

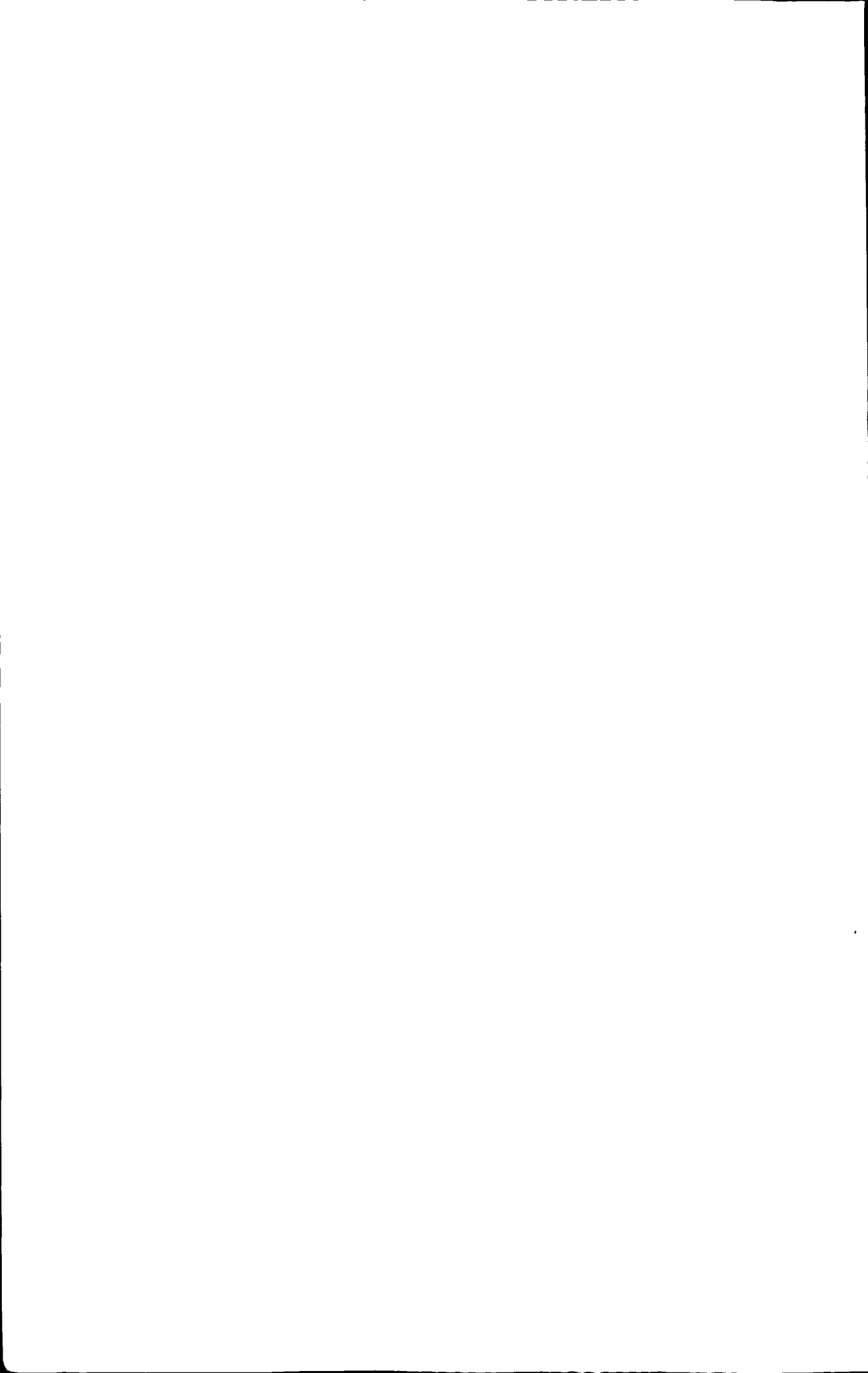


## 摘 要

航拍视频目标跟踪是近期目标跟踪领域研究的热点。从无人机到图像制导导弹,从航拍灾情分析到空基交通平台监测,无论是在军用和民用领域都得到了广泛的应用。航拍目标跟踪的最大特点是载具运动给跟踪算法带来的干扰,如何有效保持跟踪算法的稳定,是我们需要解决的问题。本文通过分析航拍视频的特点和难点,提出了一种航拍视频目标跟踪解决方案。

本文首先对检测和跟踪技术进行分类梳理,分析检测和跟踪的分类及其关系,介绍了基于贝叶斯的目标跟踪算法。针对航拍中载具运动问题,采用基于 SIFT 特征点的图像配准进行载具运动补偿,提出了基于 SIFT 补偿和粒子滤波的航拍视频目标跟踪方法。此方法是在 SIFT 点配准后利用对帧间目标和摄像头短暂匀速的假设,用帧差法检测估计中间帧目标位置送检给粒子滤波,同时对粒子的投放和扩散进行纠正的航拍跟踪算法。该算法充分利用了帧间的运动信息不仅对帧差进行了扩展同时也很好的和粒子滤波相结合,对基于普通高斯欧式距离粒子加权和巴氏颜色距离加权下的的粒子滤波进行了分析比较,实验结果表明本算法对目标形状不敏感同时定位精准,鲁棒性较强。

**关键词:** 航拍目标跟踪 SIFT 特征点 图像配准 中间帧估计

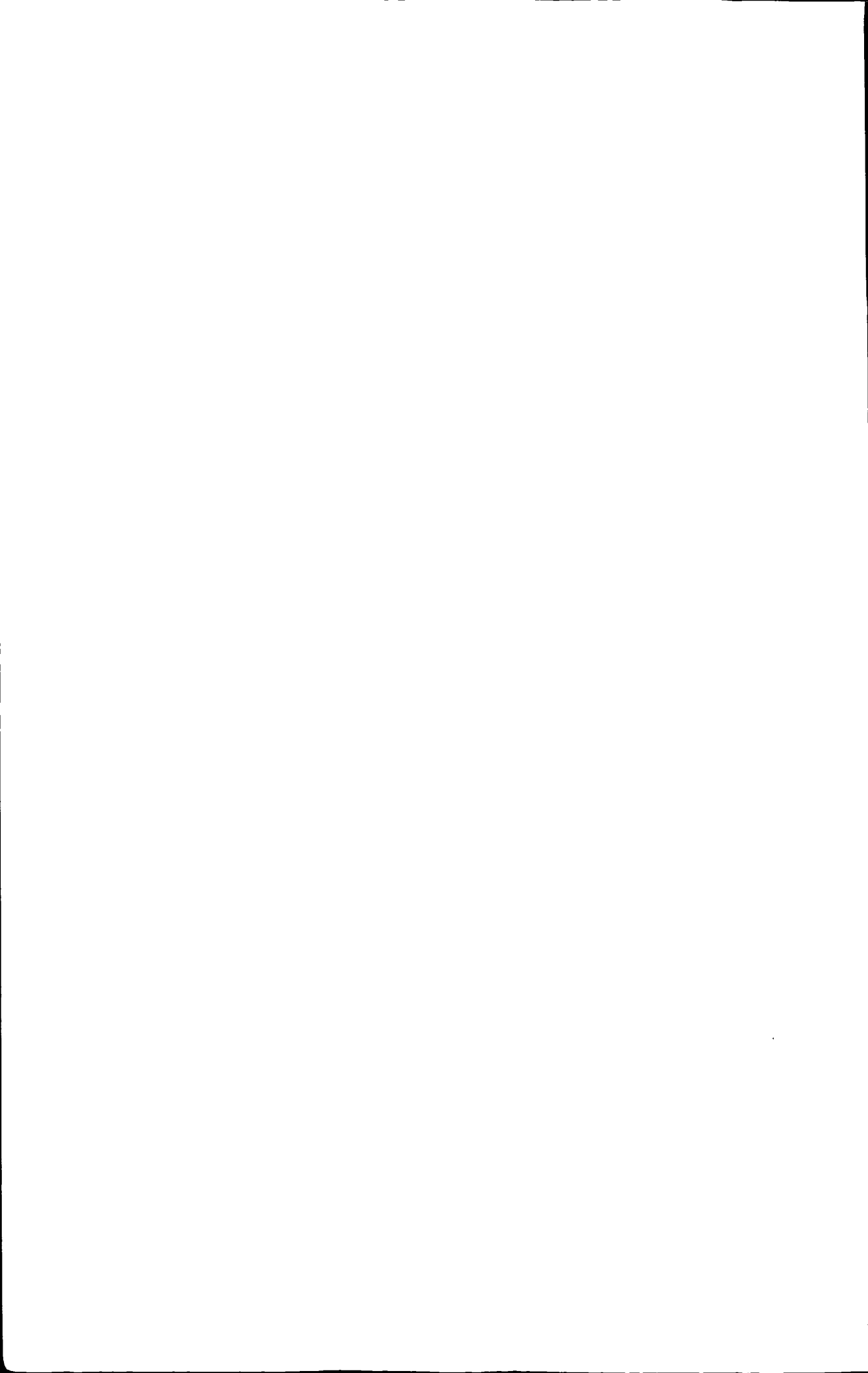


## Abstract

Aerial video target tracking is a research hotspot of target tracking field. It has been widely used in many military and civilian areas, such as UVA, image guided missiles, aerial disaster analysis, transportation monitoring based on air platform, and so on. The difficulty of aerial target tracking is that vehicle movement brings about interference to tracking. With the analysis of the characteristics and the difficulties of aerial video, the paper puts forward a kind of aerial video target tracking solutions.

This paper aims at the target tracking algorithm based on Bayesian in aerial vehicle motion problems, and adopts SIFT feature points image registration for vehicle motion compensation. A solution based on particle filter and SIFT compensation for aerial video target tracking is presented. With the assumption that both object and camera move at the same rate between the frames in the short term after SIFT matching algorithm, the middle frame's object position is sent to particle filter, and location and diffusion of the particles are modified. The presented algorithm makes full use of the time information between the frames to modify the frame subtraction. Comparing the presented algorithm with the particle filter with Euclidean distance weighted and the particle filter with Bhattacharyya distance weighted respectively, the experiment results show that it is not sensitive to target shape, and has better positioning accuracy and robustness.

**Keywords:** Tracking in the aerial video   Sift key point   Image registration  
Estimation of the middle frame



## 目 录

第一章 绪论.....	1
1.1 引 言.....	1
1.2 本课题研究背景及意义.....	1
1.3 本论文的主要工作和内容安排.....	2
第二章 目标跟踪算法概述.....	5
2.1 目标跟踪的目的和意义.....	5
2.1.1 目标跟踪的目的.....	5
2.1.2 目标检测和跟踪的关系.....	6
2.2 目标在机器视觉中的表示.....	7
2.2.1 目标外形表示.....	7
2.2.2 目标跟踪特征表示.....	10
2.3 目标检测跟踪分类.....	11
2.3.1 目标检测.....	11
2.3.2 目标跟踪.....	13
第三章 基于贝叶斯的跟踪算法.....	15
3.1 引 言.....	15
3.2 卡尔曼算法.....	15
3.2.1 卡尔曼在跟踪中的应用.....	16
3.2.2 卡尔曼滤波的条件和算法步骤.....	17
3.3 粒子滤波算法.....	19
3.3.1 蒙特卡洛方法.....	19
3.3.2 序贯重要性采样算法.....	21
3.3.3 重采样.....	22
第四章 基于 SIFT 特征点的运动补偿方案.....	25
4.1 引 言.....	25
4.2 抗载具运动方法.....	25
4.2.1 跟踪算法抗载具运动.....	25
4.2.2 运动补偿算法.....	27
4.3 图像配准方法.....	28
4.3.1 SIFT 点介绍.....	28
4.3.2 SIFT 点生成步骤.....	29
4.4 仿射变换.....	34

---

第五章 基于 SIFT 补偿和粒子滤波的航拍目标跟踪 .....	35
5.1 引 言 .....	35
5.2 SIFT 配准下基于高斯欧式距离加权的粒子滤波 .....	35
5.3 SIFT 配准下基于巴氏距离加权的粒子滤波 .....	37
5.4 基于中间帧估计的粒子滤波 .....	40
第六章 总结与展望 .....	45
致 谢 .....	47
参考文献 .....	49

## 第一章 绪论

### 1.1 引言

航拍目标跟踪在军事上具有极其重要的应用,以信息化为主导的未来战争对战场的监视系统提出了更高的要求。其中对目标的检测、跟踪和识别更是研究的热点,这些技术为对敌实施精确打击、保护本方有生力量、超视距对抗奠定了坚实的基础。其中关键技术之一就是载具运动条件下的目标跟踪,尤其是航空拍摄条件下对目标进行准确定位和跟踪。

航拍是指在飞机、飞艇等航空载具在空中飞行过程中对背景或者目标进行主动式、多角度、多方位追踪摄影摄像。他和传统的固定摄像头拍摄方式有着很大的不同,对目标可以采取主动式的跟踪方式,由于载具的运动扩大了其跟踪拍摄的范围,这在侦查上具有很大的优势。随着近年来无人机的服役和基于图像制导导弹技术的日趋完善,我们对航拍目标的跟踪检测识别技术的研究投入越来越高,这对我国国防事业的发展具有重要的意义。

无人机是航拍目标跟踪领域的一个重要应用<sup>[2]</sup>,他是一种以无线电或由自身程序控制为主的不载人飞机。它的研制成功和战场运用,带来了真正意义上的现代化非接触式战争。甚至一些专家预言:未来的空战将是具有隐身特性的无人驾驶的飞行器和防空武器之间的作战。无人机的诞生可以追溯到1914年,在第一次世界大战期间英国军事航空学会提出的一项AT计划,研制一种无线电控制的小型无人飞机用于对敌战区投掷炸弹。随着现代计算机技术和无线通信技术的发展,无人机已由小型化简单化向着大型化集成化现代化发展。我们对其赋予了更多的任务,如跟踪监测、直接攻击等。这就要求其具有更加优良的软体支持。其中对于目标跟踪算法是一个主要的研究方向,因为无人机的目标侦查、精确打击等重要任务的开展离不开改算法的支持。

本文主要研究的是航拍视频中运动目标的跟踪算法。当无人机打击确认打击目标的时候,需要通过摄像头传感过来的图像信息进行目标跟踪定位。甚至某些精确打击武器如图像制导导弹,发射出去后还需要对目标进行锁定和指引。由于涉及军事技术,该研究课题的信息开放程度一度不高,国外一些研究机构对敏感技术的细节往往不够详细。

### 1.2 本课题研究背景及意义

目标跟踪技术已经应用到了很多领域,无论军用民用都有很强的实用价值。本文研究的航拍目标跟踪近些年是一个研究热点,其首先应用的领域是军用:如无人机遥感,卫星定位跟踪等先进应用上。但随着航空工业和技术的发展,航拍门



槛的降低和新的信息采集系统的应用,我们在很多民用领域也能发其踪影,如基于空基的交通检测平台,重大突发性事件的实时监控或者地质地理灾害的探测等,甚至在摄影领域都有航拍的一席之地。

我们空中所拍摄的图片中一定是有我们所感兴趣的区域的,如果是一个动态的目标的话,我们将如何去对图片或者视频进行处理呢。当信息量增大的时候需要机器去做出一定的处理和预判,这在无人机和卫星定位跟踪上有着强烈的需求,比如空中检测的时候有的目标较小不容易通过肉眼去辨识,这就需要通过一定的程序去处理图片,获得精确的目标信息才可以进行下一步的打击或者定位。由于载具的运动我们处理这些视频图片上的目标时候和以往是有所不同的,当处理摄像头不动的时候的被动式目标跟踪时,我们有着比较成熟的技术和解决方案,而主动式的对目标进行跟踪我们将会面对的问题主要还是载具运动给以往解决方案带来的一系列困扰所造成的系统不稳定,如不能采取措施会造成很严重的实际后果,如无人机打击错误目标, GPS 定位失准等。错误的信息不仅在战场上可能造成人员伤亡,同时在现实生活中也会产生重大的人员财产损失。所以这就需要我们去研究此类问题,解决此类问题。

这里我们是在基于卡内基梅隆大学的开源数据做的实验,主要是针对的目标是车辆,这在航拍中是很常见的目标,也是具有典型研究意义的物体。当无人机或者卫星对动目标进行跟踪监测的时候,自身的姿态和运动都会影响跟踪算法的正常运行,虽然有的跟踪算法如粒子滤波本身就具有抗载具运动的能力,但是当出现载具剧烈运动的时候,目标就会跟踪不上,导致后续无法继续处理。

### 1.3 本论文的主要工作和内容安排

针对上述问题,本文研究了航拍情况下的目标跟踪方法,全文的主要工作以及内容安排如下:

