R Collins, A Lipton et al. A system for Video Surveillance and Monitoring: VASAM Final

Report[D]. Robotic Institute Carnegie Mellon University, 2000

[16] R.Collins,A.Lipton,T.Kanade. Introduction to The Special Section on Video Surveillance[J]. IEEE trans.Pattern Analysis and Machine Intelligence, 2000. 22(8):745-746

[17] Sorwar G, Murshed M, Dooley L. Fast Global Motion Estimation Using Iterative Least-square Technique. Proceedings of the 4th International Conference on Information[A]. Communications & Signal,Proceedings of the 4th IEEE Pacific-Rim Conference On Multimedia Singapore[C], 2003,1: 282- 286

[18] He Y.W, Feng B,Yang S. Fast Global Motion Estimation for Global Motion Compensation Coding[A]. Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)[C]. 2001,2: 233-236

[19] Cheung H.K, Siu W.C. Fast Global Motion Estimation for Sprite Generation[A]. Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)[C], 2002,3,26-29

[20] 郑江滨. 视频监控方法研究[D]. 西安: 西北工业大学, 2002

[21] 陈颖,李在铭. 基于鲁棒回归的运动背景补偿技术[J]. 信号处理. 2002,18(1):37-38

[22] 郭玉辉,高世伟. 一种基于 LMA 的电子稳像技术[J]. 计算机应用研究,2008,25(6): 1768- 1770

[23] Zhu Juanjuan,Guo Baolong. Electronic Image Stabilization System Based on Global Feature Tracking[J]. Journal of Systems Engineering and Electronics, 2008,19(2):228-233

[24] Hussein Alzoubi, W. David Pan . Fast and Accurate Global Motion Estimation Algorithm Using Pixel Subsampling[J]. Information Sciences,2008,178: 3415-3425

[25] Hung Changchang, Shang Honglai, Kuang Ronglu. A Robust Real-time Video Stabilization Algorithm[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2006,17: 659-673

[26] Shoichi Araki S, Matsuoka T. Haruo Takemura.Real-Time Tracking of Multiple Moving Objects in Moving Camera Image Sequences Using Robust Statistics[A]. Pattern Recognition, 1998. Proceedings. 14th International Conference on[C], 1998, (2):1433-1435

[27] Smiths M, Brady J.M. SUSAN-A New Approach to Low Level Image Processing[J]. International Journal of Computer Vision,1997, 23(1):45-78

[28] Harris C, Stephens M. A Combined Corner and Edge Detection[J]. Image Vision Computing, 1998, 6:121-127

- [29] 张俊峰,邢幸. 基于 Harris 角点的静止图像配准研究[J]. 科技资讯,2008,10:9-10.
- [30] 米长伟,刘晓利,徐明友. 基于 S 的改进算法[J]. 火箭与制导学报,2004,24(1):85-87.
- [31] Ying-Lai Xi,Chong-Yang Hao,Yang-Yu Fan.A Fast Block-Matching Algorithm Based on Adaptive Search Area and Its VLSI Architecture for H.264/AVC[J]. Image Communication, 2006,21(8): 626-646
- [32] Yeon-Ho Kim,Avinash C.Kak. Error Analysis of Robust Optical Flow Estimation by Least Median of Squares Methods for the Varying Illumination Model[J]. IEEE Trans on Pattern Analysis and Machine Intelligence,2006: 1418-1435.
- [33] Cheung H K, Siu W C. Fast Global Motion Estimation for Sprite Generation [J]. IEEE Inter Symposium on Circuits and Systems, 2002: III-1 -III-4.
- [34] 崔宇魏,潘建寿. 运动目标检测与跟踪中有关问题的研究[D]. 西安:西北大学,2005
- [35] Barron J, Fleet D, Beauchemin S.. Performance of Optical Flow Techniques[J]. International Journal of Computer Vision, 1994, 12(1):42-77
- [36] I.Haritaoglu,Davis,D.Harwood. W4:Who? When? Where? What? -- A Real Time System for Detecting and Tracking People[A]. International Conference on Automatic Face and Gesture Recognition [C],1998,235-278
- [37] 冯伟,赵荣椿. 视频序列运动目标检测与识别方法研究[D]. 西安:西北工业大学,2003
- [38] Kim J.B, Park H.S, Park M.H. A Real-Time Region-Based Motion Segmentation Using Adaptive Thresholding and K-means Clustering[A]. Proceedings of the 14th Australian Joint Conference on Artificial Intelligence[C], 2001, 213-224.
- [39] 章毓晋. 图像分割[M]. 北京: 科学出版社, 2001, 2
- [40] 周志宇,汪亚明,黄文清. 基于熵的运动目标检测[J]. 微计算机信息,2003,19(5): 35-36.
- [41] 何明一,卫保国编著. 数字图像处理[M]. 北京:科学出版社, 2008:229-230
- [42] 刘雪,常发亮教授. 基于图像序列的运动目标检测与跟踪算法研究[D]. 济南:山东大学,2007
- [43] 隋晔,马钱. 交通监控系统中运动目标分类和跟踪研究[J]. 信息与控制, 2003,8(24): 156
- [44] Avid A.Foursyth,Jean Ponce. 计算机视觉——一种现代的方法[M]. 北京:清华大学出

