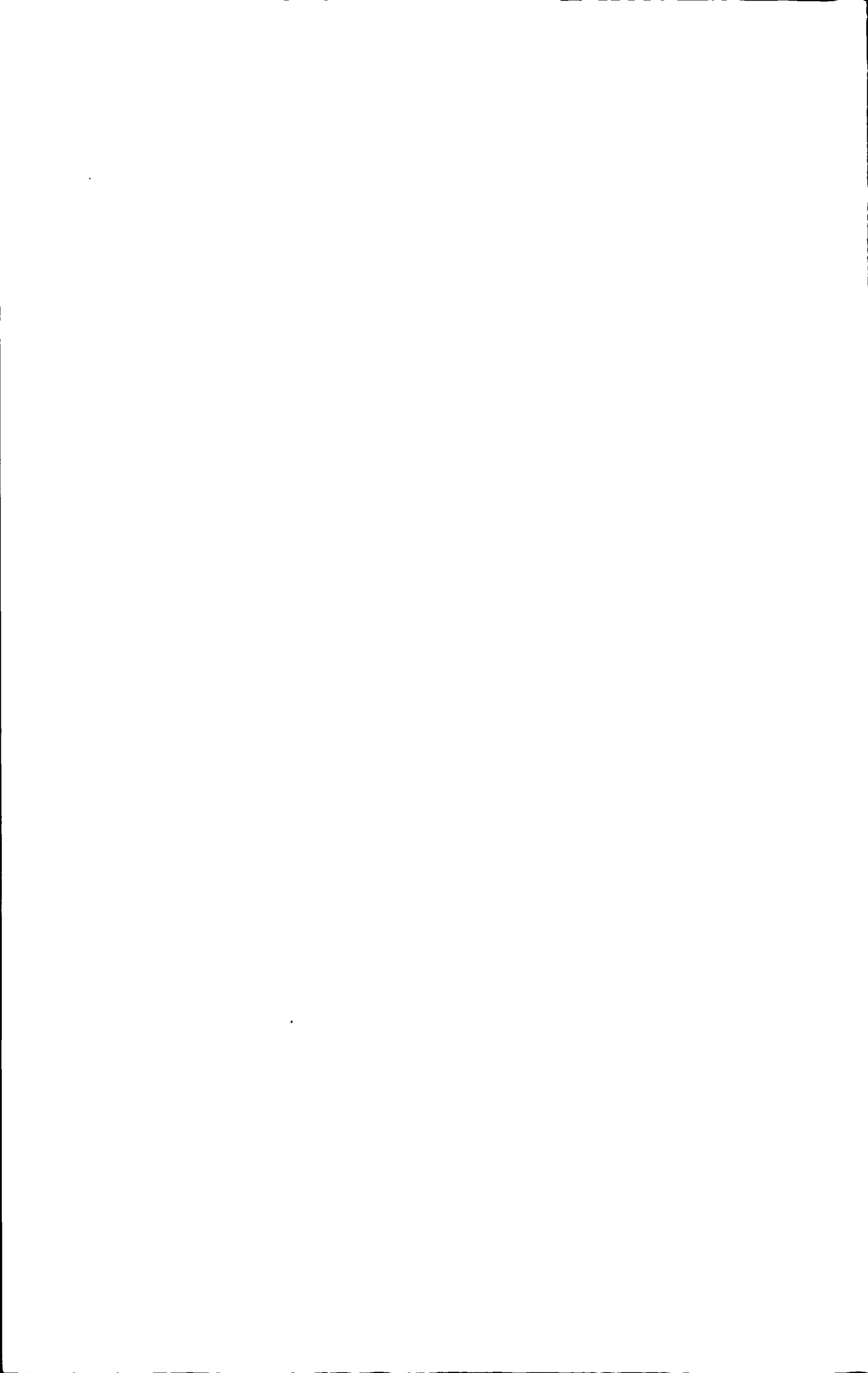


## 摘 要

人脸跟踪是计算机视觉领域中一个非常重要的研究课题,其研究成果可以应用于民用和军事的很多领域,在人脸识别、智能监控、人机交互等领域有着较高的经济价值和广泛的应用前景,并因此受到研究者们越来越广泛的重视。本文针对视频图像序列中场景和人脸目标复杂变化的情况,对 Mean Shift 跟踪算法进行了研究。

在人脸跟踪领域中,Mean Shift 算法是一种优秀的跟踪方法,它可以快速且有效的实现视频序列中相邻两帧图像之间运动目标的匹配。但是该算法同时具有目标模板无法更新、应用特征单一和核窗宽大小固定不变的缺点,在背景复杂变化、目标姿态及大小变化和外界扰动的情况下很容易丢失目标。本文针对 Mean Shift 这三方面的缺陷分别作了改进,首先提出了一种自适应模板更新的策略,在基本 Mean Shift 算法的基础上,利用人脸姿态变化信息和前一帧的目标模型实现了在线实时更新目标参考模型。随后引入了纹理和颜色结合的多特征目标模型表示方法,利用纹理和颜色信息共同决定目标模型,解决了应用特征单一的缺点,达到更为鲁棒地跟踪人脸的目的。最后针对核窗宽大小固定的缺点,加入了椭圆拟合方法来自适应调节核窗宽。本文利用 Viper 评估软件对以上改进方法进行评估,评估结果显示新算法能够有效的跟踪目标,且对于目标快速运动、目标旋转等姿态变化和大小变化问题都能很好的处理,获得可靠的跟踪效果。

关键词:人脸跟踪 Mean Shift 算法 自适应模板更新 Viper



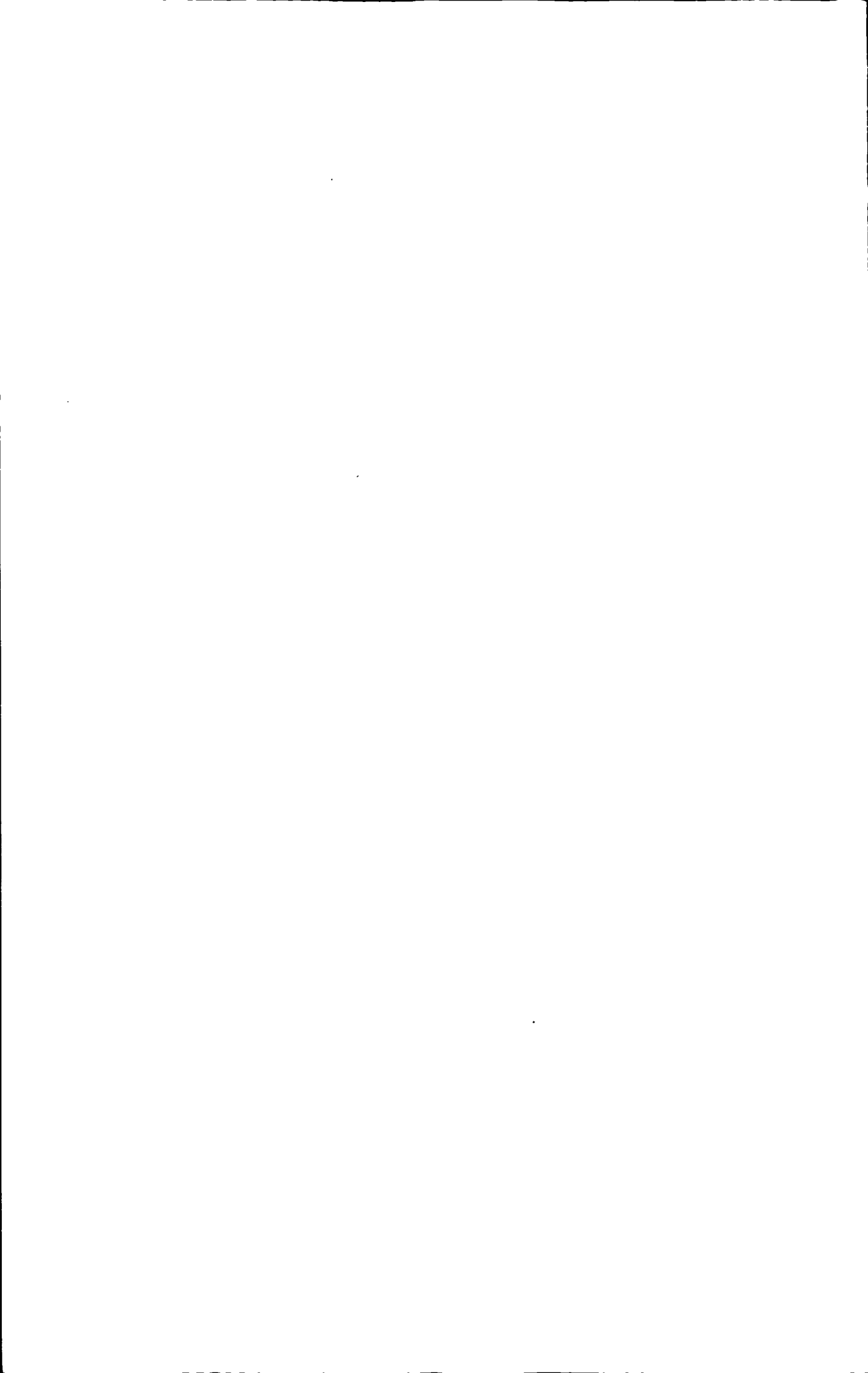
## Abstract

Face tracking is an important research subject in computer vision field, and its research results has high economic value and broad application prospects in face recognition, smart surveillance, generation computer interfaces and other many areas of civil and military, so it has been paid more and more concerns. This paper studies the face tracking algorithm based on Mean Shift to aim at the complex changes of the movements and scene in the video.