第一章绪论,从航拍跟踪的目的和应用入手介绍其特点和用途,其中着重介绍了无人机在军事上的应用。应我国国防事业和未来广阔的民用市场的需要,对本课题的研究进行了必要性分析。

第二章跟踪算法综述,从目标检测和跟踪的区别和联系入手介绍了目标跟踪的本质和目的,同时对目标跟踪中所遇见的背景模型建立,跟踪特征选择,目标表征以及算法区分做了详细的介绍。

第三章介绍了基于贝叶斯的跟踪算法——卡尔曼滤波和粒子滤波,并对其原理、推导、优缺点和算法步骤进行了概述。

第四章针对航拍视频载具运动问题,研究运动补偿方法,采用了基于 SIFT 特征点补偿背景运动的方法对跟踪算法进行修正,使跟踪系统更加稳定和可靠。

第五章首先对前几章所提出的解决方案进行了实验，采用的两种实验算法分别为 SIFT 配准下基于一般高斯欧式距离粒子加权的粒子滤波和 SIFT 配准下基于巴氏距离加权的粒子滤波，实验数据显示出两者各有优缺点，前者耗时少但是定位精度差容易受到噪声干扰，而后者定位精确但是耗时严重效率低下。提出一种基于 SIFT 补偿和粒子滤波的航拍目标跟踪方法，该方法在 SIFT 点配准后利用对帧间目标和摄像头短暂匀速的假设，用帧差法检测估计中间帧目标位置送检给粒子滤波，同时对粒子的投放和扩散进行纠正。

第六章对全文工作进行了总结和展望。

最后是致谢和参考文献。



## 第二章 目标跟踪算法概述

### 2.1 目标跟踪的目的和意义

#### 2.1.1 目标跟踪的目的

目标跟踪<sup>[3]</sup>的目的主要有：(1) 基于运动的识别，即基于步态和自动目标检测等的个体识别；(2) 自动监视，即监视现场检测可疑活动和事件；(3) 视频索引，即自动标注和检索的多媒体影片的数据库；(4) 人机交互，即手势识别，眼睛凝视追踪数据输入到计算机等；(5) 流量的监测，即实时统计数据收集的交通指挥交通；(6) 车辆导航，即基于视频的路径规划和避障功能。如图 2.1 所示

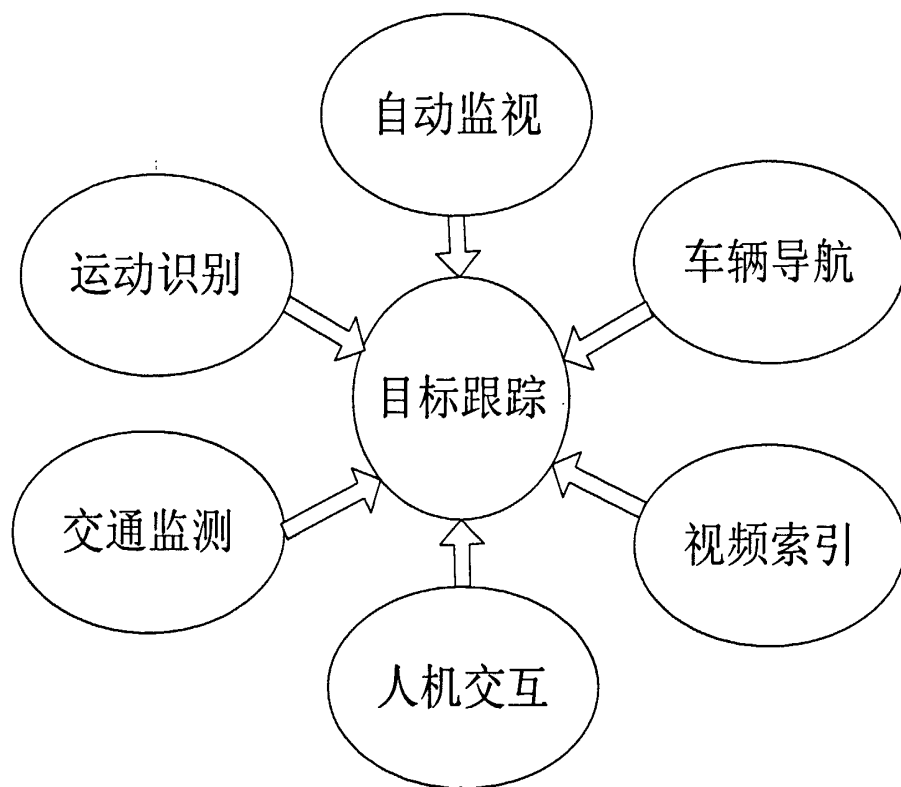


图 2.1 目标跟踪目的和应用

目标跟踪是计算机视觉领域的一个重要任务，而航拍目标跟踪是目标跟踪的一个重要方面，最先是在军事方面得到了关注，如无人机自动监视、战区指挥导航、目标定位等。近些年来随着高性能计算机和视频器材的成本降低，民用方面的需求也随之增长，几乎涵盖了以上所有的应用。其中最重要的是对安保系统需求的增加，国内外许多大城市在公共场所加装了监控装置，我们可以通过跟踪算法提取了运动目标的信息后可以做出各种判断和预读，这大大减少了人力工作量，

这在当前信息化社会中具有重要意义。

### 2.1.2 目标检测和跟踪的关系

大多数文献指出跟踪是将帧间目标关联起来从而产生目标运动的轨迹坐标。这样的解释对于单目标存在一定的歧义，当单目标检测出来的坐标点已经可以构成一组轨迹坐标，我们是否认能将其认为是一种跟踪的行为。在具体应用中有时候目标检测出来的轨迹定位准确性比跟踪后的效果要好。

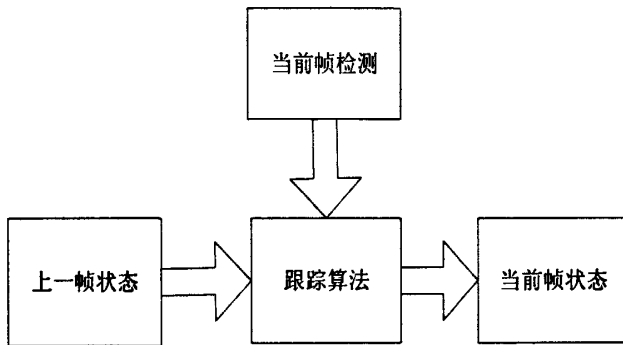


图 2.2 目标跟踪机制

如图 (2.2) 所示，跟踪算法都是在帧间对目标进行处理，处理的方式大多是利用前一帧目标的状态和当前帧目标的检测结果来对当前帧的目标状态进行估计匹配。这种机制首先让人产生的问题是检测结果和状态的关系，当然这种检测不一定是目标直接的位置速度信息也有可能是图片中的特征信息，但是一般跟踪步骤是先对当前帧检测然后利用上一帧的状态信息来估计，如果直接对当前帧检测出目标是否更加的方便。要解答这个问题首先我们需要了解两个问题：

1、目标状态和检测结果以及真实信号之间的关系，在信号处理上，其实检测或者说观测的结果更加接近于真实信号，而状态信号可以认为是我们为了更好的处理计算信号而人为认为的理想形式，我们最终需要得到的也是这种状态信号。

2、检测和跟踪的关系，检测大多是针对于静态的单帧图像来做的，目的就是要区别出单帧图片上目标和背景，而跟踪是要填补帧间目标运动的空白从而找寻帧间目标间的一致性将其关联起来。

好的跟踪是要在复杂场景条件下能跟的上，相对来说跟的准只是一个附带的参数。但反过来上一帧的跟踪定位的准确程度又会对下一帧的跟踪算法产生影响。准确性好的检测定位能为跟踪系统跟的上目标提供有力的保障。我们知道物体的运动是连续的，在数学上只能用微积分来表示。而摄像头中物体的运动被割裂成单帧的图片了。跟踪算法要做的就是将这种所谓的割裂了的运动目标串起来。同时这里目标对象的外观模型、运动模型、场景的复杂程度等因素都对这种跟踪算法提出了不同的要求，从而产生出不同种类的跟踪算法。

如何在帧间将目标关联起来，早期的点跟踪研究是将一系列的限制条件加在目标运动上，比如帧间运动速度小，目标位置帧间变换不大等条件。这些一对一的强制关联确实对某些场景很管用，但是其局限性也是显而易见的，尤其是缺乏必要的数学理论支撑。目前大多数都是用目标出现的概率密度来得到的，统计方法和概率密度的引进使其在数学上得到了有力的支撑。由于目标的出现是以帧为单位，这迎合了许多算法迭代的思想，有助于算法的输入更新和迭代。如卡尔曼和粒子滤波都是遵循贝叶斯方程推导而来，前者是在噪声是高斯，状态运动方程是线性的条件的贝叶斯最优解，而后者是应用了蒙特卡洛原理，直接对目标出现的后验概率密度用粒子加权的方法进行估计。均值漂移算法是通过非参数化的密度概率模型加上必要的核窗函数找寻区域目标出现概率密度梯度的最大值从而得到目标下一帧的出现位置。

大多数的跟踪算法都是由当前帧目标的状态和下一帧的观测信息来对下一帧进行一个状态估计和修正，从而得到下一帧的目标状态信息。这种模式也是考虑到了摄像头是帧帧图片的输入和更新来进行跟踪的一种方式，目的就是要用当前所有已知信息来推出下一帧目标的状态，当然中间也要加入对下一帧检测，这种机制下对目标的一致性的找寻就完成了跟踪算法，由于计算机算法迭代更新技术可以循环进行下去。

## 2.2 目标在机器视觉中的表示

当一个跟踪问题被提出的时候，首先需要考虑的是所跟踪的目标要如何来表示，是用他的外观还是形状还是其他？好的表示有利于跟踪的开展，但这并不是以人为因素为转移的。接下来是跟踪的环境是怎样的一个场景，是否会有光照的强烈变化，目标运动强度如何是否会有遮挡等因素的影响？最后要做的是目标的运动是怎样的一种运动，如何建立一个普遍适用的模型将其纳入其中。以上任何一个问题的考虑都可能产生不同的跟踪研究道路。我们所需要考虑的是如何在一定能力里选择最适合其应用的跟踪算法。

### 2.2.1 目标外形表示

目标的表示方法是跟踪一个很重要的起点，这会对后续算法的产生许多影响，在后面介绍到的跟踪分类里主要还是从这个源头分开的。目标对象是复杂的，主要有以下几个原因：

- 1、具有复杂形状的目标；
- 2、非刚性目标的目标模型建立；
- 3、目标作复杂的运动；

- 4、图像中的噪声干扰;
- 5、完全和部分遮挡的出现;
- 6、三维图像信息投影在二维的图像上所造成的损耗;
- 7、光线强度的变化;
- 8、实时处理要求。

我们一般用目标的形状或者外观来对其表示,当然他们相结合的情况来表示目标也是一个不错的选择。首先我们从物体的形状来划分下,在镜头里我们所看到的物体目标的形状无非可以分为三个主要的情况:

### 1、点

这里所说的点不是一个像素点,而是指图像里一个很小的区域可以由多个像素点来表示,其内涵是指像素值很稳定的一个近圆小区域。当镜头被拉远的情况下这样的目标是很常见的,因为检测工作相对容易,可以每帧都做,对于跟踪算法来说这是一个不错的目标表示情况。如图 2.3 所示:高空中的大雁,远处的船舶。

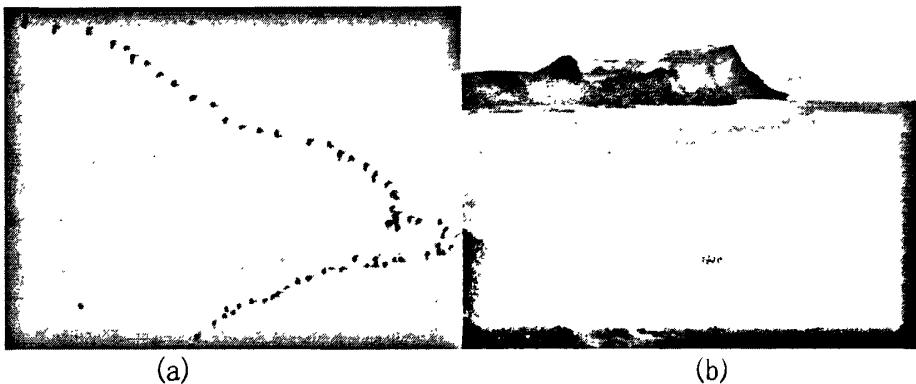


图 2.3(a)远镜头下的大雁;(b)远镜头下的船

### 2、刚性物体

虽然上述的点也是刚性的一种,但是这里在镜头里更多是指一个具有明显和较稳定外观形状特征的像素区域。这类一般都可以用几何形状来表示,如矩形、椭圆等。由于其形状外观比较稳定所以上述的表示基本可以有效的代表这个所要跟踪的物体。如果是基于模板的跟踪,一般在起始帧的时候检测出该目标区域,然后以此为基础进行跟踪。在这种表示法下目标的运动模型可以通过仿射、转移等变换来完成。在没有特殊的情况下有比较好的效果。如图 2.4 所示近镜头下的汽车,飞机等。

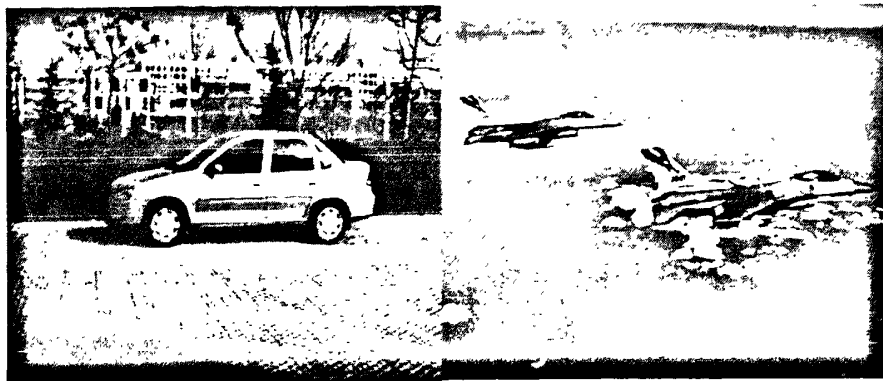


图 2.4 近镜头下的刚性物体

### 3、非刚性物体

这类物体大部分是由几个刚性部件组成的。这种物体的表示方法有时候可以和刚性物体一致，这也主要是由于镜头的拉伸和具体应用的不同所造成的。大部分情况下用其轮廓或者骨架结构等结构来表示。如图 2.5 (b) 近镜头下的人体。



(a) 镜头拉近下的大雁

(b) 近镜头下的人体

图 2.5 作为非刚性处理的目标

需要注意的是上述的分类也不是绝对的，如图 2.3 (a) 和 2.5 (a) 的不同镜头焦距的大雁，所作出的目标表示不一样的。很容易理解的是在场景中有的时候他们之间是存在转换的情况，这就要求算法根据实际可能出现的情况做出具体的调整。

目标的外观在这里可以直观的理解为他是由表示其外观的一些特征，如颜



色、纹理、边缘、特殊点等来组成的这样一个你所观察到的集合，这些特征在图像感知和构建目标中有着很好的应用。光有目标的形状的情况下对目标进行跟踪是不稳定和可靠的，尤其是在环境复杂的情况下，还需要加入一些目标本身更加稳定可靠的特征来综合跟踪。大多数的算法是将物体的形状和外观表示相结合的。一个例子是Meanshift算法中对目标形状区域进行颜色特征的直方图建模。这是一种对目标外观的无参数化建模，建立的是目标外观的概率密度。这里的参数是指某一概率密度分布的参数，如高斯的均值和方差，二项分布01的概率等。需要注意的是这里是概率密度不是概率，是对目标外观如颜色等特征建立的模型，概率密度是在不断改变的，所以需要不断被估计。为什么将目标的外观表示为一个概率密度呢，这是从数学理论出发的一個自上而下的策略，其实可以理解为当前帧目标出现的概率密度分布，那他在下一帧出现的概率密度是由他在这一帧出现的概率密度演变而来的，在后面讲到的卡尔曼和粒子滤波中都会出现这个概念，均值漂移算法在数学上也是延续了这条路线，不同的是他需要考虑的目标由漂亮的点变成了一块带有明显特征的区域，这就需要考虑更多的东西如区域核心的把握即聚类、特征模型的建立等等。最早出现的外观概率密度估计的建模方式是带参数的，如高斯分布或者混合高斯分布。这样的分布明显不能满足现实中复杂概率分布出现的情况，无参数估计的出现是适合发展规律的。