版社, 2004

- [45] Jae-Chern Yoo, Byoung Deog Choi. 1-D Fast Normalized Cross-Correlation Using Additions[J]. Digital Signal Processing, 2010:440-456.
- [46] Pieter Kleve, Richard De Mulder. Privacy Protection and the Right to Information [J]. Computer Law & Security Report, 2008, 24(3): 223-232.
- [47] Gang Tian, Rui-Min Hu, Zhong-Yuan Wang. Object Tracking Algorithm Based on MeanShift Algorithm Combining with Motion Vector Analysis[A]. Proceedings of the 2009 First International Workshop on Education Technology and Computer Science[C], 2009: 987-990
- [48] 王琳, 潘建寿教授. 视频运动目标跟踪中有关问题的研究[D]. 西安: 西北大学, 2006
- [49] 杜立东. 计算机图形图像处理[M]. 北京: 中国财政经济出版社, 2005
- [50] Jin Liu, Tianxu Zhang. Recognition of the Blurred Image by Complex Moment Invariants[J]. Pattern Recognition Letters, 2005, 8(26): 1128 - 1138
- [51] 韩思奇, 王蕾. 图像分割的阈值法综述[J]. 系统工程与电子技术, 2002, 6(24): 91-102
- [52] Fukunaga K, Hostetler L D. The Estimation of the Gradient of the Density Function with Applications in Pattern Recognition[J]. IEEE Trans on Information Theory, 1975, 21(1): 32-40
- [53] Cheng, Y. Mean-Shift, Mode Seeking, and Clustering[J]. IEEE Trans. Pattern Analysis and Machine Intelligence, 1995, 17: 790-799,.
- [54] Milan Sonka, Vclav Hlavac, Roger Boyle. 图像处理、分析与机器视觉[M]. 北京: 人民邮电出版社 (英文版), 2002.1.
- [55] M. Isard, A. Blake, Condensation-Conditional Density Propagation for Visual Tracking[J]. International Journal of Computer Vision, 1998, 29(1): 5-28,.
- [56] Comaniciu, D., V Ramesh, P. Meer. Kernel-Based Object Tracking[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 25(2): 564-577
- [57] 刘继艳, 潘建寿, 吴亚鹏. 结合 Kalman 滤波器的 Mean-Shift 跟踪算法[J]. 计算机工程与应用, 2009, 45(12): 184-186
- [58] Z. W. Zhu, Q. Ji, Robust Real-Time Eye Detection and Tracking Under Variable Lighting Conditions and Various Face Orientations[J], Computer Vision and Image Understanding. 2005, 98: 124-154
- [59] 常好丽, 史忠科. 基于单目视觉的运动行人检测与跟踪方法[J]. 交通运输工程学报. 2006, 6(2): 55-59

- [60] 王彪,王成儒,王芬芬. 固定场景下多目标运动检测与跟踪[J]. 计算机工程与设计, 2008,29(8):2014-2019
- [61] 刘海涛,谭克俊. 基于 ARM9 的视频监控与自动跟踪控制系统研究[D]. 大连:大连海事大学,2008
- [62] Shiuh-Ku Weng, Chung-Ming Kuo. Video Object Tracking Using Adaptive Kalman Filter[J]. Journal of Visual Communication and Image Representation, 2006.6(17):1190-1208

攻读硕士学位期间取得的科研成果

贺琴,潘建寿,刘继艳. 摄像机运动下基于背景匹配的运动目标检测. 计算机工程与应用

致谢

时光匆匆，研究生学习的生活即将要结束了。在论文即将完成之际，谨向我的导师潘建寿教授表示深深的感谢。三年来，我的每一步成长都凝结着潘老师的心血，导师以他高尚的品德、严谨的科学态度、孜孜不倦的工作作风培养了我良好的科研素养和踏实的学习心态；他渊博的知识、丰富的工程经验和科学的指导方法使我从论文的选题到撰写有一个系统完整的学习；他谦虚朴实、平易近人的态度是我生活中的榜样。

感谢同教研室的师兄吴亚鹏、马永刚，师姐刘继艳和张娟在学习中的指导；感谢马鹏、弥丽丽在学术讨论中对我的思维的激发和对于程序的完善性与合理性上所提出的宝贵意见；感谢师妹周静、曹玲，师弟华喜彬在论文修改中提出的宝贵意见。在此感谢你们。

衷心感谢我的父母、哥哥和嫂嫂。在我漫长的求学生涯中，总伴随着他们的理解、无私的关怀和鼓励。他们是我永远向前的精神动力。

感谢李飞、袁晓雷等好朋友在平时生活中对我的鼓励和帮助，是他们让我渡过了快乐和难忘的大学时光。

感谢曾经的和现在的室友，大家一起维系着姐妹般的感情，创造了一个温暖的休息环境。

最后，感谢所有关心和帮助过我的人。