In object tracking area, Mean Shift algorithm is a good tracking method, it can deal with the object matching problem between two successive frames fast and effectively. However it has the drawbacks: templates can not to be updated, a single feature is used, and the bandwidth is fixed. All these make object easily lost when the backgrounds vary, the gesture and sizes of the object change and the object is under the external disturb.

This paper improves the Mean Shift algorithm by using adapted templates updated, introducing multi-features object model representation with color and texture, using ellipse fitting method to update tracking bandwidth. Viper, a performance evaluation platform for object detection and tracking, is used to evaluate the above improved methods, the results showed the new approach tracks well, and yields satisfying results even under the complicated object motion.

**Keywords:** Face tracking Mean Shift Adapted templates updated Viper



## 目 录

第一章 绪 论.....	1
1.1 人脸跟踪概述.....	1
1.2 相关研究工作.....	2
1.2.1 视觉跟踪线索.....	3
1.2.2 基本跟踪算法.....	5
1.3 人脸跟踪面临的困难和挑战.....	7
1.4 本论文主要工作和内容安排.....	7
第二章 基于自适应模板更新的 Mean Shift 算法.....	9
2.1 引 言.....	9
2.2 Mean Shift 理论.....	10
2.2.1 核概率密度估计.....	10
2.2.2 密度梯度估计.....	13
2.3 基于颜色直方图的 Mean Shift 跟踪算法.....	15
2.3.1 目标参考模型的表示.....	16
2.3.2 候选目标的表示.....	17
2.3.3 基于相似性函数的测度.....	17
2.3.4 目标定位.....	18
2.4 自适应更新目标模板的跟踪算法.....	19
2.4.1 固定的目标模板.....	19
2.4.2 自适应更新目标模板.....	19
2.4.3 改进后的 Mean Shift 算法步骤.....	21
2.5 实验结果与分析.....	22
2.6 本章小结.....	23
第三章 颜色-纹理结合的 Mean Shift 算法.....	25
3.1 引 言.....	25
3.2 改进的 LBP 纹理模型.....	26
3.3 颜色-纹理结合的目标模型表示.....	29
3.3.1 加权融合.....	29
3.3.2 乘性融合.....	29
3.3.3 本文融合方法.....	30
3.3.4 改进后的 Mean Shift 算法步骤.....	30

3.4 实验结果及分析.....	31
3.5 本章小结.....	32
第四章 自适应调节核窗大小的 Mean Shift 算法.....	33
4.1 自适应调节核窗大小.....	33
4.2 综合方法的 Mean Shift 算法步骤.....	35
4.3 跟踪算法评估指标.....	36
4.4 算法的综合性实验及性能比较.....	39
4.4.1 单个改进方法的性能比较.....	39
4.4.2 综合方法的性能比较.....	41
4.5 本章小结.....	43
第五章 总结与展望.....	45
5.1 总 结.....	45
5.2 未来工作展望.....	45
致 谢.....	47
参 考 文 献.....	49

## 第一章 绪论

### 1.1 人脸跟踪概述

人脸跟踪技术是一种有重要应用价值的技术，是计算机视觉和模式识别等相关领域中的关键技术，在视频会议、高级人机交互、智能监控等方面有重要的应用价值和广泛的应用前景。人脸跟踪的目的是从连续的视频图像序列中捕获人脸目标，获得人脸的运动轨迹和大小等信息，实际上是对人脸真实的位置等状态进行估计。

人脸跟踪作为视觉跟踪的一类子问题而发展起来，而视觉跟踪问题是计算机视觉领域内一个被普遍关注的研究课题。近年来，随着计算机技术的不断发展，计算机能力的极大提高，人类对机器研究的深入和物理等科学知识的引入，使得计算机视觉的研究进入了高速发展阶段，并且在各种不同的领域获得了广泛的应用。

计算机视觉，顾名思义，就是让计算机拥有像人类一样的“视觉”，让机器代替人眼感知世界。视觉信息在人类活动所涉及的各种信息中所占的比重最大，由于它独特的空间特性和结构特性，也不能为其他任何信息所代替。视觉系统的特殊性使得生物具备从二维图像中提取三维信息的能力，但是这对于机器而言却是非常困难的事情，人类在视觉领域的探索一度陷入低谷。近年来，随着信息处理理论的蓬勃发展以及计算机的诞生，人类模拟视觉的梦想出现了新的曙光，人们开始通过使用摄像头模拟人眼采集信息，使用电脑模拟人脑对获取信息进行识别、跟踪和测量，并进一步做图像处理，获得更适合人们观察或更适合仪器处理的图像。自此，新的科学应运而生了，这就是我们所说的计算机视觉<sup>[1]</sup>。

视觉跟踪作为计算机视觉领域的重要研究课题，也是视频应用系统的重要组成部分，它可以对图像序列中运动的目标进行检测，并在检测的基础上进行目标跟踪以获取目标的运动位置、速度和运动轨迹等信息，是更高层视频理解和处理的基础。目前，不论是国际上还是国内，视觉跟踪问题都是研究热点问题，很多的重要国际期刊和国际会议每年都会发表大量关于目标跟踪的文献，国内也有很多的研究机构和大学研究这领域的问题。其中，作为视觉跟踪一类子问题的人脸跟踪也受到了广大研究者的关注。

人脸跟踪技术涉及到模式识别、计算机视觉、图像处理等多个学科，与基于指纹、视网膜等人体生物特征的身份识别有着密切的关系，但人脸相比于指纹、视网膜、虹膜等其它的人体特征更为直接和友好的优点，因此得到了国内外大量



的研究机构和大学的重视。近年来,人脸跟踪的研究在国内外也已经开展了大量相关的项目的研究。

下图是从 CNKI 的学术趋势中搜索“人脸跟踪”词条时得到的。

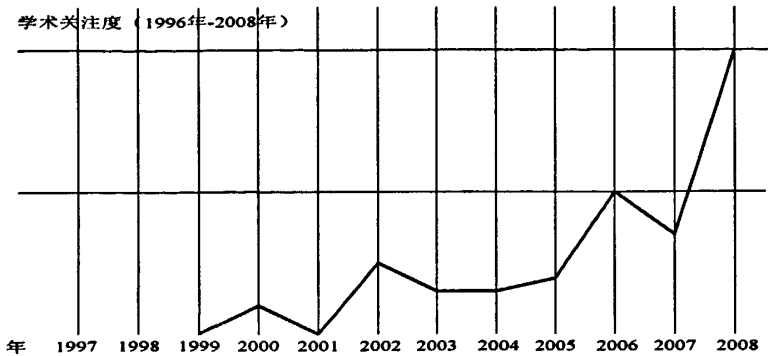


图 1.1 人脸跟踪的学术关注度

图中,横坐标表示具体的年份,纵坐标代表“人脸跟踪”的学术关注度。所谓的学术关注度是以 CNKI 知识资源总库中与“人脸跟踪”关键词最相关的文献数量为基础,统计该关键词做文献主题出现的次数,形成的学术界对某一学术领域关注度的量化表示<sup>[2]</sup>。虽然图中值统计到 2008 年,但很明显这两年的有关“人脸跟踪”的文献只会有增无减。从图中很明显的看出,人脸跟踪技术的研究是得到了国内越来越多的学者的关注。当然,国外也有大量的研究者研究人脸跟踪技术。每年的重要的国际期刊和国际会议都有大量的文献是有关人脸跟踪技术的。

人脸跟踪的研究结果可以应用于民用和军事的很多领域,具有很高的经济价值和前景。它们主要应用在智能视频监控,人体运动分析,摄像机自动聚焦,智能导播,人机交互等等。

## 1.2 相关研究工作

由于跟踪问题本身的复杂性,大量的文献从各种不同的角度研究各种不同形式的跟踪问题。因为应用环境千变万化,跟踪对象也各不相同,并且实际应用系统的需要和视频跟踪问题本质上的复杂性,使得不同的研究工作都对问题做出不同的假设和不同的目的。以人脸跟踪为例,实际的应用环境会影响跟踪时利用的信息:如果作为身份认证识别系统的组成部分,则可以假设摄像机固定,背景比较简单,人脸状况较好且变化较小,因而可以使用基于背景建模的差分机制或者更为复杂地使用脸部网络模型等作为跟踪手段;如果是用于摄像机自动对焦,那么人脸的大小,位姿,光照,以及背景等的变化都会剧烈的多,而且这时对人脸纹理信息可能由于模糊或人脸很小而不足以支持复杂的脸部模型,但是这时对人脸跟踪的精度要求就相对低一些。此外,各种研究工作的侧重点也可能大相径庭,

有些专注于针对具体问题抽取更有效的特征或视觉线索,有些则更多的专注于跟踪过程的一般模型和基本算法框架,等等<sup>[3]</sup>。

一个视觉跟踪系统可以分为以下两部分:视觉跟踪线索和跟踪算法框架。视觉跟踪线索,即跟踪中用到的目标特征信息,用于提供对输入视频数据(图像信息)的观测和理解,例如:颜色信息、纹理信息和运动信息等。而跟踪算法框架是用于利用视觉跟踪线索提供的信息来估计被跟踪物体的状态,如:Mean Shift算法、粒子滤波算法等。下面就详细地就这两方面对现有的研究工作分别进行一些分析和总结。