总而言之，跟踪算法和目标表示方式的关系是很紧密的。表示方式往往要根据具体应用来选择，而跟踪算法的输入和更新迭代也要收到目标建模的影响。

### 2.2.2 目标跟踪特征表示

在上面已经有了特征的介绍，但是如何选择合适的特征对跟踪起着关键的作用。一般来说，一个视觉系统最的独特性是我们最期待的，因为目标可以在特征空间中很容易的被辨识出来。特征选择与目标表示密切相关。例如，上一节介绍的柱状图目标表示法（颜色直方图）中选取颜色作为目标的特征就是一个很经典的应用。而对于轮廓的表示法，对象边缘通常作为其主要特征。在一般情况下，许多跟踪算法使用这些特征的结合。常见的视觉特征如下：

1、颜色：影响物体表面颜色主要有两个物理因素：首先是光源的光谱分布，其次就是物体表面反射特性。在图像处理领域，RGB 颜色空间是被我们经常使用的，所谓 RGB 就是：红（Red）、绿（Green）、蓝（Blue）三种颜色光源。RGB 色彩模型的混色属于加法混色。每种原色的数值越高，色彩越明亮。R、G、B 都为 0 时是黑色，都为 255 时是白色。RGB 虽然表示直接，但是 R、G、B 数值和色彩的三属性没有直接的联系，不能揭示色彩之间的关系。与之相对的是 HSB 模式，他是基于人眼对色彩的观察来定义的。在此模式中，所有的颜色都用色相（Hue）、

饱和度(Saturation)、亮度(Bright)三个特性来描述。除了上述两种外常用的还有 Lab 模式、CMYK 模式。这些模式不能说谁比谁更加的有效,只是应用的场景不一致,所以最有效的方法是将他们组合在一起来使用。

2、边缘:通常出现图像强度产生强烈的变化的区域,两个具有不同灰度值的相邻区域之间总存在边缘,边缘是灰度值不连续的表现。由于边缘是图像上灰度变化最剧烈的地方,传统的边缘检测就是利用了这个特点,对图像各个像素点进行微分或求二阶微分来确定边缘像素点。相对上面颜色特征来说边缘特征的一个重要的优势是他对光照不太敏感。

3、光流:光流是空间运动物体在观测成像面上的像素运动的瞬时速度。光流的研究是利用图像序列中的像素强度数据的时域变化和相关性来确定各自像素位置的“运动”,即研究图像灰度在时间上的变化与景象中物体结构及其运动的关系。一般情况下,光流由相机运动、场景中目标运动或两者的共同运动产生。光流计算方法大致可分为三类:基于匹配的、频域的和梯度的方法。

4、纹理:纹理是对物体表面细节的总称。他是物体外表的一种强度变化,这样一些诸如平滑和规则的变化确定了物体表面的特性。相较于颜色,纹理需要一个处理步骤以生成描述。与颜色相比,纹理特征对光照变化不敏感。

以上特征中,对颜色特征的应用是最广泛的,但其缺点也是显而易见的,对光照非常的敏感往往导致跟踪的失效。但是我们可以加入其它的特征来一起表示目标的外观,如光流等能够有效的抵抗光照的敏感性。目前大多数特征被用户根据应用手动的选择。然而,特征自动选择问题已得到相当多的关注。自动选择合适的特征是一个很有实用的方法,能够通过分类器或者一些提前设定好的标准去自动选择能够有代表性和极大差异性的特征,从而达到更好的跟踪效果。

## 2.3 目标检测跟踪分类

### 2.3.1 目标检测

目标跟踪有可能是每一帧都做检测也有可能只在首帧检测<sup>[3]</sup>。但无论哪一种跟踪方法都需要加入目标检测,这点是毫无疑问的。虽然大多数的目标检测都是在单帧图像上做的,但是也有少许考虑到了帧间的差异从而降低了误检、漏检的概率。目标检测考虑的图像信息对数学统计和一些物理信息关注的比较少,然而其检测的准确性对后续跟踪影响是不可忽视的。这里我们将常见的检测方法归类提出,主要有点探测器、背景相消法、图像分割法、监督学习法。

#### 1.点探测器法

我们现在所介绍的点探测器是探测图像中感兴趣的点。所谓感兴趣的点是在

图片中一些具有旋转、平移、拉伸等不变性，抗干扰力强稳定的一些点。这些点受光照等环境变化的影响不大，这是由于他们在其各自位置有其特定的纹理表现。特征点检测在图像处理中应用已经有很长时间了，从初期的Moravec探测器、Harris特征点探测器到KLT探测器和SIFT探测器。新的感兴趣点的计算方法不断被改进，所取得的效果也得到了很大的提高。

## 2. 背景差法<sup>[13]</sup>

背景差法首先选取背景中的一幅或几幅图像的平均作为背景图像建立相应的模型，然后把以后的序列图像当前帧和背景图像相减进行背景消去或者通过输入当前帧的图像和建立的模型进行偏差比较。若所得到的像素数大于某一阈值那我们就可以将其标记出来并对其做进一步的处理，比如由一些已知的方法来抗噪，抗干扰，滤波等，以便得到更为精准的目标位置坐标、个数等参数。这个处理的过程就是所谓的背景差法。

这种方法的优点是：

- (1) 其原理和算法设计简单；
- (2) 所得结果直接反映了运动目标的位置、大小、形状等信息，能够得到比较精确的反映运动目标信息。

缺点是受光线、天气等外界条件变化的影响较大。

## 3. 帧差法<sup>[6]</sup>

帧差法又称图像序列差分法或者帧间差分法。他利用的是帧间的差异像素直接相减来消除背景突出前景目标，不需要对背景建立模型，这是其区别于背景差法的关键所在。当监控场景中出现异常物体运动时，帧与帧之间会出现较为明显的差别。两帧相减得到两帧图像亮度差的绝对值，判断它是否大于阈值来分析视频或图像序列的运动特性。确定图像序列中是否有物体运动。图像序列逐帧的差分相当于对图像序列进行了时域上的高通滤波。

帧间差分法是一种比较成熟的算法，这种方法的优点是：

- (1) 实现简单，算法设计量小；
- (2) 对光线等场景变化不太敏感，能够适应各种动态环境，稳定性较好。

其缺点是：

- (1) 不能提取出对象的完整区域，只能提取出边界；
- (2) 依赖于选择的帧间时间间隔，对快速运动的物体需要选择较小的时间间隔，如果选择不合适，当物体在前后两帧中没有重叠时，会被检测为两个分开的物体，而对慢速运动的物体，应该选择较大的时间差，如果此时选择不适当，当物体在前后两帧中几乎完全重叠时，检测不到物体。

## 4. 图像分割

图像分割的目标是将图像分割成感兴趣的小区域，这里的感兴趣是根据目标

特性和一些先验知识来做的分割。比如像素大小, 形状等等标准。每个分割算法都试图解决两个问题一个是图像优良分割的标准, 另一个是得到有效分割的方法。用于目标跟踪技术的图像分割方法主要有 mean-shift 聚类、graph-cut 分割、Active-Contour 分割法。

### 5. 监督学习

目标检测可以通过一组样本来自动学习不同目标的外观来实现, 当然这组样本是要事先通过监督学习机制训练好的。给定一个例子集学习, 监督学习方法生成一个输出函数映射到所需的输入。有监督学习是标准制定的分类问题, 即学习者接近的过程。在检测范围内的对象时, 学习的例子都用手动对对象的特点和相关的对象类进行标记得来的。

特征选择对分类起着重要作用。除了上面所讨论的特征外, 它也可以使用目标方向, 目标区域和以密度函数表示的目标外观来作为特征, 如直方图。一旦特征被选定, 目标不同的外观能被采用调查学习的方法所学习和逼近。这种学习逼近的方法有神经网络, 自适应增强, 决策树, 支持向量机等。

### 2.3.2 目标跟踪

目标跟踪的目的是通过对每一帧目标的标记产生一个目标的轨迹同时将不同帧的目标关联形成对同一目标的坐标关联。目前大多数方法都是用概率的方法去计算目标下一帧出现的概率密度。最初的参数化运动模型已经难以满足复杂跟踪的需要, 目前非参数化估计优势明显得到了很快的发展和應用。如均值漂移, 粒子滤波等跟踪算法。目标检测和建立目标之间的联系可以分开来做也可以和在一起做。分开做的时候先将每一帧目标坐标检测出来然后用跟踪算法将其关联起来, 在一起做的时候当前帧的目标坐标和关联要根据对当前帧的目标检测和上一帧的目标坐标信息并结合跟踪算法即运动模型来计算得到。

无论采用上述哪种方式来进行跟踪, 我们都需要一个用来表示物体的模型。刚性物体用点或者几何形状如矩形、椭圆等来表示, 则其运动模型和形变方式就固定了下来, 如椭圆表征情况下使用衍射或者投影的参数化运动模型是合适的。非刚性物体用轮廓或者外形来建立表征模型, 其运动模型可以是参数化的也可以是非参数化的。

这里对目标跟踪进行的分类主要是从目标的外观形状来做区分的, 不同的目标表示使得跟踪算法所考虑的方面是不一样的。以下主要讲目标跟踪分为三大类:

- (1) 点跟踪: 这里目标在视频中可以用点来表示, 同时他们之间的关联需要基于其前一帧的位置和运动。这一种方式需要采用外部机制对其每一帧目标做检测;
- (2) 核窗跟踪: 这里的核心对目标而言当然是一个刚性的物体, 但是由于涉及

像素点多且具有一定的外在特征，所以不能简单的在图像视觉中理解为一个点。核心涉及到了目标的外观和形状。例如核心可以用一个矩形模板或者一个带直方图的椭圆型来表示。跟踪算法是通过计算核心在连续的帧中的运动来完成的。这种运动通常以一种参数化的转移来实现，比如平移、旋转和仿射；

(3) 轮廓跟踪：这种跟踪由于目标是非刚性的，他是由多个刚性部件所组成的，如近镜头下的人体运动等。即不能表示成点也不能简单用一个可以收缩的具有核心特性的模板和区域来表示。轮廓跟踪对目标内部进行信息编码，这种信息能以外观密度和形状模型来表示。

## 第三章 基于贝叶斯的跟踪算法

### 3.1 引言

用贝叶斯滤波器的理论描述一个跟踪任务就是如何获取一个关于目标状态的后验概率表示。卡尔曼算法是贝叶斯的最优闭合解，但是其缺点也是明显的，线性高斯的条件制约了其在目标跟踪上的应用。虽然粒子滤波是贝叶斯准则的次优解，但它却很自然的解决了非线性非高斯的限制，扩展了其在跟踪算法中的应用。具体来说卡尔曼算法主要是针对目标运动模型的建立并且利用最小均方准则作为概率判据，而粒子滤波算法利用了蒙特卡洛原理直接用随机采样对目标出现的后验概率进行估计。

本文的实验数据是基于卡内基梅隆大学的开源数据库，航拍目标主要对象是汽车，根据第一章分类，在视频中属于一个具有核心收敛性的目标区域，可以当做刚性物体来处理。同时一个十分重要的特点是目标相对视野较小且大小变换不大，这对于后续算法的选择是一个重要的指标。虽然这种区域刚性目标采用首帧目标检测的均值漂移算法对其区域像素建立概率密度模型是比较合适的方法，但是下一章所分析的运动补偿模块是必须在帧帧做配准，提取帧间的背景运动，这种机制与卡尔曼建立的目标状态转移矩阵和帧帧检测是不谋而合的。在经典均值漂移的算法中只要对首帧进行目标检测就可以了，之后的跟踪不需要每一帧都做目标检测，这种机制虽然也可以在下一帧初始定位对上一帧位置信息的引入上加入，但是没有卡尔曼这种方式来的自然，对在经过状态转移矩阵转移后的状态估计值进行背景运动的纠正后，可以较为准确的定位下一帧目标出现的位置而进行周围的搜索匹配。同时均值漂移对目标模型的建立要求较高，基本是应用在了具有一定外观表示的目标上，帧间目标必须有重合区域才能很好的进行匹配。

本章节首先介绍卡尔曼滤波然后引入粒子滤波中间详细阐述了两者间的关系和进行的具体改进。

### 3.2 卡尔曼算法

卡尔曼滤波器<sup>[18, 19]</sup>是在贝叶斯准则下以最小均方误差为判据的算法，在处理随机信号方面有其独到的地方，他可以处理非平稳随机信号，不要求保存过去的测量数据，当新的数据测得以后，根据新的数据和前一时刻的参数估计，借助于系统本身的状态转移方程，按照一套递推公式，即可算出新的参数估计值。这就大大减少了滤波装置的存储量和计算量，并且突破了平稳随机过程的限制。而且

只需要根据前一个估计值和最近的一个观测值就可以估计当前状态值。同时他也是贝叶斯估计的最优解。

### 3.2.1 卡尔曼在跟踪中的应用

我们这里的目标状态是否是一个随机信号呢, 无论是状态空间里的位置还是速度都可以理解为一个随机信号, 这也正是其可以应用到目标跟踪的大前提。所不同的是这里的目标状态是一个有着自身特点的随机信号, 目标这一时刻出现的位置和速度等状态我们都是未知的, 但是有一点可以理解的是状态信号的相关性受到上一个时刻状态的影响是最大的, 如果目标的当前帧的速度和位置这两个状态变量的值比较小的话, 那么在下一帧目标速度和位置这两个随机变量值的大小还是要过多的受制于前一帧的大小, 产生较大数值的可能性是比较小的, 这也是由目标运动的惯性所决定的。在这也迎合了卡尔曼的处理数据的要求。

科学上许多问题的解决都可以用加载一组噪声的时变系统的状态估计来解决。卡尔曼滤波器是以动态的状态方程和观测方程来描述系统。动态系统是指非线性时变系统。而离散时间逼近的方法<sup>[20]</sup>广泛适用于对动态系统的估计。这里目标的运动建立的数学模型到底是否是一个动态系统呢, 这对于状态空间中不同的状态分量来说有不同的理解。

如果我们这里只考虑目标的一个状态即位置, 那么状态的非线性程度导致后续的运算推导很难进行。虽然大多数情况下我们需要得到的状态是目标的位置状态, 而且通过后续的处理也可以得到一些诸如速度的其他状态。当然目标位置这个状态很容易理解过的是一个动态系统, 如果按照物体的运动来理解, 前一个位置状态在不同的坐标系中对后一个位置状态的影响绝对不是线性的, 是不可能用线性的方程来表示这两帧间的位置状态的。除非坐标系的零点永远在目标的下一帧的起始位置。这一点是很好理解的在同一个坐标系下位置的大小强度是线性的话, 位置越大下一次的转移将会越剧烈。

如果目标的运动理解为一个动态系统, 那么对动态系统的处理就必须要有两个模型的建立。一个随时间变换的状态演化模型(状态转移模型), 一个加载噪声的模型(观测模型)。

假设目标的状态序列  $\{\mathbf{x}_k, k \in N\}$  的关系式为:

$$\mathbf{x}_k = A_k(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}), \quad (3-1)$$

此处,  $A_k(\mathbf{x}_{k-1}, \mathbf{w}_{k-1})$  是一个关于状态  $\mathbf{x}_{k-1}$  的非线性函数,  $\mathbf{w}_{k-1}$  为状态噪声序列,  $n_x$ 、 $n_w$  分别是状态和过程噪声矢量的维数。跟踪的目的是通过下面的测量方程递归地估计状态  $\mathbf{x}_k$ 。

$$\mathbf{y}_k = c_k(\mathbf{x}_k, \mathbf{v}_k), \quad (3-2)$$

此处,  $c_k(\mathbf{x}_k, \mathbf{v}_k)$  也是一个非线性函数,  $\mathbf{v}_k$  为观测噪声。在跟踪过程中, 本文是通过一直到  $k$  时刻的所有累计测量的集合  $\mathbf{y}_{1:k} = \{\mathbf{y}_i, i=1, 2, \dots, k\}$  来估计目标的状态  $\mathbf{x}_k$ 。

我们将假设这些模型以一种概率的形式构建, 而这种状态概率的进化和接受新观测数据的方式正迎合了贝叶斯估计。贝叶斯对动态状态的估计是要基于所有可用统计信息的前提下构建状态的后验概率密度函数 PDF, 而一个状态最优的估计就来自于 PDF。

我们可通过预测和更新两个步骤递归地得到  $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ 。设  $k-1$  时刻的概率密度函数  $p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1})$  是已知的, 这两个步骤可表示为:

预测步骤:

$$p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1}, \quad (3-3)$$

上式的状态转移模型利用了一阶马尔可夫过程  $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{y}_{1:k-1}) = p(\mathbf{x}_k | \mathbf{x}_{k-1})$ , 目标的状态转移概率密度  $p(\mathbf{x}_k | \mathbf{x}_{k-1})$  由状态方程 (4-1) 式和状态噪声  $\mathbf{w}_{k-1}$  的统计特性决定。

更新步骤:

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k | \mathbf{y}_{1:k-1})}, \quad (3-4)$$

在上式的更新过程中,  $\mathbf{y}_k$  被用于修正先验概率密度来获得当前状态的后验概率密度函数。其中 (3-3)、(3-4) 式循环执行就构成了最优的贝叶斯估计的一般表达式。

### 3.2.2 卡尔曼滤波的条件和算法步骤

在上述对贝叶斯的完美推导中我们可以看出构建两套系统, 一个状态转移一个观测纠正, 是符合其预测和更新步骤的。但是在具体数学推导中, 我们构建的公式 (3-1) 和 (3-2) 是不能进行的, 其原因主要有两点:

1. 非线性系统无法将相邻状态分开进行概率统计;
2. 噪声概率的不确定导致后验概率密度的推导更新无法进行。

为此卡尔曼滤波器假设任意时间状态的后验概率密度都是高斯分布的, 也就是说, 如果是  $p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1})$  高斯分布的, 并且存在以下假设:

- 1、状态噪声  $\mathbf{w}_{k-1}$  和观测噪声  $\mathbf{v}_k$  是服从高斯分布的噪声;
- 2、 $\mathbf{x}_k$  是  $\mathbf{x}_{k-1}$  和  $\mathbf{w}_{k-1}$  的一个线性方程, 同时  $y_k$  是  $\mathbf{x}_k$  和  $\mathbf{v}_k$  的一个线性方程。

那么就可以证明  $p(\mathbf{x}_k | \mathbf{y}_{1:k})$  也是高斯分布的。同时  $\mathbf{x}_k$  是一个状态空间, 将状态之间的转移通过转移矩阵  $A_k$  来做线性表示, 因此卡尔曼的全部参数都可以用均值和方差来表示。公式 (3-1) 和 (3-2) 相应的变成卡尔曼滤波的两个方程:



状态转移方程:

$$\mathbf{x}_k = A_k \mathbf{x}_{k-1} + \mathbf{w}_{k-1}, \quad (3-5)$$

观测转移方程:

$$\mathbf{y}_k = \mathbf{c}_k \mathbf{x}_k + \mathbf{v}_k, \quad (3-6)$$

上式中  $\mathbf{x}_k$  是系统的状态向量,  $\mathbf{y}_k$  是观测向量,  $A_k$  是状态转移系统矩阵,  $\mathbf{c}_k$  是观测转移矩阵。  $\mathbf{w}_{k-1}$  和  $\mathbf{v}_k$  的协方差  $Q_{k-1}$  和  $R_k$  分别是均值为零并且统计独立的高斯白噪声。且状态转移及观测矩阵  $A_k$  和  $\mathbf{c}_k$  以及噪声参数矩阵  $Q_{k-1}$  和  $R_k$  都可以为动态矩阵。

我们最后所需要得到的是当前帧下的目标出现的概率密度函数  $\rho(\mathbf{x}_k | \mathbf{y}_{1:k})$  来估计当前的状态  $\mathbf{x}_k$ 。理论上, 通过递归贝叶斯滤波可以得到优化解, 结合贝叶斯的预测和更新公式 (3-3) 和 (3-4), 可以把整个滤波过程分为两个阶段:

预测阶段 (通过状态转移矩阵预测下一帧状态):

通过使用状态方程和已计算出的  $k-1$  时刻的概率密度函数  $\rho(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1})$ , 导出当前时刻的先验概率密度函数  $\rho(\mathbf{x}_k | \mathbf{y}_{1:k-1})$ 。公式如下:

$$\rho(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) = N(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, P_{k-1|k-1}^{XX}) \quad (3-7)$$

$$\rho(\mathbf{x}_k | \mathbf{y}_{1:k-1}) = N(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}^{XX}) \quad (3-8)$$

上面公式中,  $N(\mathbf{x}; \hat{\mathbf{x}}, P)$  表示参数  $\mathbf{x}$  服从均值为  $\hat{\mathbf{x}}$  和协方差  $P$  的高斯分布。

首先由公式(3-5) (3-6) 进行预测:

K时刻的状态均值  $\hat{\mathbf{x}}_{k|k-1} = A_k \mathbf{x}_{k-1|k-1}$ ;

K时刻的状态方差  $P_{k|k-1}^{XX} = Q + A_k P_{k-1|k-1}^{XX} A_k^T$ ;

K时刻的观测值  $\hat{\mathbf{y}}_{k|k-1} = \mathbf{c}_k \hat{\mathbf{x}}_{k|k-1}$ ;

K时刻的观测方差  $P_{k|k-1}^{YY} = R + \mathbf{c}_k P_{k|k-1}^{XX} \mathbf{c}_k^T$ ;

K时刻的状态和观测的互相关矩阵  $P_{k|k-1}^{XY} = P_{k|k-1}^{XX} \mathbf{c}_k^T$ ;

上述推导过程都是在依据噪声服从均值为0的高斯白噪声, 且后验为高斯分布的条件下进行的。

更新阶段 (通过观测矩阵加入观测值修正预测值):

使用当前量测的密度函数  $\rho(\mathbf{y}_k | \mathbf{x}_k)$  来计算后验概率密度函数  $\rho(\mathbf{x}_k | \mathbf{y}_{1:k})$ 。计算公式为:

$$\rho(\mathbf{x}_k | \mathbf{y}_{1:k}) = N(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, P_{k|k}^{XX}) \quad (3-9)$$

上面公式中,  $N(\mathbf{x}; \hat{\mathbf{x}}, P)$  表示参数  $\mathbf{x}$ , 均值  $\hat{\mathbf{x}}$  和协方差  $P$  的高斯密度函数, 用观测值更新预测, 也就是最终获得的递推公式:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + H_k (\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}), \quad (3-10)$$

上式中  $H_k = P_{k|k-1}^{XY} [P_{k|k-1}^{YY}]^{-1}$  是卡尔曼增益方程,  $P_{k|k}^{XX} = P_{k|k-1}^{XX} - H_k P_{k|k-1}^{YY} H_k^T$  是估计误差的协方差矩阵。

在卡尔曼滤波算法启动前,我们通常都将观测噪声的协方差  $R_k$  认为是一致的。实验中通常通过一些观测值样本得到观测噪声的协方差。然而待估计的过程一般不能直接观测,所以状态噪声的协方差  $Q_{k-1}$  相对难以决定。有时在相对简单的过程模型中,当观测过程相当可靠时,通过  $Q_{k-1}$  引入一定不确定的噪声,依然能够得到可接受的结果。实际上,根据统计理论,无论是否有合理的选择噪声参数的基础,滤波过程通常可以通过调整滤波参数  $R_k$  和  $Q_{k-1}$  完成。只有当  $R_k$  和  $Q_{k-1}$  都是常数时,估计误差的协方差和滤波器的增益将会快速收敛。

### 3.3 粒子滤波算法

在具有加性高斯噪声的条件下,卡尔曼滤波器是贝叶斯估计的最优解。但是现实中线性高斯条件绝大多数是不满足的,使用线性的简单的 Kalman 滤波器很难对一个实际的跟踪任务产生好的效果。我们为了现实应用的需要,必须要找到一种能够解决非线性非高斯信号处理的机制。

上一节我们已经发现了贝叶斯估计的完美闭合解是卡尔曼滤波,通过 (3-3)、(3-4) 我们不难发现两个方程中我们首先需要得到的是目标的状态后验概率密度函数  $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ 。在已知上一帧的后验密度和先验密度的条件下我们要如何才能得到下一帧的后验呢? (3-3) 式子中给出了一个漂亮的答案,一个简单的积分和先验概率密度函数  $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ 。在卡尔曼中这个积分被几个假设轻易化解了,在满足线性高斯条件下,以后每一帧的目标出现的后验概率密度函数  $p(\mathbf{x}_k | \mathbf{y}_{1:k})$  都是一个漂亮的高斯分布<sup>[21]</sup>,所以 (3-3)、(3-4) 改写成了 (3-5)、(3-6),从而可以展开具体算法推导。而实际计算这个积分是很难的,没有卡尔曼的假设我们几乎不可能开展任何数学推导。如何破解这个积分成为破解线性高斯假设的关键,也是信号处理领域的一个突破。

定积分是可以近似通过无穷项相加得到的,这一点我们早已知道,那是否可以在此处破解呢?答案是肯定的,虽然无穷项很难实现但是这毕竟是一种近似完美的逼近。Monte Carlo 积分为我们提供了这样一种机制,这也是粒子滤波的原理。该方法是一种通过 Monte Carlo 仿真实现非线性递归 Bayesian 滤波的方法,它采用一组带权的粒子来近似运动状态变量的后验概率密度分布。由于粒子滤波器具有处理非线性、非高斯信号的能力,因此得到了广泛的应用。

#### 3.3.1 蒙特卡洛方法

蒙特卡罗方法 (Monte Carlo method) 也称随机模拟法 (stochastic simulation) 或者统计实验法<sup>[25]</sup>。他是以概率统计理论为基础,依据大数定理 (样本均值替代

总体均值), 利用电子计算机数字模拟技术, 解决一些很难直接用数学运算求解或用其他方法不能解决的复杂问题的一种近似计算。其在数学中最常见的应用就是蒙特卡罗积分。

不少统计问题, 如计算概率、各阶矩阵, 最后都归结为定积分的近似计算问题。概率是用于描述不确定性事件发生的可能性的, 对于一些比较简单的随机事件的概率计算问题, 我们可以通过一些比较常用的概率计算公式进行准确的计算, 但是随着研究的深入复杂的概率问题的出现, 计算的困难度陡然增大, 甚至有时候根本无法计算。

蒙特卡洛方法的基本思想是针对某一具体问题, 通过建立一个概率模型或者随机过程的模型来模拟具体问题的参数, 从而达到间接求解该问题的解决方案。定积分的计算是蒙特卡洛方法引入计算数学的开端, 同时许多计算积分的问题上, 此方法一般都能有效的解决。尽管其计算精度不是很高, 但是结果至少可以作为真实值的参考。其求解一维定积分的方法有随机投点法和平均值法。这里对两种方法做一个简要的概述:

#### (1) 随机点法

假设所求积分为  $U = \int_a^b f(x)dx$ , 其中  $a, b$  为有限数, 被积函数  $f(x)$  为连续随机变量的概率密度函数。步骤为:

- 1、首先产生服从给定分布的随机变量值,  $X_1, X_2, X_3, \dots, X_n$
- 2、检查  $X_n$  落入积分区间  $[a, b]$  的次数, 然后与实验总次数相比即为所求概率积分  $U$  的近似值。

#### (2) 样本平均值法

随机点法要求被积区间有界, 且被积函数有界, 在现实中这两个条件是不一得到满足。因此我们一般采用样本平均值法。样本平均值法的基本原理是当我们所求积分为  $U = \int_a^b f(x)dx$  时, 假设  $g(x)$  是  $(a, b)$  上的一个密度函数, 我们可以将积分改写成  $U = \int_a^b \frac{f(x)}{g(x)} g(x)dx = E\left[\frac{f(x)}{g(x)}\right]$ , 将积分表示为某个随机变量的期望。根据矩估计法, 若有  $n$  个来自  $g(x)$  的观测值, 则可以给出  $U$  的一个矩估计。这里  $a, b$  都可以为无界值, 在具体求解过程中需要考虑到此因素进行具体分析。

综上所述, 蒙特卡罗方法的解题过程都是要通过某种实验得到该事件出现的概率或者是这个随机变量的平均值, 并用他们作为问题的近似解。具体中并不一定要进行真实的实验, 而是通过事物运动过程的数量和几何特性, 利用数学方法加以模拟, 进行数字模拟实验, 实验次数越多其结果就越完美。有时候甚至要进行上百万次的数字模拟实验。我们首先需要构造或描述概率过程, 使得它的某些参量正好是所要求问题的解。然后实现从已知概率分布抽样, 与从  $(0, 1)$  上均匀分布

抽样不同, 这些方法都是借助于随机序列来实现的。最后建立各种估计量, 作为所要求的问题的解, 我们称它为无偏估计, 也是我们最后得到的问题解。

### 3.3.2 序贯重要性采样算法

SIS (Sequential Importance Sampling) 方法又称序贯重要性采样算法, 是粒子滤波器最基本的一种形式, 通过 Monte Carlo 模拟来实现递归滤波。 $\{x_{0:k}^i, \omega_k^i\}_{i=1}^{N_s}$  表示后验概率密度函数  $P(x_{0:k} | y_{1:k})$  特性的随机测量集合, 其中, 粒子状态  $\{x_{0:k}^i, i=1, 2, \dots, N_s\}$  是具有关联权值  $\{\omega_k^i, i=1, 2, \dots, N_s\}$  的点集,  $x_{0:k} = \{x_j, j=0, 1, \dots, k\}$  表示直到  $k$  时刻的所有状态的集合, 对权值归一化使得  $\sum_{i=1}^{N_s} \omega_k^i = 1$ , 这样  $k$  时刻的后验密度可表示为:

$$P(x_{0:k} | y_{1:k}) \approx \sum_{i=1}^{N_s} \omega_k^i \delta(x_{0:k} - x_{0:k}^i) \quad (3-11)$$

上式即为后验密度  $P(x_{0:k} | y_{1:k})$  的一个离散加权表达式, 权值需要通过重要采样 (Important Sampling) 选择, 这是因为, 一般情况下, 本文难以直接从真实分布  $P(x_{0:k} | y_{1:k})$  中抽取采样, 而是利用某些易于采样的分布, 例如:  $q(x_{0:k} | y_{1:k})$ , 实现采样过程, 该分布被称为重要性密度函数。

设  $x_{0:k}^i$  为从重要密度  $q(x_{0:k} | y_{1:k})$  中抽取的采样, 则权值为:

$$\omega_k^i \propto \frac{P(x_{0:k}^i | y_{1:k})}{q(x_{0:k}^i | y_{1:k})} \quad (3-12)$$

对于序贯情况而言, 每一次迭代时, 我们已经得到了用来近似密度  $P(x_{0:k-1} | y_{1:k-1})$  的采样集合, 并且希望用一个新的采样集合近似  $P(x_{0:k} | y_{1:k})$ , 如果所选择的重要采样能分解成如下形式:

$$q(x_{0:k} | y_{1:k}) = q(x_k | x_{0:k-1}, y_{1:k}) q(x_{0:k-1} | y_{1:k-1}) \quad (3-13)$$

在这种情况下, 就可在已有采样  $x_{0:k-1}^i \sim q(x_{0:k-1} | y_{1:k-1})$  的基础上, 增加一组新采样  $x_k^i \sim q(x_k | x_{0:k-1}, y_{1:k})$ , 从而获得  $q(x_{0:k} | y_{1:k})$  的采样  $x_{0:k}^i$ 。由于:

$$p(x_{0:k} | y_{1:k}) \propto p(y_k | x_k) p(x_k | x_{k-1}) p(x_{0:k-1} | y_{1:k-1}) \quad (3-14)$$

因此, 权值可表示为如下的递推形式:

$$\omega_k^i \propto \omega_{k-1}^i \frac{P(y_k | x_k^i) P(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{0:k-1}^i, y_{1:k})} \quad (3-15)$$

如果  $q(x_k | x_{0:k-1}, y_{1:k}) = q(x_k | x_{k-1}, y_k)$ , 这时, 重要密度仅依赖于  $x_{k-1}$  和  $y_k$ , 这一点非常有用。在这种情况下, 只需保存当前的粒子  $x_k^i$ , 可以丢弃其他的路径

粒子  $x_{0:k-1}^i$  及历史观察  $y_{1:k-1}$ 。这样, 权值可表示为如下的递推形式:

$$\omega_k^i \propto \omega_{k-1}^i \frac{P(y_k | x_k^i) P(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, y_k)} \quad (3-16)$$

于是, 后验滤波密度  $P(x_k | y_{1:k})$  可以被近似表示为:

$$P(x_k | y_{1:k}) \approx \sum_{i=1}^{N_s} \omega_k^i \delta(x_k - x_k^i) \quad (3-17)$$

### 3.3.3 重采样

SIS 粒子滤波器所遇到的一个常见问题是粒子的退化现象, 其表现形式是经过几个循环之后, 除了一个粒子之外, 其他所有粒子的权值小的可以忽略。已经证明重要权值的方差只会随着时间的持续而增加, 因此这种退化现象是不可避免的。这种退化现象也隐含了对计算  $P(x_k | y_{1:k})$  几乎没有贡献的一个巨大的运算负担, 一般引入一个有效采样大小的参数  $N_{eff}$ , 为算法的退化程度提供了一个合适的量度, 其定义为:

$$N_{eff} = \frac{N_s}{1 + \text{var}(\omega_k^i)} \quad (3-18)$$

其中,  $\omega_k^i = P(x_k^i | y_{1:k}) / q(x_k^i | x_{k-1}^i, y_k)$  指的是给粒子的权值, 这个值不能准确地得到, 但是可通过下式得到  $N_{eff}$  的估计值  $\hat{N}_{eff}$ 。