### 1.2.1 视觉跟踪线索

视频序列中的目标跟踪是以动态图像处理技术为基础的,其根本任务是在每一帧图像中完成对目标的定位。目前目标跟踪的线索是多种多样的,以下主要以人脸跟踪线索为例介绍典型的几种。

#### (1) 基于肤色信息的人脸跟踪

颜色特征是物体最直观的特征,特别是在人脸跟踪中,人脸颜色往往与周围的环境颜色差异很大,利用颜色特征跟踪物体是非常有效的。一些研究证明肤色能够被聚类并且可以很好的与其他颜色区分。颜色信息对于人脸跟踪来说是一种有效的线索,归因于它对人脸的尺度变化和旋转等都具有一定的鲁棒性,因而被很多人脸跟踪算法所采用<sup>[4-6]</sup>。

Stern<sup>[6]</sup>等人利用多个色度空间进行人脸的跟踪,提出一个自适应的颜色空间转换算法。Comaniciu<sup>[7]</sup>等以颜色直方图为跟踪线索,提出一种均值移动(Mean Shift)目标跟踪算法。实验证明该方法计算复杂度低、效率高,因此被广泛地应用在了人脸跟踪中<sup>[8, 9]</sup>。后来,基于目标带宽固定的缺点,Bradski<sup>[10]</sup>在颜色概率分布图概念的基础上提出了顺序自适应均值移动(CAMSHIFT: Continuously Adaptive Mean Shift)人脸跟踪方法。

但是,肤色信息对光照比较敏感,无法处理背景与肤色相似的情况,而且当目标姿态(人脸不可见时)发生变化、目标被遮挡和光照变化时,目标的颜色信息可能发生较大的变化,影响目标之间的匹配问题,使得目标定位不准,这是利用颜色直方图的方法面临的最主要的问题。

#### (2) 基于纹理信息的人脸跟踪

纹理可以认为是灰度在空间以一定的形式变化而产生的模式。一般可以认为是由许多相互接近的、相互编制的元素构成<sup>[11]</sup>。在图像处理及计算机视觉中,纹理特征已经被普遍认为是一种重要的视觉特征,它描述了图像具有的局部不规则而全局又呈现某种规律的物理现象。现在纹理特征被越来越多的研究者应用到视频跟踪中来。

局部二值模式 (Local Binary Pattern 简称 LBP) 是一种纹理分析领域的特征提取方法, 最初是由 Ojala 在文献[12]中提出来的, 是描述图形纹理特征的一种优秀算子。LBP 特征是一种简单的灰度不变的纹理统计特征, 它准确的描述了图像中每一点与其周围点在灰度值上的大小关系, 总结图像的局部特征结构, 通过计算图像的局部信息来实现图像的特征提取过程, 本质上提取的是图像边缘、拐点等局部变化特征。它突出的优点是计算简单且对旋转、亮度变化等不敏感。Wu<sup>[13]</sup>结合 LBP 人脸分类和 CAMSHITF 算法进行人脸跟踪, 在速度和跟踪准确性上都有较好的提高。Wang<sup>[4]</sup>将 LBP 特征和肤色信息结合起来表示目标模型, 利用 Mean Shift 算法进行人脸跟踪, 也取得了较好的效果。Comaniciu D.在文献[7]中也提出了纹理与边缘结合表示目标的设想。

### (3) 基于人脸局部特征的跟踪

基于人脸局部特征跟踪的思想是利用人脸的器官特征信息对人脸进行定位, 该方法利用眼角、鼻尖等面部的一些具有代表性的点的跟踪来完成对整个人脸的跟踪。例如:邢昕<sup>[14]</sup>等提出了利用形态学运算对嘴进行跟踪, 然后通过对人脸对称性的分析, 实现了对头肩序列中人脸的实时跟踪。Strom<sup>[15]</sup>等人通过跟踪特征点来估计三维运动参数, 而后通过自适应地更新纹理来跟踪有较大旋转的模型边缘的特征点<sup>[16]</sup>。

这类算法的关键是特征点的选取问题, 特征点应该具备对目标大小、位置、方向、光照变化不敏感的特性, 因此, 如何选取具有代表性的特征点是这类算法的特点。

### (4) 基于运动信息的人脸跟踪

针对静态背景的人脸跟踪, 另一种常用的方法是根据运动检测来对人脸进行跟踪。运动检测方法主要有差分图像法<sup>[17, 18]</sup>和光流法<sup>[19]</sup>两种。

差分图像法是通过当前图像和背景图像的差或者用当前连续两帧间图像的差来进行动态像素的判断。当差分图像中的像素值超过一定阈值时, 就把这些像素判定为运动像素, 通过这些运动像素和其它特征结合来检测人脸等运动目标。但是这种方法的缺点是仅适合背景固定而且比较简单的情况, 且易受光照影响。处理非刚性目标时存在一定的困难。

光流法也是动态图像分析的重要方法, 通过光流计算获取运动特征进行人脸跟踪也是一种常用的方法。它的主要思想是通过分割二维平面上不同光流得出光流场来进行人脸跟踪。但是它仅仅适用于跟踪具有适度运动的人脸目标, 当人脸具有较大幅度的旋转和大小变化时, 往往会导致跟踪的失败<sup>[16]</sup>。

### (5) 基于轮廓形状的人脸跟踪

目标的轮廓形状也是跟踪中常用的一种信息, 因为人脸近似椭圆的形状, 所以很多文章都是把人脸形状描述成椭圆模型来进行跟踪的<sup>[20]</sup>。

虽然把人脸描述成椭圆很简单,但椭圆毕竟是用一种非常规则的形状去对人脸形状进行近似的。相比之下,可变形的曲线或轮廓描述就会灵活许多,其中,基于活动轮廓模型的人脸跟踪就属于此类方法。活动轮廓模型一般分为两大类:参数活动轮廓模型<sup>[21]</sup>和几何活动轮廓模型<sup>[22]</sup>。

活动轮廓(ACM,是 Snake 模型的衍生)方法是利用一条封闭的参数化的曲线来表示目标的轮廓形状,在由图像构造的特征场中通过极小化以曲线函数为参数的能量来动态迭代,使得该轮廓能够自动地更新。该方法相对于区域的跟踪方法具有计算量小的优点,但是初始化轮廓是算法的关键,而且不易做好。

#### (6) 多线索融合的人脸跟踪

单个线索进行跟踪往往不够鲁棒,比如只用颜色信息跟踪目标时,就丢失了目标的空间信息,在颜色信息发生大的变化时就容易引起跟踪失败。因此,现在有越来越多的研究者将多种线索结合起来应用于跟踪中,增强跟踪的性能。

HS Lee<sup>[23]</sup>将肤色信息和形状信息结合应用于人脸跟踪中。庄莉等<sup>[24]</sup>提出了一种多线索综合的方法,多线索中包括基于深度信息的人脸区域粗分割、基于多关联模板匹配的人脸检测、利用多尺度 Sobel 卷积的特征提取、基于“特征眼”的人眼验证以及基于多视图的校验方法。王永忠<sup>[25]</sup>提出了一种基于多特征自适应融合的核跟踪框架,利用目标的颜色特征与 LBP 纹理特征具体实现了自适应融合的核跟踪方法。Haritaoglu<sup>[26]</sup>等提出了一种基于目标颜色及边缘的核跟踪方法,将目标的 RGB 颜色特征和边缘特征组合起来构造了目标的四维特征,并将其用于基于核的跟踪方法中。

除了以上这些视觉跟踪线索,文献[27]将 SIFT 特征嵌入 Mean Shift 算法进行目标跟踪,F. Porikli<sup>[28]</sup>提出一种协方差矩阵跟踪方法,它是综合多个特征的跟踪方法,这些方法都可以应用于人脸跟踪。

### 1.2.2 基本跟踪算法

视觉跟踪线索实际上是使机器在一定程度上具备对输入的图像和跟踪目标的理解能力,而为了能很好的利用这些线索,我们需要一个跟踪算法框架来对一段视频的每一帧给出跟踪结果。

目前主流的跟踪算法可以分为以下几类:

#### 1) Mean Shift 算法

均值移动(Mean Shift)是一种基于非参数核概率密度估计理论,为在概率空间中求解概率密度的极值问题而发展起来的高效优化求解方法。Comaniciu D<sup>[7]</sup>利用颜色直方图为跟踪线索,提出了一种基于均值移动的目标跟踪算法,该算法能快速高效地获得局部最优解,计算复杂度低、效率高。这些优点使得 Mean Shift 算法在目标跟踪中迅速发展起来,越来越多的学者研究起 Mean Shift 跟踪算法。

Mean Shift 算法我们可以简单地总结为：在初始帧中，利用检测结果或者手动选定目标区域，利用目标的颜色直方图表示目标的模型，计算目标特征空间中所有特征的概率值，在下一帧中，以上一帧中目标的位置作为该帧目标的起始位置，计算该候选区域的候选目标模型，利用相似性函数度量初始帧参考模型和该候选模型间的相似度，通过求最大相似度来获得目标的 Mean Shift 向量，该向量使得目标从初始位置向最可能的目标位置不断迭代，并最终收敛到目标的真实位置，达到目标跟踪的目的。详细的理论和算法在下一章中有详细介绍。

虽然 Mean Shift 算法具有速度快，效率高的优点，但同时也有很多缺陷：首先，跟踪过程中目标模型固定不变，使得由于光照、人体姿态变化等引起颜色较大变化时跟踪容易失败；其次，使用颜色直方图的单一特征表示目标模型时，目标的空间信息丢失，完全放弃了目标运动模型的使用；最后，由于跟踪目标的核窗宽固定，使得当目标存在尺度变化时，跟踪定位不准确或失败。

## 2) Particle Filter 算法<sup>[29-31]</sup>

粒子滤波器 (Particle Filter) 是一种适用于强非线性、无高斯约束的基于模拟的统计滤波器，它利用一定数量的粒子来表示随机变量的后验概率分布，从而可以近似得到任意函数的数学期望，并且能应用于任意的非线性随机系统。

近年来，粒子滤波器成为解决跟踪问题的一个广泛应用的算法。最初是在 1998 年，CONDENSATION<sup>[32]</sup>作为粒子滤波器的一种具体的实现算法被提出来，并且首次成功的应用在视频跟踪中，此后这类方法就逐渐流行起来。粒子滤波通过蒙特卡罗仿真方法实现递推贝叶斯滤波，提供了一种统一开放的理论框架，能有效处理各种非线性、非高斯的系统中。正因为它可以解决目标跟踪中非线性分布的情况，并随着计算机性价比的不断提高，粒子滤波以其实现的灵活性和可并行计算性，越来越受到研究者的重视，逐渐成为跟踪算法的首选。

文献[33]结合粒子滤波跟踪框架，以颜色直方图为跟踪线索给出了基于概率模型的跟踪方法，并取得了一定的跟踪效果；文献[34]提出了以颜色为主，运动及声音等特征为辅，多特征相结合的粒子滤波跟踪方法，用于克服颜色特征的缺陷，但是运动和声音不是持续性的特征，不能持续提供有效信息。

同时，粒子滤波也有自己的缺陷：首先，它的复杂度高，难以实现实时跟踪；其次，在同时跟踪多个目标时，样本会迅速集中到观测模型响应最强的一个目标上，导致多目标跟踪失败。

## 3) Kalman Filter 算法<sup>[35]</sup>

卡尔曼滤波器是最优化自回归的处理算法，是一种高效的递归滤波器，它能够从一系列的不完全包含噪声的测量中估计动态系统的状态。而对于非线性系统的估计问题，最经典的且得到广泛应用的方法是以扩展的卡尔曼滤波为代表，这一类的方法需要对模型进行线性化，而且要求期望的概率密度函数满足高斯分布，

但是在实际的系统建模中,模型往往是非线性高斯的,此时最优估计很难实现,这正是卡尔曼滤波器的一大缺陷。因为它只适用于单高斯分布的线性情况,对于目标跟踪中很普遍的多峰分布非线性情况是无能为力的。

同多线索融合解决单线索不够鲁棒的方法一样,目前也有许多的研究者将两种及以上的算法结合起来,以弥补各自的缺陷,提高跟踪的性能。Cheng Chang 和 Rashid Ansari<sup>[36]</sup>研究了粒子滤波器和 Mean Shift 方法结合的情况,但是粒子滤波本身的复杂性降低了跟踪的实时性。Yang<sup>[37]</sup>提出了一种使用自适应带宽、形状和方向的 Mean Shift 算法,将该算法结合卡尔曼滤波应用于细胞的跟踪,取得了较好的效果。Bai<sup>[38]</sup>将 Mean Shift 嵌入粒子滤波器中对目标进行跟踪,该算法能选择更多具有较大权值的例子并剔除贡献率较小的粒子,从而使粒子滤波算法更快的收敛,实验证明该算法能减少粒子的退化并能降低计算量。

### 1.3 人脸跟踪面临的困难和挑战

人脸跟踪技术虽然已经取得了很大的发展,但是还有许多的问题有待进一步解决,视频序列中人脸跟踪技术仍然面临着巨大的挑战,其主要来自于:

#### (1) 人脸目标的复杂性:

脸部器官具有非刚性特点,由嘴部及眼睛的张和闭、人的一喜一怒等表情变化所引起的脸部器官及肌肉的运动,均会直接造成人脸外观的很大变化。再者,不同的发型,胡须,化妆等都会造成人脸表现形式的变化,况且,人脸的运动也不规则,人脸旋转所引起的人脸姿态变化同样会使人脸跟踪的难度增加。

#### (2) 跟踪环境的复杂性:

对于一个实际的人脸跟踪系统,成像条件一般都是未知的,不同的光照,不同的外部环境,不同的成像设备均会造成人脸图像的巨大差异。

#### (3) 遮挡影响:

对于实际的人脸跟踪环境,很难保证人脸目标不被遮挡,例如,眼镜、围巾等饰物以及其它的背景干扰均有可能造成人脸目标的遮挡,这在很大程度上增加了人脸跟踪的难度。

此外,还有比如摄像机的运动、人脸目标的突然移动、人脸目标的尺度变化和跟踪的时时要求等都给人脸跟踪带来了巨大的挑战。

### 1.4 本论文主要工作和内容安排

本文研究的 Mean Shift 跟踪方法是一种无参密度估计算法,在计算机视觉领域中具有广泛的应用。本文从第二章起,首先详细介绍了基本的 Mean Shift 跟踪

理论和基于颜色直方图的 Mean Shift 跟踪算法。然后根据基本算法中目标模板无法更新、目标模型利用的特征单一性和无法调节核窗宽大小等多种缺陷,依次提出了相应的改进方法。本文一共分为五章,具体安排如下:

第一章为绪论,主要介绍了视觉人脸跟踪的相关知识,人脸跟踪的相关研究成果以及面临的挑战。

第二章介绍了自适应更新目标模型的 Mean Shift 跟踪。该部分详细的说明 Mean Shift 理论和基于颜色直方图的 Mean Shift 跟踪算法,并且根据它目标模板固定的缺陷问题引入了自适应更新目标模型方法,给出了改进后的方法与基本方法的跟踪结果图示。

第三章主要针对 Mean Shift 算法只应用目标的单一颜色特征,而丢失了空间信息的缺点,引入了颜色和纹理结合的多特征融合的目标模型表示方法,并给出了该改进方法与基本方法的跟踪结果图示。

第四章首先介绍利用椭圆拟合的方法自适应调节核窗大小,然后介绍了流行的算法评估软件 Viper 的各个评估参数,最后用四种实验方法比较各种改进方法与基本方法并给出实验结果图示和评估结果。

第五章对文章提出的方法进行了总结和展望。

## 第二章 基于自适应模板更新的 Mean Shift 算法

### 2.1 引言

Mean Shift 算法是一种非参数密度估计的高效模式匹配算法,最早是由 Fukunaga 和 Hostetler<sup>[39]</sup>于 1975 年在一篇关于概率密度梯度函数估计的文章中提出。文中对 Mean Shift 的概念作了贴切的诠释,即偏移的均值向量,然而由于当时缺乏对其更深入的研究,并未开发至应用领域层面。Mean Shift 方法在近 20 年的时间内并没有引起人们的关注,直到 1995 年 Yizong Cheng<sup>[40]</sup>发表了一篇关于 Mean Shift 的重要文献。在这篇文献中,Cheng 对 Mean Shift 算法进行了推广:定义了一族核函数,使得偏移量对均值偏移向量的贡献随着样本与被偏移点的距离不同而变化,设定了一个权重系数,使不同的样本点权重不同,大大地扩大了 Mean Shift 的使用范围,Mean Shift 才引起人们的注意。

近年来 Mean Shift 算法引起了人们的普遍关注,有关的改进算法的研究论文也层出不穷。Mean Shift 算法处理的计算模型对于视频空间分析来说是非常多面的工具,能够提供相当可靠的结果,因此它被用于解决图像分割、目标检测、视频跟踪等多种问题中。Mean Shift 算法基本的计算模块是采用传统的模式识别程序,通过对特征空间的分析,将图像空间中的元素用对应的特征空间以实现图像的分割,如文献[41, 42]。它也可以根据它将领域内的数据点反复迭代移向数据平均值这一特性,直接估计特征空间概率密度函数的局部极大值来获得未知类别的密度模式,并确定这个模式的位置,然后使之聚类到和这个模式有关的类别中去,如文献[40]。在文献[7]Comaniciu 等人以颜色直方图为跟踪线索,提出一种均值移动(Mean Shift)目标跟踪算法,发挥了 Mean Shift 算法能快速高效获得局部最优解的特点,研究并实现了在目标实时跟踪方面的应用<sup>[43]</sup>。

虽然 Mean Shift 在解决目标匹配跟踪方面具有快速有效的优点,但是由于人脸跟踪本身的复杂性(跟踪环境复杂多变,人脸目标姿态大小也变化无常等),使得 Mean Shift 算法有一定的缺陷:首先,跟踪过程中目标模型固定不变,使得由于光照、人体姿态变化等引起颜色较大变化时跟踪容易失败;其次,单纯使用颜色直方图时,目标的空间信息丢失,完全放弃了目标运动模型的使用;最后,由于跟踪目标的核窗宽固定,使得当目标存在尺度变化时,跟踪定位不准确甚至跟踪失败。

近年来,很多的研究者致力于 Mean Shift 算法的改进上,Collins<sup>[44]</sup>将尺度空间和 Mean Shift 相结合解决了核函数带宽实时变化时目标跟踪问题,但算法速度