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s} (\omega_k^i)^2} \quad (3-19)$$

其中  $\omega_k^i$  是由 (3-15) 式得到的规范化值。注意到  $N_{eff} < N_s$ , 并且  $N_{eff}$  越小表明退化越严重。解决退化现象的一个简单办法是选择一个非常大的  $N_s$ , 这种方法显然是不实用的。除此之外, 还可采用另外两种方法解决这一问题: 1) 选择一个好的重要密度。2) 增加一个重采样过程。由于最优重要性密度函数不易选择。因此, 通常采用重采样过程解决这一问题, 下面对重采样步骤的实现过程作简要的说明。

程序是在判定  $N_{eff}$  小于某个设定的门限  $N_r$  的情况下启动重采样过程的。重采样的基本思想是消除无效的粒子, 这些粒子的权值已经变得很小, 需要我们将采样粒子集中在权值大的区域。这一过程是通过 (3-17) 式所表示的离散近似密度的  $N_s$  次采样实现的, 由  $N_s$  次采样得到一个新的集合  $\{x_k^{i*}\}_{i=1}^{N_s}$ , 并且有  $P_r(x_k^{i*} = x_k^j) = \omega_k^j$ 。采样的结果事实上是从式 (3-17) 所表示的离散密度中获得的一组独立同分布的样本, 因此所有粒子这时的权值皆为  $1/N_s$ 。重采样保证了粒

子滤波算法的顺利序贯，增强了粒子的有效性，是粒子滤波中不可缺少的一个重要环节。



## 第四章 基于 SIFT 特征点的运动补偿方案

### 4.1 引言

和其他跟踪任务一样，航拍视频中的运动目标跟踪算法同样受着图片噪声，场景光线和伪目标的影响。但是航拍视频最大的特点是载具的运动和由此导致跟踪系统的不稳定，目标自身的运动是不受载具影响的，但是背景相对载具的运动是不可避免也是未知的。有的时候由于载具运动的强度过大有可能导致下本来在做某一方向运动的目标在下一帧中会出现沿着此方向上的倒退运动。这个整个跟踪系统带来的影响是巨大的，尤其是在不做任何处理的情况下。如图 4.1 所示每隔 10 帧背景的变化是较大的，右边的的树木以一种可见的速度出现，

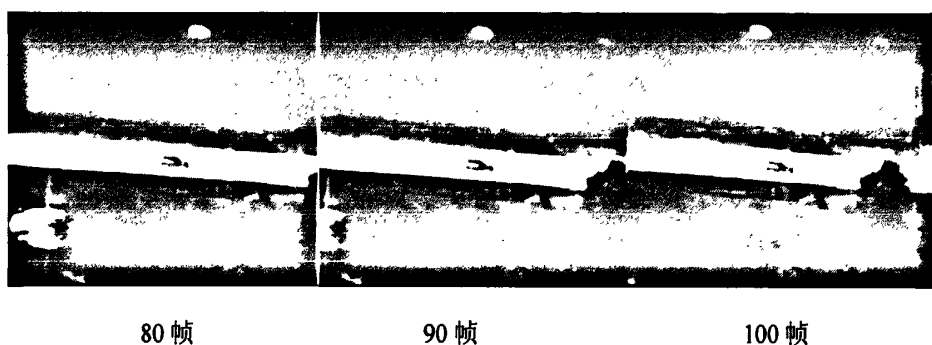


图 4.1 航拍场景

本章首先对跟踪算法中的抗载具方法进行介绍。分析了算法本身的抗载具运动能力，其中以卡尔曼跟踪算法，粒子滤波跟踪算法和 Meanshift 跟踪算法为例进行单独阐释。根据研究需要选择了基于 SIFT 特征点运动补偿算法作为载具运动的解决方案，介绍了基于 SIFT 特征点的原理和算法步骤。最后还分析了仿射变化用来描述载具运动即背景运动的依据和原理。

### 4.2 抗载具运动方法

抗载具运动的方法主要有两种：一种是跟踪算法自身所具有的抗干扰能力；另外一种是在加入运动补偿算法抵消载具运动。这里我们来做具体分析。

#### 4.2.1 跟踪算法抗载具运动

由第二章可以知道，跟踪算法是多种多样的，有的是针对点，有的是针对区域的，有的是要建立目标的运动模型然后通过观测输入进行纠正，如卡尔曼和粒子滤波；有的则不需要建立运动模型而建立的是目标外观的模型从而进行一系列



搜索和匹配,如 Meanshift 算法。其中大部分算法是可以在一定程度上对抗摄像头运动的,如果是针对区域目标的跟踪那么他一定是可以抗摄像头干扰的,因为目前大多数都是要在其上一帧的周围进行一定程度的搜索和匹配的。那么如果是卡尔曼和粒子滤波这种前期主要是针对点目标进行跟踪的算法呢?只要是建立了运动模型的输入我们都是要对其运动进行一些纠正和补偿的。如卡尔曼跟踪点目标的情况,但是粒子滤波是否一定需要呢?以下我们对常用的跟踪算法卡尔曼,粒子滤波和 Meanshift 算法进行详细的解释说明:

### 1、卡尔曼算法

在卡尔曼跟踪算法中我们建立的目标运动模型是一个动态系统,先由上一帧的信息对目标下一帧运动位置和速度等各种状态进行估计,然后由目标检测系统送来的观测值对估计值进行纠正。

这两个步骤中前期估计步骤深受载具的运动的影响。这里估计出来的目标坐标和速度是两个运动的合成,是依据上一帧的合成运动来进行的估计。合成运动一个是目标自身的独立运动,另外一个载具的运动,假设当前时刻的坐标是  $(x, y)$ ,载具的运动速度是  $(3,4)$ ,目标独立的运动速度是  $(5,6)$ ,那么下一帧目标检测下理想的结果是  $(x+8,y+10)$ 这和目标的运动是不相符合的,目标真实的运动是  $(x+5,y+6)$ 。这样的结果去预测下一帧的坐标和速度对算法产生的影响是可想而知的,而且由于这一帧的信息还要通过状态转移方程积累到下一帧,所以最后可能导致系统不稳定和跟踪的失效。

但是只要在目标检测中我们能输入准确的检测结果还是能够适当纠正由于估计带来的错误影响的。只要摄像头载具运动的强度不大,非线性程度不高的情况下还是能够保证算法的运行。

### 2. 粒子滤波

粒子滤波的前半部分可以延续卡尔曼的状态转移方程用来对目标下一帧进行估计,也可以依据上一帧位置直接投放粒子。当然投放的精度是存在影响的,正确的估计还是能对粒子的后续加权起到良好的作用从而提高概率密度的准确度。这是由于粒子滤波中重要性密度函数选取的原理得到的。载具的运动带来的扰动主要是对粒子投放扩散的扰动。

由于投放粒子理论上是全局投放的,所以只要在后续加权中能够得到精确的加权我们的算法在一定程度上是能投保证算法的正常运行的。但是初期的加权一般是取得各个粒子到估计值的各种特征距离,所以对概率密度的错误估计是明显的,给算法造成了负担,甚至是无法收敛。

### 3. 均值漂移

均值漂移算法 (Meanshift) 和上述的两种算法不一致的地方是其主要用在目标区域进行跟踪,这一点取得了不错的效果。由于前两种算法在前期跟踪应用

中将目标理解为漂亮的点,所考虑的图片特征不是很明显,虽然在后续研究中陆续加入了颜色等特征的考虑让其能更加适应于区域目标跟踪,但是这几种算法的本质都是一样的。他们都是要对目标出现的概率密度进行估计,找到其演化的规律,用其出现的最大概率密度推出其出现位置。在此算法中,下一帧的前期估计还是要依靠上一帧的位置得出,和上述两种算法不同的是没有采用运动模型即状态转移估计,而是直接在上一帧的位置上搜索计算。这是因为区域目标在帧间一般都是具有重合区域的,这在载具运动强度不大的时候也是适用的,当前摄像机的采样频率也为此提供了可靠保证。所以相对来说载具的运动对区域目标尤其是相对背景所占比例较大情况下在此算法下是可以抵消的。

综上所述三种算法都具有一定的抗干扰能力。其中卡尔曼是由于其两个步骤中的最后更新一步采用了观测值来纠正预测,而粒子滤波即使前期针对了目标的运动建立了模型的输入和观测的更新,但是由于后期的粒子投放的方式是全局的所以还是具有很强的抗干扰能力。Meanshift 算法的匹配过程是以上一帧目标的位置开始找寻,由于是针对目标区域所以重合的部分保证了最终能找寻到概率密度的最大。但是以上方法在载具强度加大时候,由于卡尔曼的误差积累、粒子权值的不准或者模板匹配的负担加重会造成系统不稳定,得下一帧的概率密度演化的误差增大,造成算法失效。

#### 4.2.2 运动补偿算法

本文所讨论的跟踪系统采用的是运动补偿算法。在后续跟踪算法中将载具的运动从全局运动中剔除,留下目标独立的运动。虽然采用的粒子滤波具有全局的搜索能力,但是在下一帧粒子投放精度上我们得到了保证,概率密度的演化精度提高了,减少了粒子加权计算的负担。全局动态补偿效果的好坏也直接关系到搜索区域的计算是否精确。而搜索区域计算的是否精确直接关系到跟踪任务的成败。采用精确度不高的补偿算法的跟踪系统是无法稳定跟踪运动目标的。航拍中的视频与普通的摄像头获取的视频不同。固定镜头下获取的视频中对目标跟踪不需要考虑摄像头的运动。在视频序列帧中显示出目标的位置变化实际上是由于目标的独立运用和摄像头的运动两种运动合成的效果造成的。我们可以将目标在视频帧中表现出来的速度理解为目标自身的速度与摄像头速度的合成。同样的道理也可以放在位置状态参数上来理解,既然速度是目标自身的速度和摄像头的速度的合成,那么运动位置也是一个合成的参数了。在一些跟踪算法中,这两个状态有可能都会用到,至少需要用到位置状态,如果观测的输入没有进行处理就输入算法中的话会引起算法的不稳定,所以对状态参数进行补偿是必须的。运动补偿在算法对航拍跟踪系统是不能缺少的一个模块。

运动补偿算法的实质是找到两幅图像之间的运动信息，并且将其提取出来，我们这里也可以认为是载具（摄像头）的运动信息。这里我们采用的是基于静态图像配准算法的技术，从图像中找寻其一致之处将其关联起来。

### 4.3 图像配准方法

图像配准算法可以分为基于像素的方法和基于特征的方法，其中基于特征的方法又包括基于特征区域的方法和基于特征点的方法。我们这里采用的是基于特征点的方法。

要如何建立两帧图片之间变换的几何模型呢？关键在于找到两帧图片中相似的点，同时也要建立起这些相似点间的一对一关系。只有这样才能通过几何变换描述两帧图片之间的变换关系。我们在这里不是要对图片中每一个像素点进行匹配，而是使用检测特征点算法检测出图像中具有特征的点，然后在这些点之间建立联系。

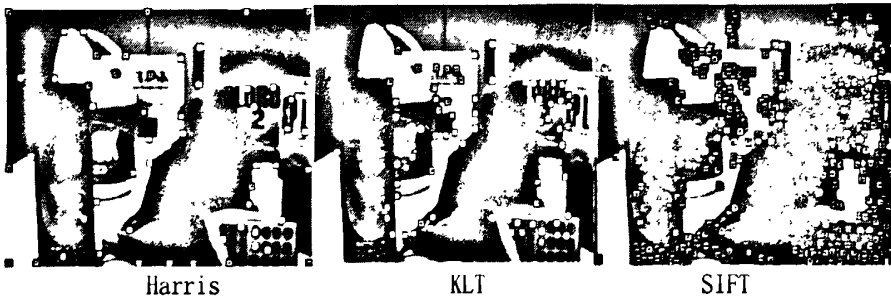


图 4.2 特征点探测实例<sup>[3]</sup>

第二章我们介绍过常用于视频应用中经典的特征点有 Moravec、Harris 特征点和 KLT 点跟踪以及 SIFT 点。前面三个都是基于找寻图像中角点的，Harris 改进了 Moravec 提出的角点检测算法，直接计算角量函数 (corner score)，等价于自相关函数对  $x$ 、 $y$  方向的微分，大提高了检测效率精确度。而 Moravec 是使用移动的块匹配 (shifted patches) 的计算自相关函数，定义小的自相关值的点为角点，这种方法计算量大而且往往得不到最优解。与 Harris 提出的算法不同的是 KLT 算法给出了如何建立特征点之间的联系的方法，按 KLT 的术语叫做特征点跟踪。与使用匹配方法建立联系不同的是特征点跟踪只需要在第一帧的图像内检测特征点，在后续的视频帧中只需要在搜索区域中搜索即可，省去了检测特征点的时间。同时 KLT 对 Harris 所选取的特征点阈值也提出了自己的改进。我们这里要介绍的是 SIFT 特征点，他的主要思想和上述三者有所不同。

#### 4.3.1 SIFT 点介绍

这里我们选取 SIFT 点探测器对图片特征点进行探测。SIFT 特征英文全称为

Scale-invariant feature transform,即尺度不变特性转换。它是由 David. Lowe 教授于 1999 年提出, 2004 年完善总结。该方法提出的是一种基于尺度空间极值的特征点提取算法和局部特征表述方法。

SIFT 算法是一种提取局部特征的算法, 在尺度空间寻找极值点, 提取位置, 尺度, 旋转不变量。SIFT 特征是图像的局部特征, 其对旋转、尺度缩放、亮度变化保持不变性, 对视角变化、仿射变换、噪声也保持一定程度的稳定性。在只有少数几个物体的情况下也可以产生大量 SIFT 特征向量。

#### 4.3.2 SIFT 点生成步骤

首先我们将输入的图像与不同的高斯核卷积得到高斯尺度空间, 并将由不同的高斯核卷积得到的高斯图像两两相减生成高斯差分尺度空间(DOG scale-space)。在相邻的三个尺度内寻找极值点, 即关键点。以它邻域像素的梯度方向分布特性为每个关键点指定方向参数, 确定一个 SIFT 特征区域, 生成 128 维的特征点描述字, 实现图像的特征提取。在我们得到两幅特匹配图像的 SIFT 特征之后, 采用关键点特征向量的欧式距离作为特征点的相似性测度。取参考图像中的某个关键点。并找出当前图像中与其欧式距离最近的前两个关键点, 若最近距离, 次近距离的值小于某个阈值, 则接受选一匹配点对。降低该阈值, SIFT 匹配点数将减少且更加稳定。初次匹配后得到若干对候选匹配点, 但是由于这里的特征点匹配结果不一定完全正确, 需要我们再次对其进行处理筛选。

具体步骤如下:

##### 1、高斯尺度空间的生成

尺度空间理论目的是模拟图像数据的多尺度特征。高斯卷积核是实现尺度变换的唯一线性核, 于是一幅二维图像的尺度空间定义为:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (4-1)$$

其中  $G(x, y, \sigma)$  是尺度可变高斯函数,  $(x, y)$  是空间坐标,  $\sigma$  是尺度坐标。

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (4-2)$$

为了有效的在尺度空间检测到稳定的关键点, 提出了高斯差分尺度空间(Difference of Gaussian scale-space), 简称 DOG。他是利用不同尺度的高斯差分核与图像卷积生成, 其中  $k$  是一常数, 如式子 4-3 所示:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (4-3)$$

DOG 算子计算简单, 是尺度归一化的 LOG 算子的近似。Lowe 教授建议 octave 之间的  $\sigma$  是二倍的关系。在一个 octave 中, 其和不同尺度高斯函数卷积的图像数目是恒定的。这里的 octave 代表一组图像, 下一组的图像由上一组图像降采样得