不好; Yang<sup>[37]</sup>提出了一种使用自适应带宽、形状和方向的 Mean Shift 算法, 将该算法结合卡尔曼滤波应用于细胞的跟踪, 取得了较好的效果。HS Lee<sup>[23]</sup>将肤色信息和形状信息结合应用于人脸跟踪中。王永忠<sup>[25]</sup>提出了一种基于多特征自适应融合的核跟踪框架, 利用目标的颜色特征与 LBP 纹理特征具体实现了自适应融合的核跟踪方法。Bai<sup>[38]</sup>将 Mean Shift 嵌入粒子滤波器中对目标进行跟踪, 该算法能选择更多具有较大权值的例子并剔除贡献率较小的粒子, 从而使粒子滤波算法更快的收敛, 实验证明该算法能减少粒子的退化并能降低计算量。

本文也针对上述缺点提出相应的改进措施, 对跟踪器的鲁棒性进行改进研究是极具意义的。本章主要介绍基本的 Mean Shift 理论以及针对目标模板固定不变问题, 提出相应更新策略。章节安排如下: 2.2 节介绍 Mean Shift 理论, 2.3 节说明基于颜色直方图的 Mean Shift 算法在跟踪中的应用, 2.4 节提出自适应更新目标模板的策略, 并给出了改进后的跟踪算法在实验中应用的效果图和实验结果分析, 2.6 节本章小结。

## 2.2 Mean Shift 理论

Mean Shift 是一种无参核密度的估计函数, 作为一种聚类分析方法, 它的密度估计的梯度是递增的, 其收敛点为密度梯度的局部极大值点, 这个局部极大点其实就对应特征空间中的一个模式。Mean Shift 算法对概率密度函数的估计是由多维核函数估计器衍生而来。

### 2.2.1 核概率密度估计

概率密度估计是根据有限的样本集来估计样本的概率分布, 在工程领域有广泛的应用。而在计算机视觉中处理目标分割和跟踪等问题时, 需要通过对目标的外观进行有效的建模来实现可靠比较的目的, 即用概率密度估计来表征目标。通用的概率密度估计方法一般可分为两种, 即有参概率密度估计和无参概率密度估计。

有参概率密度估计是假设待估计的概率密度函数模型已知, 而只需通过目标区域的数据对概率密度函数的未知参数进行估计。它要求将目标的信息映射到一个特征空间中, 特征空间的随机变量即目标信息的特征值, 通过以上未知参数的估计最终得到整个特征空间的概率密度分布函数。常用的有参概率密度估计方法有矩估计法和最大似然估计方法。它的优点是能够实现有效的数据压缩, 仅用几个简单的参数就能描述全部数据的分布, 复杂度较低, 在工程实践中能有效的提高工作效率。但同时也带来了缺点, 那就是必须事先知道概率密度分布函数的形式, 同时它的选择缺乏有效的指导和理论依据, 一旦选用错误的密度分布函数

将会带来较大的估计误差, 而且还会由于数据偶然表现的局部特性收敛到不合要求的区域内, 不利于进行深入的研究。

而针对有参概率密度估计中分布函数难以选择确定的缺点, 一些学者又提出了无参概率密度估计方法。无参概率密度估计方法是从一个范围较宽的函数集中估计概率密度, 而不是局限于参数化的函数集。它没有显式的概率分布函数, 而是从观测数据直接来分析得出概率分布空间中各点的概率密度值, 有效的解决了参数密度估计带来的问题, 具有很强的实用价值。常见的无参概率密度估计方法有: 直方图法、最近邻域法和核概率密度估计方法。

在以上方法中, 最为常用的是核概率密度估计法。直方图估计方法是把数据的值域分成多个相等的区间, 数据按照这些区间分成对应的多个组, 每组对应一个矩形, 矩形的高与数据大小成正比, 这些矩形依次排列形成对应于数据概率分布的直方图。而最近邻域法是一个迭代函数作为迭加基函数, 每个样本点处作为迭加节点, 使用  $k_n$  个以样本  $x_i$  为中心的迭代函数迭加对  $x$  处的概率密度进行估计。其中, 样本较密集的区域上必然概率密度估计值(迭加函数)较大, 因此可以通过估计一个迭加函数来估计概率密度函数。但这两种方法各有各的缺陷: 前者对维数低的数据更有效, 后者则容易受到噪声的干扰<sup>[43]</sup>。

核概率密度估计法是在直方图估计法的基础上增加了一个用于平滑数据的核函数, 该方法有效的解决了以上两种方法的缺陷, 克服了直方图估计对高维数据的失效性, 能较好的抑制噪声的影响, 增强数据的有效性, 特别适合中小规模数据的概率密度估计, 且具有良好的概率估计性能。

核概率密度估计的主要思想是: 针对一组采样数据, 把数据的值域分成若干相等的区间, 每个区间称为一个 bin, 数据就对应各区间分成若干组, 每组数据的个数与总参样个数的比率就是每个 bin 对应的概率值。同时通过定义一组核函数来平滑数据, 使得随样本与被偏移点距离的不同, 其偏移量对偏移向量的贡献也不同, 从而形成对目标的有效估计。其数学原型如下<sup>[41]</sup>:

设  $d$  维欧式空间  $R^d$  中的观测数据样本集为  $\{x_i\}_{i=1,2,\dots,n}$ , 那么典型的核概率密度可定义为:

$$\hat{f}_k(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (2-1)$$

式中  $K(x)$  为  $R^d$  上的可测函数, 也称窗或核函数,  $h$  是同  $n$  有关的正数, 称其为窗宽, 为核函数的带宽,  $n$  为样本容量。核密度估计既与样本有关, 也与核函数及窗宽的选择有关, 在给定样本以后, 核概率密度估计的好坏取决于核函数及窗宽的选择。这两个条件直接影响了密度函数估计的精度。一般核函数都是对称的密度函数族, 核函数  $K(x)$  必须满足以下条件:

$$K(-x) = K(x); K(x) \geq 0; \int K(x) dx = 1, \quad (2-2)$$

由上式可见：核函数都具有对称性，单峰性和有限局部支撑性。

常用的用于核概率密度估计的核函数主要有<sup>[16]</sup>：

Uniform 核函数：

$$K_U(x) = \begin{cases} c & \|x\| \leq 1 \\ 0 & \text{其它} \end{cases}, \quad (2-3)$$

Epanechnikov 核函数：

$$K_K(x) = \begin{cases} c(1 - \|x\|^2) & \|x\| \leq 1 \\ 0 & \text{其它} \end{cases} \quad (2-4)$$

Normal 核函数：

$$K_N(x) = c \exp\left(-\frac{1}{2}\|x\|^2\right) \quad (2-5)$$

以上三种核函数在一维空间中的效果图分别如下图 2.1 所示。

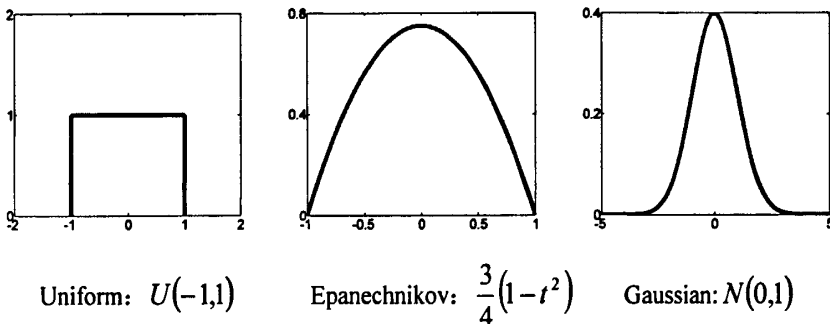


图 2.1 各种核函数图示<sup>[43]</sup>

由式(2-1)可以看出， $x$  处的概率密度估计值实际上就是以  $x$  为中心，以  $h$  为辐射半径范围内观测数据的加权平均。

一般情况下，对于一个各向对称的核函数  $K(x)$  而言，存在一个函数  $k(x)$  使得下式成立：

$$K(x) = c_k k(\|x\|^2) \quad (2-6)$$

式中  $c_k$  为常数，一般把式中的  $k(x)$  称为核函数  $K(x)$  的轮廓函数。该公式中只有当  $x \geq 0$  时将式(2-6)代入式(2-1)中可以得到：

$$\hat{f}_k(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) = \frac{c_k}{nh^d} \sum_{i=1}^n k\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \quad (2-7)$$

在密度函数为  $f_k(x)$  的特征空间中，Mean Shift 算法分析的第一步就是找到这个密

度模式，然后对这个模式进行相关的聚类。

### 2.2.2 密度梯度估计

Mean Shift 的实质就是在概率空间中寻求概率密度的极值点，而基于梯度的方法是极值求解的一种常用手段，Mean Shift 也是建立在对概率密度梯度的求解之上。

对式(2-6)所示的概率密度估计作梯度计算可得：

$$\hat{\nabla}f_k(\mathbf{x}) \equiv \nabla \hat{f}_k(\mathbf{x}) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^n (\mathbf{x} - \mathbf{x}_i) k' \left( \left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \quad (2-8)$$

现定义  $g(\mathbf{x}) = -k'(\mathbf{x})$ ，并把以  $g(\mathbf{x})$  为轮廓函数的核函数  $G(\mathbf{x})$  记为：

$$G(\mathbf{x}) = c_G g(\|\mathbf{x}\|^2) \quad (2-9)$$

式中  $c_G$  为归一化常数，通常把核函数  $K(x)$  称为  $G(x)$  的影子函数。由上面的定义，式(2-8)可以写成：

$$\begin{aligned} \hat{\nabla}f_k(\mathbf{x}) &= \frac{2c_k}{nh^{d+2}} \sum_{i=1}^n (\mathbf{x}_i - \mathbf{x}) g \left( \left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \\ &= \frac{2c_k}{nh^{d+2}} \left[ \sum_{i=1}^n g \left( \left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \right] \left[ \frac{\sum_{i=1}^n \mathbf{x}_i g \left( \left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^n g \left( \left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)} - \mathbf{x} \right] \end{aligned} \quad (2-10)$$

其中  $\sum_{i=1}^n g \left( \left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)$  是一个正数。式(2-10)中前半部分正比于以  $G(\mathbf{x})$  为核函数的

概率密度估计，后半部分为观测数据集的加权平均值与  $\mathbf{x}$  的差。通常把后半部分直观地称为均值移动向量，这也正是均值移动算法名字的由来。根据上述式(2-10)两部分的含义，可以把它改成如下形式：

$$\hat{\nabla}f_k(\mathbf{x}) = \frac{2c_k}{h^2 c_G} \hat{f}_G(\mathbf{x}) m_G(\mathbf{x}) \quad (2-11)$$

式中

$$\hat{f}_G(\mathbf{x}) = \frac{2c_G}{nh^d} \sum_{i=1}^n g \left( \left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \quad (2-12)$$

$$m_G(\mathbf{x}) = \frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \quad (2-13)$$

由式(2-11)可得:

$$m_G(\mathbf{x}) = \frac{1}{2} h^2 \frac{c_G}{c_K} \frac{\hat{\nabla} f_K(\mathbf{x})}{\hat{f}_G(\mathbf{x})} = \frac{1}{2} h^2 c \frac{\hat{\nabla} f_K(\mathbf{x})}{\hat{f}_G(\mathbf{x})} \quad (2-14)$$

式(2-14)表明, 基于核函数  $G(\mathbf{x})$  的均值移动向量  $m_G(\mathbf{x})$  和以  $K(\mathbf{x})$  为核函数的概率密度的梯度方向  $\hat{\nabla} f_K(\mathbf{x})$  一致, 因此可以说均值移动的方向是指向概率密度增长最快的方向。从式(2-14)也看出, 均值移动步长随着  $\hat{f}_G(\mathbf{x})$  自适应调节, 当  $\hat{f}_G(\mathbf{x})$  较大时, 也即在概率密度大的区域, 步长较小, 反之当  $\hat{f}_G(\mathbf{x})$  小时, 步长却较大, 所以说 Mean Shift 算法是一种适应性的梯度上升算法, 这正是我们期望的 Mean Shift 的寻优过程。

下面说明 Mean Shift 算法收敛的一个充分条件<sup>[41]</sup>。根据以上推导, 均值移动的过程可以定义为计算平移矢量  $m_G(\mathbf{x})$  和根据  $m_G(\mathbf{x})$  变更窗函数  $G(\mathbf{x})$  的中心的一个递归的过程。假设  $\{y_j\}_{j=1,2,\dots}$  表示核函数  $G(\mathbf{x})$  中心位置的连续位置序列, 其中  $y_{j+1}$  是  $y_j$  点处用核  $G(\mathbf{x})$  计算得到的加权均值, 而  $y_1$  是核的初始中心位置。则:

$$y_{j+1} = \frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{y_j - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{y_j - \mathbf{x}_i}{h}\right\|^2\right)}, \quad j = 1, 2, \dots, n \quad (2-15)$$

假设这些连续位置上相应的密度估计组成一个序列, 则:

$$\hat{f}_K = \{\hat{f}_K(y_j)\}_{j=1,2,\dots,n} \equiv \{\hat{f}_K(y_j)\}_{j=1,2,\dots,n}, \quad (2-16)$$

基于 Mean Shift 的模式测试是首先得到  $\hat{\nabla} f_K(\mathbf{x}) = 0$  的点, 这个点是  $\hat{f}_K(\mathbf{x})$  的一个稳定点, 然后找到它的所有稳定点, 最后再修整这些稳定点而只保留极大值点。这个极大值点即为  $\hat{f}_K$  的一个模式, 那么收敛于这个模式的所有稳定点的集合 (即被 Mean Shift 程序访问过的点都收敛于一个模式) 的集合就构成了这个模式的收敛域, 这个收敛域既是待分割的一个任意类<sup>[43]</sup>。

以下两幅图显示了一个用  $L^*u^*v$  的色彩模型的彩色图像的特征空间分析聚类方法的一个实例<sup>[41]</sup>。



图 2.2 一幅 400\*276 的彩色图像

下图 2.3 是上图 2.2 的二维空间表示, 以及对其密度梯度进行估计后得到的七个聚类模式和收敛到每个模式的轨迹,  $L^*u^*v$  空间和  $L^*a^*b$  空间是两类最接近感知色彩的标准色彩空间。在这两类空间中,  $L$  是表示亮度的坐标, 不同的是色度坐标的表示。事实上, 用  $L^*u^*v$  或者  $L^*a^*b$  没有明显的差别, 在 Mean Shift 算法中由于线性映射性质使用了  $L^*u^*v$  空间。

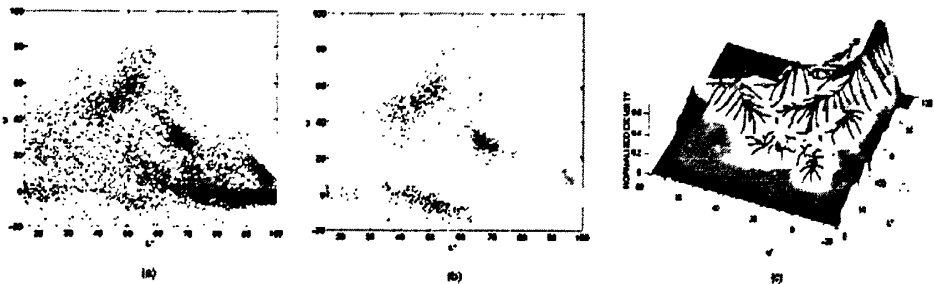


图 2.3 2D 特征空间的分析。(a)  $L^*u^*v$  色彩空间中 110400 个  $L^*u$  二维数据点集, (b) 运行 159 次 Mean Shift 程序进行聚类的结果, (c) 对(a)和(b)的数据集合进行密度梯度估计后得到的聚类的轨迹, 红点表示最终聚类的不同类别

此外, 文献[45]中李乡儒基于多重假设证明了均值移动算法的收敛性, 经过若干次均值移动操作后, 其可以收敛到概率密度的局部极值点, TL Shieh<sup>[46]</sup>也分析和研究了 Mean Shift 算法的收敛性问题。

### 2.3 基于颜色直方图的 Mean Shift 跟踪算法

前面提到了多种视觉跟踪线索, 而最基本也最常用的是目标的颜色信息。颜色直方图则是利用颜色信息表示目标特征的简单且有效的方式。目标的颜色直方图具有特征稳定、抗部分遮挡、计算方法简单和计算量小的特点, 因此基于 Mean Shift 的跟踪一般采用颜色直方图对目标进行建模, 然后利用梯度优化方法来减少特征搜索匹配的时间, 实现快速的目标定位, 同时还利用 Bhattacharyya 距离作为对目标模板和候选目标的相似性测度, 完成特征的匹配。

Comaniciu<sup>[7]</sup>等利用颜色直方图为跟踪线索,将均值移动理论引入到视频序列目标跟踪问题中,提出了一种基于均值移动的目标跟踪算法,实验证明该方法计算复杂度低、效率高。

Mean Shift 算法是通过人机交互的方式对被跟踪的目标进行初始化的。在初始帧中,手动地选定一个包含所有目标特征的矩形或者椭圆,即为被跟踪目标的目标区域,也就是核函数作用的区域,区域的大小等于核函数的带宽,本文中我们用矩形区域表示目标区域。假定我们在彩色的视频序列中跟踪目标,则最常用的颜色空间是 RGB 颜色空间,按照直方图的方式将 RGB 颜色空间中的每个子空间 R、G 和 B 空间分成  $K$  个相等的区间,将每个区间称为一个 bin,整体构成特征空间,特征空间中的特征值的个数(即总 bin 数)为  $m = k^3$ 。

统计初始帧目标区域内的所有像素点,计算特征空间中每个特征值的概率,其实就是目标区域的颜色直方图,它即为目标参考模型。在之后的视频帧中可能存在目标的候选区域中,用相同的方法(即颜色直方图)得到每个特征值的概率,它即为目标的候选模型。然后利用相似性函数度量初始帧目标参考模型和当前帧的目标候选模型的相似度,通过求相似性函数最大得到关于目标的 Mean Shift 向量,这个向量即为上文中提到的 Mean Shift 矢量,它指向概率密度增长最快的方向,最终通过不断迭代收敛到目标的真实位置,从而达到跟踪的目的。以下是对上述算法的详细描述。

### 2.3.1 目标参考模型的表示

假设目标区域中有  $n$  个像素,  $\{\mathbf{x}_i^*\}_{i=1..n}$  为各像素点相对于目标中心位置的坐标,特征值 bin 的个数为  $m$  个,  $\beta = 1..m$  为各直方图段数索引值,则目标模型的核颜色直方图表示为:

$$\hat{q}_\beta = c \sum_{i=1}^n k(\|\mathbf{x}_i^*\|^2) \delta[b(\mathbf{x}_i^*) - \beta] \quad (2-17)$$