到, 如果降采样因子为 2, 即对上一组图像进行奇数点或者偶数点采样, 对于二维图像来说就是两个方向上的取样了。图 4.3 为两组高斯尺度空间图像的构建示例。

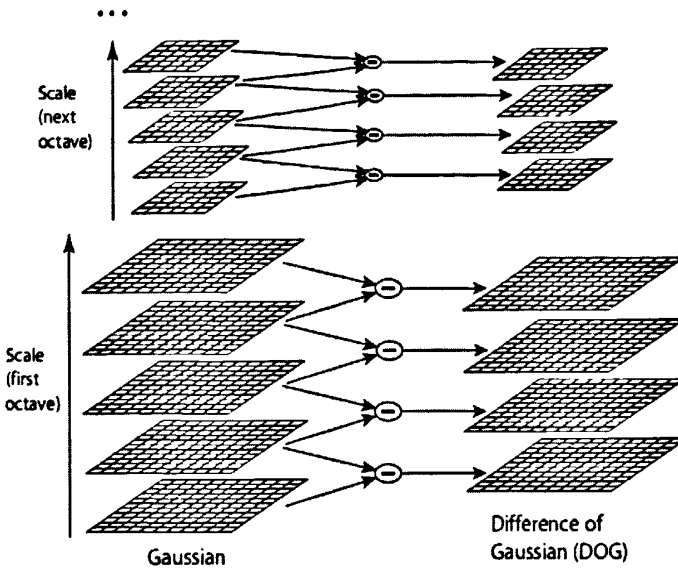


图 4.3 DOG 算子的构建

以下是构建尺度空间需确定的参数:

- O—octave 坐标
- S—sub-level 坐标
- $\sigma$ —尺度空间坐标

其中  $o \in o_{\min} + [0, \dots, O-1]$ ,  $s \in [0, \dots, S-1]$ ,  $\sigma(o, s) = \sigma_0 2^{o+s/S}$ 。空间坐标  $x$  是组 octave 的函数, 设  $x_0$  是  $o$  组的空间坐标, 则

$$x = 2^o x_0, o \in Z, x_0 \in [0, \dots, N_0-1] \times [0, \dots, M_0-1]$$

如果  $(M_0, N_0)$  是基础组  $o=0$  的分辨率, 则其他组的分辨率由下式获得:

$$N_o = \left\lfloor \frac{N_0}{2^o} \right\rfloor, M_o = \left\lfloor \frac{M_0}{2^o} \right\rfloor$$

Lowe 教授的文章建议使用如下参数:

$$\sigma_n = 0.5, \sigma_0 = 1.6 \cdot 2^{1/S}, o_{\min} = -1, S = 3$$

在组  $o=-1$ , 图像用双线性插值扩大一倍 (对于扩大的图像  $\sigma_n=1$ )。

## 2、空间极值点的初步检测和精确定位

为了寻找尺度空间的极值点, 每一个采样点要和它所有的相邻点比较, 看其是否比它的图像域和尺度域的相邻点大或者小。如图 4.4 所示, 中间的检测点和它同尺度相邻的 8 个点以及上下相邻尺度对应的 18 个点, 总共 26 个点比较, 这是为了确保在尺度空间和二维图像空间都能检测到极值点。如果这个像素点是这 26 个像素中的最大值或者是最小值则初步认为这个像素点是特征点。

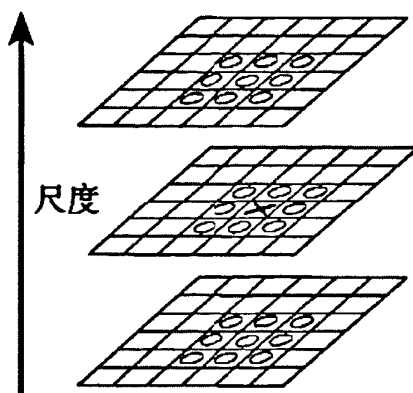


图 4.4 尺度空间上的极值点检测图示

因为 DOG 算子会产生较强的边缘响应，我们需要通过拟和三维二次函数以精确确定关键点的位置和尺度，同时去除对比度低的特征点和不稳定的边缘响应点，以增强匹配稳定性、提高抗噪声能力：

(1) 去除对比度低的特征点

对比度低的特征点是不稳定的。删除  $|D(\hat{x}) < 0.3|$  的特征点。其中

$$D(\hat{x}) = D(x_0) + \frac{1}{2} \frac{\partial D(x_0)^T}{\partial x} \hat{x} \quad (4-4)$$

(2) 去除不稳定边缘响应的特征点

仅删除低对比度的特征点是不够的，出于对稳定性的考量，还要删除一些不稳定边缘响应的特征点。这些特征点在边缘上定位不够精确，高斯差分算子的极值在沿着边缘的方向有较小的主曲率，而在垂直边缘的方向有较大的主曲率。主曲率通过一个  $2 \times 2$  的 Hessian 矩阵  $H$  求出：

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (4-5)$$

导数由采样点相邻差估计得到。H 矩阵的特征值和函数 D 的主曲率成正比，令  $\alpha$  为最大特征值， $\beta$  为最小的特征值，则

$$\text{Tr}(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (4-6)$$

$$\text{Det}(h) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \quad (4-7)$$

令  $\alpha = r\beta$ ，则：

$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \quad (4-8)$$

$(r+1)2/r$  的值在两个特征值相等的时候最小, 随着  $r$  的增大而增大, 因此, 为了检测主曲率是否在某域值  $r$  下, 只需检测  $\frac{Tr(H)^2}{Det(H)}$  是否大于  $\frac{(r+1)^2}{r}$  这里一般取  $r=10$ 。

### 3、特征点方向计算

为了使描述向量达到对旋转的不变性。需要给每个特征点计算一个方向, 这个方向是基于该特征点的局部图像信息的。描述向量的以这个方向作为基准表示, 从而达到旋转变换的不变性。

$$\begin{cases} m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \\ \theta(x, y) = \alpha \tan 2 \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \end{cases} \quad (4-9)$$

式(4-9)为  $(x, y)$  处梯度的模值和方向公式。其中  $L$  所用的尺度为每个关键点各自所在的尺度。

而我们在计算时, 是以关键点为中心的邻域窗口内采样, 用直方图统计出邻域像素的梯度方向。梯度直方图的范围是  $0 \sim 360$  度, 其中每  $10$  度一个柱, 总共  $36$  个柱。直方图的峰值则代表了该关键点处邻域梯度的主方向, 即作为该关键点的方向。图 4.5 是采用  $7$  个柱时使用梯度直方图为关键点确定主方向的示例。

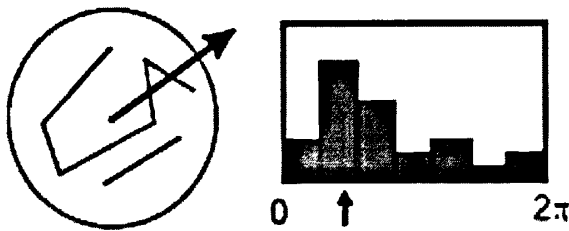


图 4.5 由梯度方向直方图确定主梯度方向

在梯度方向直方图中, 当存在另一个相当于主峰值  $80\%$  能量的峰值时, 则将这个方向认为是该关键点的辅方向。一个关键点可能会被指定具有多个方向 (一个主方向, 一个以上辅方向), 这可以增强匹配的鲁棒性。

至此, 图像的关键点已检测完毕, 每个关键点有三个信息: 位置、所处尺度、方向。由此可以确定一个 SIFT 特征区域 (用椭圆或箭头表示)。

### 4、特征点描述字生成

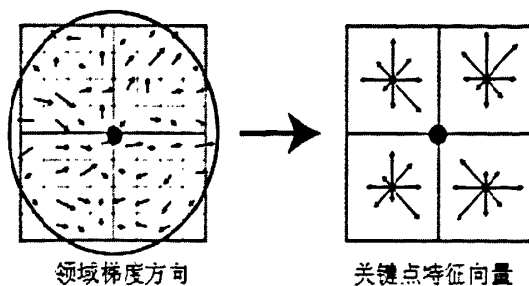


图4.6 由关键点邻域梯度信息生成特征向量

先将坐标轴旋转为关键点的方向，确保其旋转不变性。然后以关键点为中心取  $8 \times 8$  的窗口。图 4.6 左边部分的中央黑点为当前关键点的位置，每个小格代表关键点邻域所在尺度空间的一个像素，箭头方向代表该像素的梯度方向，箭头长度代表梯度模值，左图中蓝色的圈代表高斯加权的范围，越靠近关键点的像素梯度方向信息贡献越大。然后在每  $4 \times 4$  的小块上计算 8 个方向的梯度方向直方图，绘制每个梯度方向的累加值，即可形成一个种子点，如图 4.6 右部分所示。此图中一个关键点由  $2 \times 2$  共 4 个种子点组成，每个种子点有 8 个方向向量信息。这种邻域方向性信息联合的思想增强了算法抗噪声的能力，同时对于含有定位误差的特征匹配也提供了较好的容错性。

实际计算过程中，为了增强匹配的稳健性，Lowe 教授建议对每个关键点使用  $4 \times 4$  共 16 个种子点来描述，这样对于一个关键点就可以产生 128 个数据，即最终形成 128 维的 SIFT 特征向量。此时 SIFT 特征向量已经去除了尺度变化、旋转等几何变形因素的影响，再继续将特征向量的长度归一化，则可以进一步去除光照变化的影响。

## 5、特征点匹配

当两幅图像的 SIFT 特征向量生成后，下一步我们采用关键点特征向量的欧式距离来作为两幅图像中关键点的相似性判定度量。取前一帧中的某个关键点，并找出下一帧中欧式距离最近的前两个关键点，在这两个关键点中，如果最近的距离除以次近的距离少于某个比例阈值，则接受这一对匹配点。降低这个比例阈值，SIFT 匹配点数目会减少，但更加稳定。

但是经过匹配的点对一定是正确的吗？答案是不一定的。首先，背景高度相似的图像中特征向量的欧式距离相差不大容易产生错误的匹配。其次，我们这里图片中还存在着运动的目标，如果特征点取在了运动目标上，会产生错误的匹配结果。再次，如果两帧图片相差比较大提取的特征点对不足以描述背景的运动也是有可能的，由于 SIFT 特征点在纹理简单的图片中的提取数量明显小于复杂纹理场景中的数量，这造成了特征点分布的不均匀。这里我们采用 RANSAC 方法对特征点对的参数进行拟合，该方法通过迭代能有效的对存在大量错误匹配的点对



的参数进行拟合得到正确的结果，效果好于最小二乘法。

#### 4.4 仿射变换

平面内点之间的一个线性变换叫做仿射变换，如式子4.10所示，原图上的任意一点的坐标通过线性方程转化到新图上。我们通过图片配准算法最后拟合出的是一个仿射变换矩阵，用此矩阵来描述两张图片之间的运动关系，该模型能够很好的描述背景的运动。这里我们为什么会选用仿射变换来表示两帧图片的运动关系呢？

$$\begin{cases} x' = a_{11}x + a_{12}y + a_{13} \\ y' = a_{21}x + a_{22}y + a_{23} \end{cases} \quad \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \neq 0 \quad (4-10)$$

这是由于仿射变换的基本性质决定的<sup>[16]</sup>。

- (1) 同素性：仿射变换把平面上的点变为变换后平面上的点，把平面上的直线变成变换后平面上的直线。
- (2) 结合性：直线上的一点对应点是变换后仍是此直线上的点。
- (3) 不变性：共线三点变换后的单比保持一致。



(a) 无变焦 (b) 三倍变焦

图4.7镜头拉伸下的物体

由图4.7我们可以直观的理解图片中的仿射变换的三个基本性质，镜头变焦后直线AC还是直线AC，点B仍然在直线AC上，AB与BC的比例仍然保持不变。

根据仿射变换的基本性质，对于平面几何中的一些图形变换后的结果我们也了一定的认识：圆、三角形、正方形和等腰梯形，经过仿射变换后分别变为了椭圆、任意三角形、平行四边形和任意梯形；反之也是成立的。这里我们可以想像在摄像头不同角度位置下的相同背景也正是出现了上述的变化。所以摄像头运动下的目标和背景也可以认为是在做了上述的某种仿射变换后所呈现的视角。

## 第五章 基于 SIFT 补偿和粒子滤波的航拍目标跟踪

### 5.1 引言

从上述的讨论中我们可以看出,粒子滤波的步骤相对来说是比较清晰的。关键在于重要密度函数  $q(x_{0:k} | y_{1:k})$  的选择上,同时近年来的改进地方也还集中在了输入观测似然函数  $P(y_k | x'_k)$  和重采样的方式选择上。

根据我们已经分析过的航拍的特点和解决的方案,运动补偿首先要做的就是对图片的配准,这种方法让我们思考到的第一个解决方案就是帧差法输入检测,简单实用,配准后两帧相消就可以检测到目标。然后由第二章讨论的帧帧送检方式输入给粒子滤波,同时还可以在后续对粒子投放上增强精度从而得到更准确的目标出现概率密度分布。所以我们首先采取的是基于 SIFT 配准的帧帧目标检测输入,在观测函数上面使用欧式距离为基准的普通粒子加权和我们投放的粒子进行加权统计。但是出现的结果却不如人意,问题还是集中在检测输入的不准确性上,由于帧差法不对目标建模,输入粒子滤波的是简单的目标坐标,只能通过普通粒子加权给粒子,所以他的抗噪能力较弱所以导致检测输入的不精确。为此我们使用了基于颜色直方图模板的粒子滤波解决方案,利用目标颜色特征,对目标建立良好的模型,这样观测释然函数的输入变为了对模板巴氏距离的加权输入,而 SIFT 点的运动补偿主要集中在了前期对粒子投放的纠正上。得到的结果比上面的算法有所改进,尤其是在定位的精度和框选目标区域的增加上。但是随之而来的问题也是明显的,经过 SIFT 点计算和颜色模板巴氏距离的计算匹配,算法耗时太长。

对此作者最后提出了 SIFT 配准下基于中间帧估计的欧式加权粒子滤波,通过对帧差间目标和摄像头匀速运动的假设估计中间帧运动状态,此方法不仅计算量远小于基于颜色的粒子滤波,同时准确程度上要好于原来改进前的帧差法粒子滤波,抗干扰抗形变能力增强,定位精准。

### 5.2 SIFT 配准下基于高斯欧式距离加权的粒子滤波

视频数据中的运动目标检测方法一般分为背景差分法、光流法和帧间差分法。背景差分法为了适应背景的不断运动变化,需要不断地通过单模态或多模态背景模型算法(如高斯模型)进行背景模型的建立和更新,效率低,比较适合由固定相机获取的视频数据中的变化区域检测。光流法运算速度慢、耗时多,难以实时检测运动目标。

帧间差分法最为常用的运动目标检测和分割方法之一,基本原理就是在图像

序列相邻两帧或三帧间采用基于像素的时间差分通过闭值化来提取出图像中的运动区域。首先, 将相邻帧图像对应像素值相减得到差分图像, 然后对差分图像二值化, 在环境亮度变化不大的情况下, 如果对应像素值变化小于事先确定的阈值时, 可以认为此处为背景像素; 如果图像区域的像素值变化很大, 可以认为这是由于图像中运动物体引起的, 将这些区域标记为前景像素, 利用标记的像素区域可以确定运动目标在图像中的位置。由于相邻两帧间的时间间隔非常短, 用前一帧图像作为当前帧的背景模型具有较好的实时性, 其背景不积累, 且更新速度快、算法简单、计算量小。算法的不足在于对环境噪声较为敏感, 阈值的选择相当关键, 选择过低不足以抑制图像中的噪声, 过高则忽略了图像中有用的变化。对于比较大的、颜色一致的运动目标, 有可能在目标内部产生空洞, 无法完整地提取运动目标。

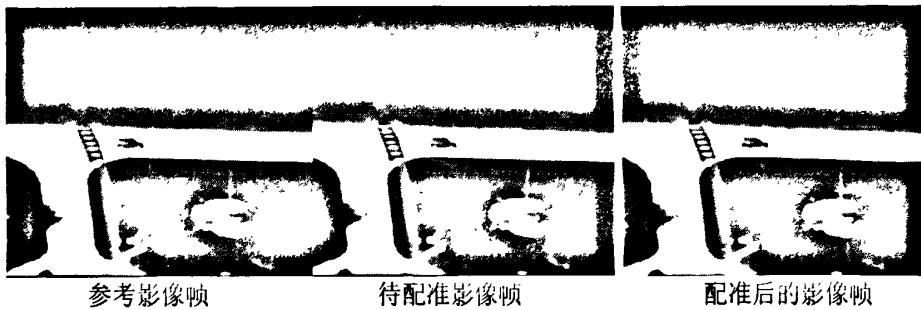


图 5.1 采用 SIFT 点的图片配准

两帧配准后相减的结果在理论上设置一个正数的阈值可以得到一个目标车身, 但由于目标形状的不规则和像素级的干扰太大, 无论怎样设置拟合和阈值参数检测结果始终不理想。在图 5.2 中我们看到的是帧差后经过滤波拟合二值化的图片。



图 5.2 帧差法下的检测结果

上图中的切影是我们设置的一个正数的阈值所得到的一个车位的位置, 此车是基准帧中的目标, 也就是送检的首帧, 一次送检的共有首尾两帧, 每帧的间隔数目要根据目标的运动强度来决定, 一般是 3 到 5 帧, 过小的话不容易检出车身, 过大会产生空洞。

在对粒子进行投放扩散加重的过程中我们依据的是卡尔曼的运动模型, SIFT 点的补偿在这里也加入了进去, 目的是让粒子投放扩散更加的精确。对粒子加权我们采用的是基于欧式距离 (Euclidean distance) 下的高斯加权, 欧式距离也称欧

几里得距离, 他是在  $N$  维空间两点的真实距离。由于检测值的  $X$  和  $Y$  坐标的关联性不是太大, 这里我们可以以每个粒子  $X$  轴坐标和  $Y$  轴坐标分别与基准观测值的一维欧式距离下的高斯加权。当然我们也可以以二维的欧式距离下的高斯加权对粒子进行加权。

每个粒子一维  $X$ ,  $Y$  轴下对应的加权为:

$$\begin{cases} Xweight = \frac{e^{-(x-x_0)^2/2}}{\sqrt{2\pi}} \\ Yweight = \frac{e^{-(y-y_0)^2/2}}{\sqrt{2\pi}} \end{cases} \quad (5-1)$$

每个粒子二维下对应的加权为:

$$Dweight = \frac{e^{-((y-y_0)^2+(x-x_0)^2)/2}}{\sqrt{2\pi}} \quad (5-2)$$

上两式子中  $(x, y)$  为每个粒子的坐标,  $(x_0, y_0)$  为基准观测下的目标坐标。这里的方差我们取值为 1。

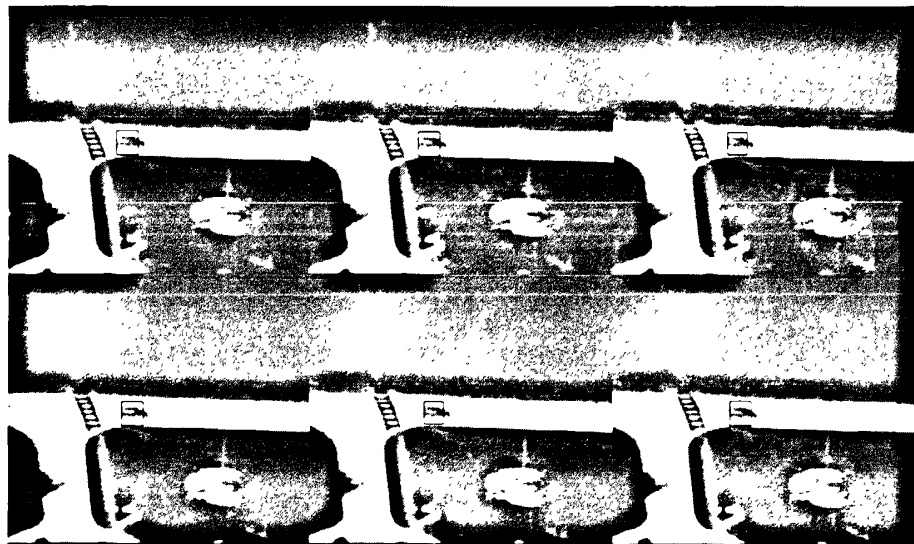


图 5.3 SIFT 配准下基于高斯欧式距离加权的粒子滤波

图 5.3 是我们采用此方法得到的最终试验结果, 从以上 6 帧图中看出该车的质点定位明显靠后。这是由于两车重合区域在相减时会出现正负交替状况, 光靠阈值的设定是不能有效解决的。

### 5.3 SIFT 配准下基于巴氏距离加权的粒子滤波

上一节中算法没有考虑到目标的特征信息, 只是根据航拍的解决方案运动补偿对图片进行配准相消, 从而选取的帧差和标准粒子滤波的结合。所产生的问题让我们回到第二章去进行一系列的思考, 我们上一个解决方案实际上是将目标认为是点来解决的, 而实际情况却不是这样的。这里的目标是一个具有稳定颜色分

布的目标区域，所以我们这里可以充分利用目标的颜色信息来进行跟踪，对目标建立颜色直方图模板是一个不错的方案，在和粒子滤波结合点也在于观测释然函数上，和帧差法帧帧送检的方式不同，这里我们只需要首帧手标出目标区域或者用自检测程序自动检测出首帧目标就可以了，算法运行后我们所做的是一种目标周围的模板匹配搜索，从而对粒子进行加权得到每一帧的目标概率密度分布。

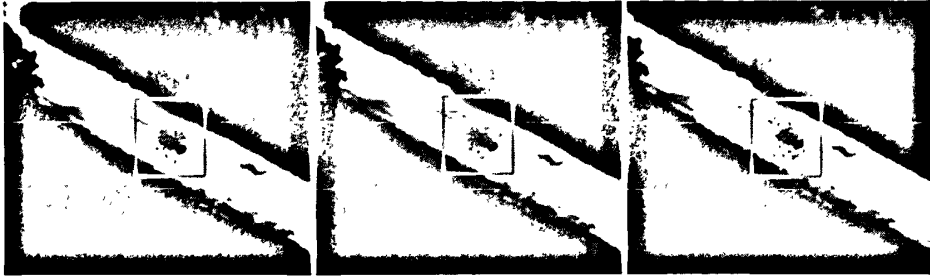


图 5.4 基于巴氏距离加权的粒子投放

图 5.4 中我们可以看到粒子的投放大部分都是在框中的，这和我们以前讨论的粒子全局投放方式看似不一致，但是一般粒子投放遵循的是高斯分布，总是集中在目标周围的，帧间粒子投放的变化在图片中显示的还是比较明显的。如下图 5.5 的标准高斯分布所示，在 $[-3,3]$ 这个区间粒子的投放是最多的，如果超过这个区域，投放的概率几乎趋近于零。

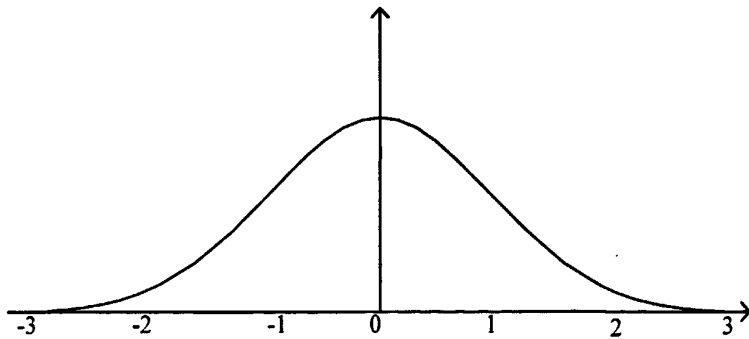


图 5.5 标准高斯分布

这里对颜色的利用是根据目标区域的颜色直方图来的，由第二章我们对颜色特征的分析，目前主要的颜色模型有 RGB 模型、HSB 模型等。这里我们选取 RGB 模型，其中一方面是因为他比较直观，另外一方面场景中的颜色只有灰度这个单色，我们采用 8 位灰度值的时候 RGB 建立模型中三色的数值都是一致的，当  $R=0, G=0, B=0$  的时候颜色是全黑的，而当  $R=255, G=255, B=255$  的时候颜色是全白的。而粒子集所代表的状态就必有他所代表的这个区域的各种参数信息，这里需要说明的是粒子所代表的是以该粒子坐标为中心所建立起来的模板框长宽的颜色直方图信息。由于颜色模板上维数过多，欧式距离已经不适合颜色模板间的距离测度，

我们采用了巴氏距离<sup>[17, 31]</sup> (Bhattacharyya 系数), 给粒子代表的估计模板和参考模板进行匹配测度。

$$p_y^u = \frac{1}{N} \sum_{i=1}^N \|y - x_i\| \delta[h(x_i) - u] \quad (5-3)$$

给目标区域建立的颜色模型是  $p_y = \{p_y^u\}_{u=1, \dots, m}$ , 其中  $N$  表示目标区域内的像素个数;  $x_i$  为目标区域中的点;  $h(x_i)$  将点的颜色分配给颜色直方图中的相应的条柱;  $\delta$  是 Kronecker delta 函数。其中颜色空间的划分: RGB, HSB 及其中各个分量的离散度都需要根据任务的需要选择。这里我们定义的巴氏系数是:

$$\rho[p, q] = \sum_{u=1}^m \sqrt{p^u q^u} \quad (5-4)$$

其中  $p$  和  $q$  都待匹配的颜色模型, 其数值越大说明相似度越高。两个完全一样的分布的巴氏系数为 1。

巴氏距离为:  $d = \sqrt{1 - \rho[p, q]}$ , 与巴氏系数刚好相反, 是两目标模板的量测距离, 即粒子的加权数值。d 越大则越不匹配, d 越小则越相似权值就越高。

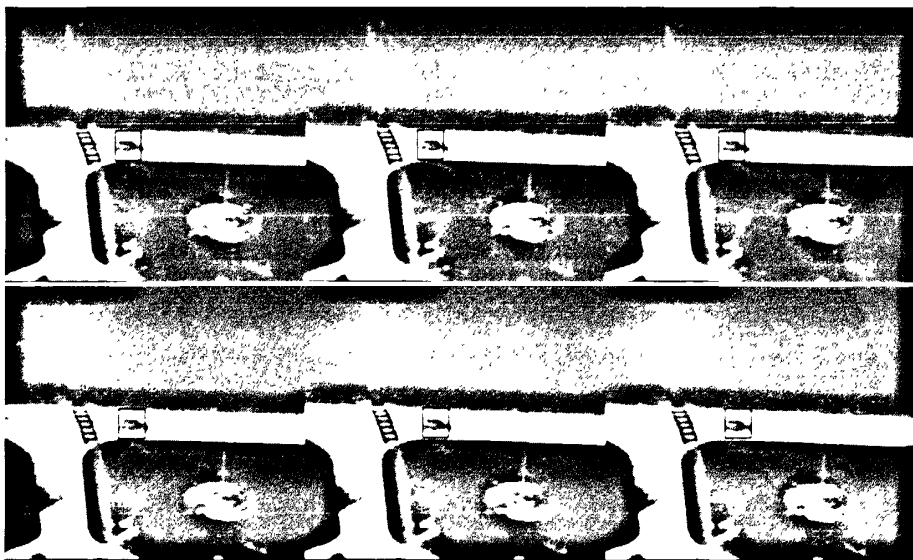


图 5.6 SIFT 配准下基于巴氏距离加权的粒子滤波

这里的 SIFT 点图片配准所提取的图片运动参数即仿射矩阵, 主要是用在了粒子下一帧的投放和扩散上, 粒子扩散的速度是车体运动速度和摄像头运动的合成。投放也可以理解为粒子的扩散的前序。每一帧的粒子都要以上一帧的目标位置为基准进行再次投放, 同时粒子的扩散是以上一帧的车速和摄像头速度的合成为基础进行的预测, 当预测完毕后就形成了多个粒子模板, 后续对其和候选模板进行巴氏距离测度加权形成新的概率密度。这里的模板是上一帧的目标, 所以这就是不需要帧帧进行目标检测的依据 (对区域目标进行跟踪大多是不需要帧帧做目标

检测的)。然后从概率密度中提取出最新位置来更新此模板, 算法可以得以继续迭代。图 5.6 所示中大部分还是得到了比较精确的跟踪效果的, 在第二帧到第四帧由于汽车的加速扰动出现了些许偏差, 不过后续还是平稳的跟踪上了。

在此算法下运行时间明显比第一种增加, 以下是我们在同配置 PC (CPU: Core(TM)6300 1.86GHz/ 0; 内存: 1.86G/ 269M; 操作系统: WindowsXP-SP3; Matlab7.6.0) 下, 对相同场景进行跟踪的时间对比:

表 5.7 两种算法耗时对比

实验帧数 粒子算法选择	10 帧	50 帧	100 帧
SIFT 下的欧式距离	98.42s	489.89s	993.31s
SIFT 下的巴氏距离	322.99s	1189.55s	2417.51s

由上表明显看出耗时相差巨大, 超过 2 倍。SIFT 点计算复杂的耗时问题在特征点中已经属于很高级别, 再加上图片颜色建模和巴氏距离的话系统负担远远超出想象。对此本文作者提出了改进算法。

#### 5.4 基于中间帧估计的粒子滤波

通过对上述两种跟踪算法的了解和比较, 我们既想要得到精确的目标定位, 又想有效的缩短算法时间, 这是一对矛盾。SIFT 点作为特征点的一种我们已经在第四章分析了他的优势是定位精确但是耗时太重, 而目前我们由上两个耗时结果来看主要时间损耗是集中在了目标颜色建模和巴氏距离加权上。所以如果能将巴士距离用欧式距离代替将能成 2 倍缩短耗时, 但是不对目标建立颜色模型不利用颜色信息, 只用帧差检测目标会产生严重的目标偏移, 从而使欧式距离加权出现严重错误, 导致跟踪精度降低。

由第二章我们知道检测和跟踪的区别和联系, 检测一般是单帧来做的, 虽然帧差法是两帧来做, 但他没有有效的利用帧间目标的运动信息。作者从此处出发思考是否能利用帧间信息来加强对帧差检测的定位精度, 从而提高粒子滤波中粒子加权的准确度。

通过分析帧间的目标运动, 我们可以假设两帧甚至几帧间的目标作的是一个匀速运动, 这是完全可以理解的, 当前的视频采样率也充分保证了这样的假设的可行性。同理摄像头在帧间的运动我们也可以短暂的假设为是匀速的。有了以上两个假设, 我们提出了自己的改进算法。通过在间隔偶数帧的两帧送检图片中进行帧差, 阈值取其绝对值裁剪出具有重叠车身的两个目标车位, 将中心点赋值给

中间那帧图片的目标位置，由于此时 SIFT 做的图片配准是以第一帧为基础的，忽略了中间帧上的摄像头的运动，所以还需要对中心点进行再次纠正赋值给中间帧，同时在对粒子的投放和扩散上也要进行必要的修改，因为此时我们不是在对相邻两帧间进行投放了，跨越几帧的投放需要我们进行估计投放，同时扩散的速度是一个合成的速度。

这里我们通过三个步骤对算法进行改进，最后实验我们得到了很好的改进效果，同时耗时大大低于基于颜色的粒子滤波并与高斯欧式加权的粒子滤波接近。

算法具体修改步骤：

### 1、检测阈值修改

修改背景相减的阈值得到重影的两个目标车位（加强抗干扰能力，提高小目标检测率），此时假设目标在第一帧到第五帧间作匀速运动，提取两车位中心点作为中间第三帧目标位置。

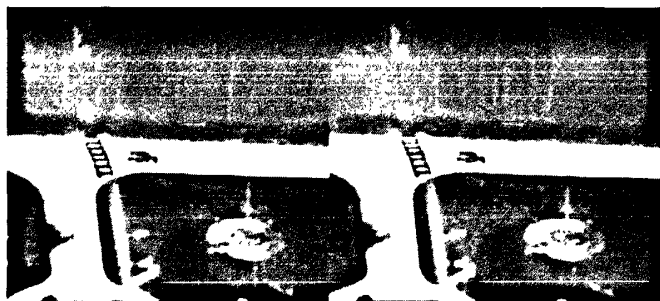


图 5.8 等待匹配后帧差的第一帧和第五帧

配准后两帧相减在理论上设置一个正数阈值可以得到一个目标车身，但是由于目标形状不规则和像素级的干扰，检测结果不理想，这个已经在 5.1 节做过介绍，这里不再多做介绍。

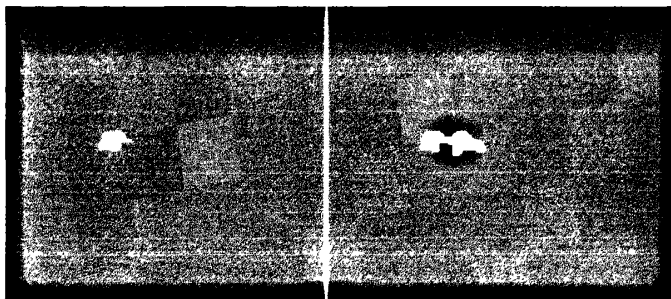


图 5.9 第一帧和第五帧设置正数阈值下和正负阈值下的帧差结果

这里我们得到的中心点的坐标为  $[x_3 \quad y_3]$ ，这里检测到的并不是第三帧图片上的目标坐标，而是以第一帧图片为背景下的目标坐标。所以需要进一步的修正。

### 2、检测坐标的修正

此时虽然得到了第三帧的目标坐标，由于是以第一帧为基础的配准，第三帧的目标坐标忽略了前两帧摄像头的运动，这里我们可以有两种纠正方式：

(1) 直接对第一帧和第三帧再做一次配准得到摄像头在这两帧的运动模型，



然后修正由步骤 1 得到的坐标;