式中  $k(\bullet)$  为加权核函数,  $\delta(\bullet)$  为 Kronecker  $\Delta$  函数,定义为:

$$\delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}, \quad (2-18)$$

$b(\mathbf{x}_i^*)$  表示像素点  $\mathbf{x}_i^*$  所在直方图段数的指示函数,由于直方图归一化即  $\sum_{\beta=1}^m \hat{q}_\beta = 1$ ,

所以归一化常数  $c$  为:

$$C = \frac{1}{\sum_{i=1}^n k\left(\left\|\mathbf{x}_i\right\|^2\right)} \quad (2-19)$$

### 2.3.2 候选目标的表示

设  $\{\mathbf{x}_i\}_{i=1..n_k}$  是当前帧中以  $\mathbf{y}$  为中心的候选目标中各像素点的位置坐标， $\beta = 1..m$  为各直方图段数索引值，利用尺度为  $h$  的核函数  $k(x)$  的候选目标的核颜色直方图可以表示为：

$$\hat{p}_\beta(\mathbf{y}) = c_h \sum_{i=1}^{n_k} k\left[\left\|\frac{\mathbf{y} - \mathbf{x}_i}{h}\right\|^2\right] \delta[b(\mathbf{x}_i) - \beta] \quad (2-20)$$

同上，归一化常数  $c_h$  为：

$$c_h = \frac{1}{\sum_{i=1}^{n_k} k\left(\left\|\frac{\mathbf{y} - \mathbf{x}_i}{h}\right\|^2\right)} \quad (2-21)$$

### 2.3.3 基于相似性函数的测度

相似性函数描述的是目标参考模型和目标候选之间的相似程度，在理想情况下两个模型的概率分布应该是一样的。这种函数有很多，比如：Bhattacharyya 系数、Fisher linear discriminant 等。Comaniciu 在文献<sup>[47]</sup>中说明在 Mean Shift 算法中 Bhattacharyya 系数是优于其它相似性函数的一种选择。

定义目标模型和目标候选两个离散分布之间的距离为：

$$d(\mathbf{y}) = \sqrt{1 - \hat{\rho}(\mathbf{y})} \quad (2-22)$$

其中：

$$\hat{\rho}(\mathbf{y}) \equiv \rho[\hat{p}(\mathbf{y}), \hat{q}] = \sum_{\beta=1}^m \sqrt{\hat{p}_\beta(\mathbf{y}) \hat{q}_\beta} \quad (2-23)$$

式(2-23)即为目标模型  $\hat{q}$  和当前候选目标分布  $\hat{p}(\mathbf{y})$  的 Bhattacharyya 系数。在当前帧中进行目标定位问题实质上就是求式(2-23)最大的  $\mathbf{y}$  值。



## 2.3.4 目标定位

为了使  $d(y)$  最小或者  $\rho(y)$  最大, 在当前帧的目标中心先定位为前一帧中目标中心的位置  $y_0$ , 在当前帧中从这一点开始寻找最优匹配的目标新位置。首先计算当前帧估计位置  $\hat{y}_0$  对应的核颜色直方图  $\{\hat{p}_\beta(\hat{y}_0)\}_{\beta=1,\dots,m}$ , 当当前帧的目标模型的核颜色直方图  $\{\hat{p}_\beta(y)\}_{\beta=1,\dots,m}$  相对于  $\{\hat{p}_\beta(\hat{y}_0)\}_{\beta=1,\dots,m}$  变化不大, 且对于所有的  $\beta = 1\dots m$  都满足  $\hat{p}_\beta(\hat{y}_0) > 0$  时, 对式(2-23)在  $\hat{p}_\beta(\hat{y}_0)$  附近进行 Taylor 展开, 可以得到一个线性近似:

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{\beta=1}^m \sqrt{\hat{p}_\beta(\hat{y}_0) \hat{q}_\beta} + \frac{1}{2} \sum_{\beta=1}^m \hat{p}_\beta(y) \sqrt{\frac{\hat{q}_\beta}{\hat{p}_\beta(\hat{y}_0)}} \quad (2-24)$$

将式(2-20)代入上式得到:

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{\beta=1}^m \sqrt{\hat{p}_\beta(\hat{y}_0) \hat{q}_\beta} + \frac{c_h}{2} \sum_{i=1}^{n_h} \omega_i k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \quad (2-25)$$

其中:

$$\omega_i = \sum_{\beta=1}^m \sqrt{\frac{\hat{q}_\beta}{\hat{p}_\beta(\hat{y}_0)}} \delta[b(x_i) - \beta] \quad (2-26)$$

式(2-24)中第一项与  $y$  无关, 为了得到最大的 Bhattacharyya 系数, 只需使式中第二项最大。而第二项是在当前帧中利用核函数  $k(x)$  和各像素点加权值计算得到的概率密度估计。因此根据均值移动理论, 该概率密度估计的极值点可以从  $y_0$  处递归得到, 最终计算得到新的目标位置  $y_1$  为:

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i \omega_i g\left[\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right]}{\sum_{i=1}^{n_h} \omega_i g\left[\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right]} \quad (2-27)$$

其中:

$$g(x) = -k'(x) \quad (2-28)$$

## 2.4 自适应更新目标模板的跟踪算法

### 2.4.1 固定的目标模板

采用固定模板的 Mean Shift 跟踪算法在跟踪背景与目标直方图相似时有一个缺点是：当目标尺寸增大或者减小并且目标附近背景直方图与目标模型直方图相似时，会导致背景被包含进来或者跟踪失败。

此外，在跟踪过程中，由于光照，姿态的变化，目标的颜色可能发生很大的改变，因此在跟踪初始建立的单个参考直方图很可能不够理想，很可能导致跟踪定位不准甚至失败。在这种复杂情况下仍能准确的跟踪目标就是本节要解决的问题。

### 2.4.2 自适应更新目标模板

前面提到，跟踪算法中使用最多的视觉线索是目标的颜色信息。对于颜色信息的利用，我们一般采用颜色直方图的形式。使用颜色直方图作为观测模型有以下三大要素：1)直方图的构成：包含颜色空间的选取、离散化和直方图 bin 的设置；2)候选直方图和参考直方图的相似程度的度量准则；3)被跟踪目标的参考直方图的获取。前面两点我们在 2.3 节有介绍，以下着重介绍一下参考直方图的获取问题。

参考直方图的建立是一个比较重要的问题。如果参考直方图不准确，结果将是灾难性的，因为这样就表示着整个颜色观测模型失效。一般情况下，参考直方图的确立是在跟踪开始时由人脸检测器自动初始化或手工初始化确定，因此一般比较准确。

但是，在跟踪过程中，由于光照、目标姿态的变化等原因，目标的颜色可能发生很大的改变而导致跟踪初始建立的参考直方图不能反应目标的实际情况，最终会导致跟踪失败。其中，目标姿态变化是直方图变化的一大因素。目前主要有两种方法可以在跟踪过程中自适应更新目标参考模型直方图。

#### (1) 引入目标的姿态变化信息<sup>[3]</sup>

人脸目标姿态变化主要有正面人脸、侧面人脸和人脸不可见的情况，针对人脸目标的姿态变化引起的目标颜色直方图变化的情况，改进如下：在算法中建立三种不同的参考直方图，分别对应人脸正面、侧面和不可见的情况，并且在人脸目标首次出现相应的姿态时进行目标参考模型的初始化。在使用颜色模型时，假定目标在两帧之间没有明显的姿态变化，则可根据上一帧的人脸目标姿态来选择最邻近的参考直方图，与之进行比较，计算色彩似然度。这种方法主要针对应用多种视频线索的情况，不同的姿态使用的参考直方图不同，分别由不同的视觉线

索得到。

## (2) 在线更新参考直方图<sup>[3]</sup>

另一种常用的方法是在线更新参考直方图。之所以要更新直方图是因为，即便根据以上的姿态划分得到不同的参考直方图，但目标的色彩还受到姿态以外因素的影响，另一方面，姿态划分是离散的，而目标变化是连续的。

参考直方图的更新方法一般是：

设当前参考直方图为  $H = (h_1, h_2, \dots, h_k)$ ，当前帧输出目标  $\hat{X}$  的直方图为

$\hat{H} = (\hat{h}_1, \hat{h}_2, \dots, \hat{h}_k)$ ，认为它们满足关系：

$$\tau \dot{H} = \hat{H} - H \quad (2-29)$$

其中  $\dot{H}$  为  $H$  的变化速度（一阶导数）， $\tau$  为常数。

假设新参考直方图为  $H'$ ，则  $\dot{H} = H' - H$ ，由上式得：

$$H' = \frac{1}{\tau} \hat{H} + \left(1 - \frac{1}{\tau}\right) H \quad (2-30)$$

目前，大多数涉及更新直方图的文献中[48-51]，采用的方法都是式(2-30)。其中， $\tau$  的选取有些是固定常数，有些则是在跟踪中动态变化。实际上  $\tau$  越小，直方图更新幅度越大，其反映的是当前色彩观测模型的可靠程度。