(2) 假设摄像头在第一帧和第五帧间作匀速运动, 可以由前期得到的这两帧的运动模型来修正由步骤 1 得到的坐标, 这就大大节省了算法运行时间。

通过分析程序不难发现算法时间主要消耗在了 SIFT 点配准上, 如果按照策略

(1) 修正的话时间将会成倍增加, 这和我们前期所要达到的目的是不符合的, 所以这里我们选择的是策略 (2) 进行实验:

我们以第一帧和第五帧为例, 以下是第五帧到第一帧的仿射变换矩阵:

$$Modle = \begin{bmatrix} 0.99877 & -0.0024 & 0 \\ 0.0089 & 0.9993 & 0 \\ 4.9323 & 0.6086 & 1 \end{bmatrix} \quad (5-5)$$

因为此时我们要修正第三帧就必须得到摄像头前五帧间的运动信息, 假设第一帧目标不动摄像头运动的条件下, 我们提取出摄像头的运动, 得到以第五帧为背景目标的位置:

$$\begin{bmatrix} x_5^* & y_5^* & 1 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix} \times Modle^{-1} \quad (5-6)$$

策略 (2) 下的修正第三帧的检测坐标为:

$$\begin{bmatrix} x_3 & y_3 \end{bmatrix} = \begin{bmatrix} x_3^* & y_3^* \end{bmatrix} - \frac{\begin{bmatrix} x_1 & y_1 \end{bmatrix} - \begin{bmatrix} x_5^* & y_5^* \end{bmatrix}}{2} \quad (5-7)$$

此时的  $\begin{bmatrix} x_3 & y_3 \end{bmatrix}$  才是真正以第三帧为背景下检测到的坐标, 是在我们做出了目标和摄像头帧间短暂匀速运动的假设后得到的估计, 同时也是两种运动的合成。

### 3、粒子投放和扩散的修正

由第三章对粒子滤波的分析我们可以发现, 和加权一样, 投放和扩散对粒子是很重要的。投放的精度和扩散的速度会对概率密度估计产生重要的影响。这里我们如果采用了上两个步骤, 那对粒子投放和扩散都是需要修正的:

#### (1) 投放

这里我们首先从首帧启动来分析, 首帧我们可以手标也可以采用自检测方式自动标记。第一帧的位置有了以后, 我们经过步骤 1 和 2 可以得到修正后的第三帧坐标, 但是粒子滤波告诉我们必须用第二帧的状态和第三帧的检测去估计第三帧的状态, 此时的投放也必须是要以第二帧的目标状态去投放的。这里我们还是可以回到第二个步骤由第一帧和第三帧间的匀速假设去求得第二帧的目标坐标, 所不同的是我们现在没有第二帧的初次检测, 所以这里估计第二帧的坐标是由第一帧加上摄像头运动和目标运动合成而来的:

$$[x_2 \ y_2] = [x_1 \ y_1] - \frac{[x_1 \ y_1] - [x_5^* \ y_5^*]}{4} + \frac{[x_3^* \ y_3^*] - [x_1 \ y_1]}{2} \quad (5-8)$$

在首帧过后我们以后要求的第四帧状态由于已经有了上一帧第三帧状态为基础了所以就可以不用重复上述步骤了，直接以第三帧位置状态投放到第四帧上。这里还有第二种策略可以省略第三帧跟踪直接从第四帧做起，这时有相邻第三帧的积累就可以直接启动粒子滤波了。

## (2) 扩散

这里需要说明的是如果载具不运动的情况下，我们所说的扩散速度都是指目标的运动速度，而此时载具的运动使得我们必须将其运动加入，所以此情况下的扩散速度是一个合成的速度。这一点是十分重要的，也是运动补偿加入粒子滤波最重要的关键点。

$$\text{Speed} = \text{speedobject} + \text{speedcamera}$$

在上两节介绍过的算法中都加入了这种补偿，但是这时加入补偿的方式却和上两种算法是不一样的，因为此时求得的是跨越好几帧的运动模型。但是经过上面的分析，我们依然可以用匀速假设去衡量中间帧的运动，无论是目标的还是载具的。由步骤 2 和上一个定位的积累我们可以将 `speedobject` 和 `speedcamera` 都提取出来进行相加最终得到合成速度 `Speed` 去进行粒子扩散。

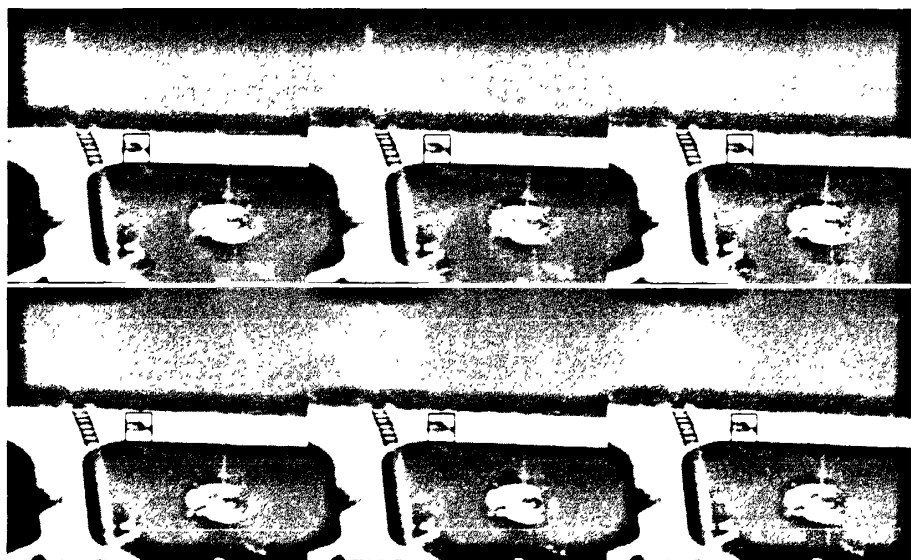


图 5.10 SIFT 配准下基于中间帧估计的高斯加权粒子滤波

这里我们仅以第三帧为例对第三帧的粒子合成扩散速度 `speed` 进行阐述：

$$speed_{objec\ t} = \frac{[x_3^* \ y_3^*] - [x_1 \ y_1]}{2} \quad (5-9)$$

$$speed_{camera} = \frac{[x_1 \ y_1] - [x_5^* \ y_5^*]}{4} \quad (5-10)$$

$$speed = \frac{[x_3^* \ y_3^*] - [x_1 \ y_1]}{2} - \frac{[x_1 \ y_1] - [x_5^* \ y_5^*]}{4} \quad (5-11)$$

这是我们最后得到的第三帧粒子扩散速度  $speed$ ，是一个合成速度。到这里我们的改进结束。之后采用高斯欧式距离对粒子进行加权产生本帧概率分布，统计出第三帧目标状态并开始下一轮算法迭代。

这里我们是在基于卡内基梅隆大学的数据做的实验，主要是针对的目标是车辆，帧数为 1400 帧。图 5.10 给出了改进后的结果，图中目标的精度明显好于图 5.3 和图 5.6 的结果。为了容易辨识，我们这里的图 5.3，图 5.6 和图 5.10 都是在针对前八帧图片在做，这在后面几百帧中也能很明显的辨识。下面我们用对三种算法的耗时效率进行比较。

表 5-12 三种算法同帧下耗时比较

实验帧数 粒子算法选择	10 帧	50 帧	100 帧
SIFT 下的欧式距离	98.42s	489.89s	993.31s
SIFT 下的巴氏距离	322.99s	1189.55s	2417.51s
中间帧估计法	114.44s	563.77s	1133.92s

表 5-12 中我们对比发现本文提出的中间帧估计算法的效率接近于三者中的最优效率，大大小于基于颜色巴氏距离加权的粒子滤波耗时，接近基于欧式距离加权的粒子滤波。

实验表明基于中间帧估计的航拍解决方案将前两种算法的优势结合，解决了两者的不足之处，得到了比较好的实验效果。

## 第六章 总结与展望

航拍目标跟踪受到越来越多研究者的关注,不仅因为其在军事应用中的卓越贡献,同时民用安保产业的发展也催生了大量的需求。从无人机到空基交通,从精确打击到载具自动避障碍,现实生活中其应用的广度和深度正在逐步扩展和加强。本文首先从目标跟踪的角度入手,剖析大致原理,然后引入具体问题矛盾对航拍的特殊性进行分析,最后提出解决方案并加以改进。

目标跟踪和检测是相辅相成的,有共同点也有不同。检测时区别出单帧图片上目标和背景,而跟踪是要填补帧间目标运动的空白从而找寻帧间目标间的一致性将其关联起来。只有对其关系有深入的认识才能在后续的算法选择和设计上结合场景需要取得很好的效果。

航拍中最大的特点是载具的运动带来的干扰,解决的方法要联系跟踪算法的扰动点来提出。消除和抵消摄像头运动是一个不错的方案,这里我们采用的方法是用图片配准找出两帧图间的背景运动即摄像头运动。在加入以贝叶斯为准则的粒子滤波算法中,我们要对其粒子的加权和扩散进行适当的纠正才能达到去除载具扰动的效果。通过分析高斯欧式距离和巴氏距离加权的优缺点后提出一种航拍目标跟踪的解决方案。此方法是在 SIFT 点配准后利用对帧间目标和摄像头短暂匀速的假设,用帧差法检测估计中间帧目标位置送检给粒子滤波,同时对粒子的投放和扩散进行纠正的航拍跟踪算法。算法鲁棒性好同时效率高,取得了良好的实验结果。

展望后续研究,可以分为两个方向:一、我们可以对多目标进行跟踪的时候将此算法用不同的粒子滤波器进行关联,可以实现帧帧检测目标的模式。二、将高斯欧式距离和巴氏距离适当的结合起来,提高其检测精度同时降低耗时。



## 致 谢

值此论文成稿之际，我想首先感谢我的导师田捷教授！自我归入田老师门下，他严谨的治学态度就深刻影响了我。在我三年的学习过程中，田老师在各个方面给予了大力的支持和帮助。从做学问的精益求精到做人的正直无私上，无一不给我树立了学习的榜样。他渊博的知识和敏锐的科学洞察力成为我不断进步的动力。他平易近人的工作作风以及严以律己、宽以待人的高尚品格，更为我的成长指明了航向。借此机会，向田老师的苦心栽培致以深深的谢意！

感谢智能生物信息中心主任梁继民老师，赵恒老师、胡海虹老师、侯彦宾老师，张毅老师，秦伟老师，他们给予了我很多的帮助和指导，使我受益匪浅；他们的辛勤工作，为我们创造了良好的实验室条件。

感谢教研室的诸位同窗，与我一起度过这段难忘的时光，他们是宋健强、孟庆虎、汪小龙、唐余波、王倩、潘静、高海鹏、王静、邹亚如等。与他们的友好交往增添了我学习的乐趣，有益的学术讨论也使我开阔了思路，获益匪浅。感谢他们对我的关心，感谢他们的友好建议。

感谢我的家人，在我遇到困难想要退缩的时候，他们是最坚强的后盾，让我能够坚持下去，义无反顾。

向在过去所有的日子里，关心、爱护和帮助我的人们致以衷心的感谢！

最后感谢参加论文评审和答辩的各位老师对本文的认真审阅。



## 参考文献

- [1] Medioni G, Cohen I, Bremond F, Honggeng S, and Nevatia R. Event Detection and Analysis from Video Streams. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2001, vol.23, pp.873-889, August.
- [2] <http://baike.baidu.com/view/120502.htm#4>
- [3] Yilmaz, A., Javed,O., and Shah, M. 2006. Object tracking: A survey. *ACM Comput. Surv.*38,4, Article13(Dec.2006)
- [4] Black M.AND Jepson, A. 1998. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int.J.Comput.Vision*26, 1, 63-84.
- [5] COMANICIU,D.,RAMESH,V.,and MEER,P.2003.Kernel-basedobject tracking. *IEEETrans.Patt.Analy.Mach.Intell.*25, 564-575.
- [6] TEKALPAM. *Digital video processing [M]*. PrenticeHall,1995
- [7] Lucas-Kanade 20 Years On: A Unifying Framework S. Baker and I. Matthews *International Journal of Computer Vision*, Vol. 56, No. 3, March, 2004, pp. 221 - 255
- [8] Lowe D. Distinctive Image Features from Scale-Invariant Key points. *International Journal of Computer Vision*, 2004, pp.91-110.
- [9] Ke Yan, Sukthankar T. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.
- [10] David G Lowe. Distinctive Image Features from Scale-Invariant Key points[J]. *International Journal of Computer Vision*, 2004.
- [11] Lowe DG. Object recognition from local scale invariant features[C]. *Proceedings of the Seventh International Conference on Computer Vision(ICCV'99)*,1999.
- [12] Fischler M, Bolles R. Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography. *Communications of ACM*, 1981, pp.726 - 740.
- [13] HARITAOGLU I. HARWOOD DAVID. DAVIS L S Real-time surveillance of people and their activities 2000 (08)
- [14] Martin A,Fischler, and Bolles R C, Randome Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated



- Cartography, Communications of ACM 1981.
- [15] Bolles B, and Fischler M. RAMSAC: An Historical Perspective, presentation on RANSAC 25<sup>th</sup>, 2006, June.
- [16] 梅向明等. 高等几何<第二版> [M].北京: 高等教育出版社, 2000.
- [17] Comaniciu D, Ramesh V, and Meer P. Real-Time Tracking of Non-Rigid Objects Using Mean Shift. Proc. Conference on Computer Vision and Pattern Recognition, 2000, vol.2, pp.142-149.
- [18] Kalman G E. A New Approach to Linear Filtering and Prediction Problems. Trans. of the ASME-Journal of Basic Engineering, 1960, pp.35-45.
- [19] Welch G, and Bishop G. An Introduction to the Kalman Filter. 2004 April
- [20] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 50, NO. 2, FEBRUARY 2002
- [21] Y. C. Ho and R. C. K. Lee, A Bayesian approach to problems in stochastic estimation and control. IEEE Trans. Automat. Contr., vol.AC-9, pp.333-339, 1964.
- [22] Anderson C., Bert P., Vander Wal G. Change detection and tracking using pyramids transformation techniques. In: Proceedings SPIE Conference on Intelligent Robots and Computer Vision, Cambridge, MA, 1985, 579:72-78.
- [23] Nummiaro K, Koller-Meier E, Van Gool L. An Adaptive Color Based Particle Filter, Image and Vision Computing, 2003. pp.353-360.
- [24] Kailath T. The Divergence and Bhattacharyya Distance Measures in Signal Selection. IEEE Trans. Comm. Technology, 1967, vol.15 pp.52-60.
- [25] 曲双石 王会娟 Monte Carlo方法及其应用 Statistical Think tank Series No.112 2009
- [26] MATTHIES, L., SZELISKI, R., AND KANADE, T. 1989. Kalman filter based algorithms for estimating depth from image sequences. Int. J. Comput. Vision 3, 3, 209-238.
- [27] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Transactions on Signal Processing, 2002, vol.50, pp.174-188, February.
- [28] Yang Changjiang, Duraiswami R, and Davis L, Fast Multiple Object Tracking via a Hierarchical Particle Filter. International Conference on Computer Vision, 2005, vol.1, pp.212-219, October.
- [29] 陆光华, 彭学愚, 张林让, 毛用才, 随机信号处理, 西安电子科技大学出版

- 社, 2002.
- [30] Comaniciu D, Ramesh V, and Meer P. Kernel-based Object Tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2003, vol. 25, pp. 564-577.
  - [31] Kailath T. The Divergence and Bhattacharyya Distance Measures in Signal Selection. *IEEE Trans. Comm. Technology*, 1967, vol.15 pp.52-60.
  - [32] Collins R. T., Lipton A. J., Kanade T.. Introduction to the Special Section on Video Surveillance. *IEEE Transactions. Pattern Analysis and Machine Intelligence*. 2000. Vol. 22, No. 8, Pp. 745-746.
  - [33] Perez P, Hue C, Vermaak J, and Gangnet M. Color-Based Probabilistic Tracking. *European Conference on Computer Vision*, 2002, pp.661-675.
  - [34] Pless R, Wright J. Analysis of Persistent Motion Patterns using the 3d Structure Tensor. 2005, vol2, pp.14-19.
  - [35] Pless R. Spatio-temporal Background Models for Outdoor Surveillance. *Eurasip Journal on Applied Signal Processing*, 2005, pp.2281-229.
  - [36] Krumm J, Toyama K, Brumitt B, and Meyers B. The Ultimate Futility of Background Subtraction. Submitted to *International Journal of Computer Vision*.