然而，任何包含在线更新的算法都会面临丧失稳定性的危险。参考直方图错误更新时，若在其后的跟踪中不断放大误差，也会导致跟踪失败。

本文结合上述两种方法来实现自适应更新目标模板，不仅克服了利用目标姿态变化信息中利用多线索的繁琐，而且利用了在线更新的方法且不失鲁棒性。因为方法一中要使用多种视觉线索，对于仅仅使用颜色直方图的跟踪线索来说，本文只是应用了姿态变化的思想。当当前帧的目标姿态与初始目标姿态一致时，参考直方图不变，否则认为目标的姿态发生变化，参考直方图由初始目标直方图和上一帧的目标直方图共同决定，即用方法二得到。本文中用初始帧目标模型直方图和当前候选直方图的相似度来界定目标的姿态是否发生变化。具体方法是：用初始帧目标模型直方图和候选直方图计算核内每个像素的权重后，求所有像素权重的统计平均值，设定一个阈值，当平均值大于该阈值时，认为当前目标姿态与初始目标姿态一致，否则不一致。

下面用具体模型来实现上述的改进方法，应用上述方法自适应更新目标模型时，目标参考模型和目标候选模型间的距离和 Bhattacharyya 系数公式都相应的发生改变，式(2-22)和(2-23)分别改写为：

$$d(\mathbf{y}) = \sqrt{1 - \hat{\rho}[\mathbf{p}(\mathbf{y}), \hat{\mathbf{q}}; r]} \quad (2-31)$$

$$\hat{\rho}(\mathbf{y}) \equiv \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}; r] = \sum_{\beta=1}^m \sqrt{\lambda \hat{p}_{\beta}(\mathbf{y}) \hat{q}_{\beta} + (1-\lambda) \hat{p}_{\beta}(\mathbf{y}) \hat{r}_{\beta}} \quad (2-32)$$

其中:  $r$  是上一帧输出目标的颜色直方图, 对应式(2-30)的  $\hat{H}$ ,  $\lambda$  表示的是  $\hat{q}_{\beta}$  和  $\hat{r}_{\beta}$  在相关系数中的权重, 对应式(2-30)中的  $\tau$ 。

由式(2-31)和(2-32), 我们用 2.3 节中的推导方法推导最终可以得到:

$$\rho[\mathbf{y}] \approx \frac{1}{2} \sum_{\beta=1}^m \sqrt{\hat{p}_{\beta}(\hat{\mathbf{y}}_0) (\lambda \hat{q}_{\beta} + (1-\lambda) \hat{r}_{\beta})} + \frac{c_h}{2} \sum_{i=1}^{n_h} \omega_i k \left( \left\| \frac{\mathbf{y} - \mathbf{x}_i}{h} \right\|^2 \right) \quad (2-33)$$

$$\omega_i = \sum_{\beta=1}^m \sqrt{\frac{\lambda \hat{q}_{\beta} + (1-\lambda) \hat{r}_{\beta}}{\hat{p}_{\beta}(\hat{\mathbf{y}}_0)}} \delta[b(\mathbf{x}_i) - \beta] \quad (2-34)$$

将上式(2-34)代入式(2-27)中, 我们就可计算得到在加入直方图更新方法后的目标新位置。

#### 2.4.3 改进后的 Mean Shift 算法步骤

将自适应更新目标模型的思想融合到实际的 Mean Shift 算法中去, 假定初始帧中手动选取目标区域, 并获得目标模型为  $\{\hat{\mathbf{q}}_{\beta}\}_{\beta=1 \dots m}$ , 可以获得如下的目标跟踪定位算法步骤:

- 1) 在上一帧中利用公式(2-17)计算上一帧的目标模型  $\{\hat{\mathbf{r}}_{\beta}\}_{\beta=1 \dots m}$ ;
- 2) 将上一帧目标的位置定为当前帧目标的初始位置  $\mathbf{y}_0$ , 利用公式(2-20)计算当前帧的目标候选直方图  $\{\hat{p}_{\beta}(\hat{\mathbf{y}}_0)\}_{\beta=1 \dots m}$  和 Bhattacharyya 系数

$$\hat{\rho}(\mathbf{y}_0) \equiv \rho[\hat{\mathbf{p}}(\mathbf{y}_0), \hat{\mathbf{q}}] = \sum_{\beta=1}^m \sqrt{\hat{p}_{\beta}(\mathbf{y}_0) \hat{q}_{\beta}}$$

- 3) 利用公式(2-26)计算图像像素的加权  $\{\omega_i\}_{i=1 \dots n_h}$ ;
- 4) 计算核内所有像素的权重的统计平均值, 当它大于一定阈值 (本文给定经验值), 则利用(2-27)计算候选目标的下一个新位置  $\mathbf{y}_1$ , 否则利用式(2-34)、和  $\{\hat{\mathbf{r}}_{\beta}\}_{\beta=1 \dots m}$  重新计算各像素的权重和候选目标的新位置  $\mathbf{y}_1$ ;
- 5) 循环判断  $\rho[\hat{\mathbf{p}}(\hat{\mathbf{y}}_1), \hat{\mathbf{q}}] < \rho[\hat{\mathbf{p}}(\hat{\mathbf{y}}_0), \hat{\mathbf{q}}]$ , 若满足则  $\hat{\mathbf{y}}_1 \leftarrow \frac{1}{2}(\hat{\mathbf{y}}_0 + \hat{\mathbf{y}}_1)$ ;
- 6) 判断循环终止条件  $\|\hat{\mathbf{y}}_1 - \hat{\mathbf{y}}_0\| < \varepsilon$ , 若满足则终止循环, 否则  $\hat{\mathbf{y}}_1 \leftarrow \hat{\mathbf{y}}_0$ , 并返回

第 2)步继续寻找满足条件的候选目标位置;

7) 确定最终的目标位置后, 利用公式(2-17)更新当前帧的目标模型  $\{\hat{r}_\beta\}_{\beta=1\dots m}$ , 已备后用。

第 6)步中的条件阈值  $\varepsilon$  是通过限制矢量  $\hat{y}_0$  和  $\hat{y}_1$ , 在原始图像的坐标中有同样的位置而获得的, 较低的阈值可以保证搜索结果有更好的精度。同时为了保证跟踪过程的实时性, 可以限制搜索的迭代次数, 一般这个迭代次数为 15, 而在实验中发现实际的迭代次数一般不超过 5 次。在上面的均值偏移算法步骤中, 通常可以不需要第 6)步, 因为这一步是为了防止基于最大化的均值偏移在数学计算上出现问题而设计的, 此问题多是由于 Bhattacharyya 系数的线性近似而产生的。

## 2.5 实验结果与分析

本章将 Mean Shift 跟踪算法应用到视频序列的人脸跟踪中, 实验环境是 Matlab2010。以下实验中视频均来自于斯坦福大学的视觉实验室提供的专门为检测人脸跟踪算法的人脸序列, 本实验中视频 seq-dk 共 50 帧, 帧率为 16 帧/S, 包含人脸姿态变化、人脸不可见的情况。下图是基本 Mean Shift 方法和加入自适应更新目标模型改进后的 Mean Shift 方法的跟踪效果图。

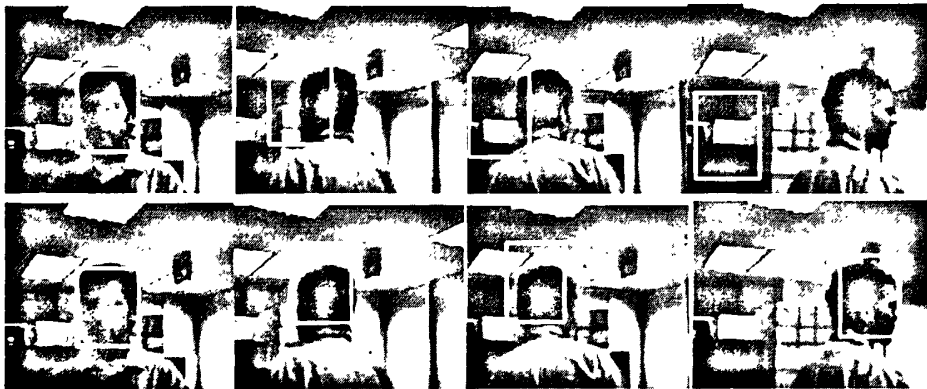


图 2.4 两种方法的效果图, 分别对应视频的 1、38、43、50 帧

上行是基本 Mean Shift 方法的结果, 下行是加入自适应更新目标模型的改进方法后的结果。首帧是手动选定的, 两种方法首帧目标区域近似, 结果显示基本方法从 38 帧开始丢失目标, 直到后来完全丢失, 再也没找回目标, 而改进后的方法可以很好的跟踪目标到最后。

实现结果表明, 加入自适应模板更新策略后, 该改进方法可以实时的判断人脸目标姿态有没有发生变化, 在判断人脸姿态变化后可以自适应的更新目标的参考模板, 以便更准确的跟踪人脸及头部的位置, 因此自适应更新模板的改进方法对人脸姿态变化情况下的跟踪是很有效的。

## 2.6 本章小结

本章首先介绍了 Mean Shift 理论和基于颜色直方图的 Mean Shift 跟踪算法，主要包含核概率密度估计和其梯度估计、目标参考模型和候选模型的表示及两者的相似性度量、目标定位等，然后针对基本 Mean Shift 跟踪算法中目标模型固定的缺点提出了自适应目标模型更新的改进方法，最后给出改进方法的实验结果显示和分析。

实验结果表明，自适应更新目标模型的方法可以实时的判断人脸目标的姿态有无变化并相应的更新目标的参考模型，该方法对有人脸目标姿态变化的视频跟踪有明显的效果，即便在人脸不可见的情况下，仍然可以准确的跟踪到人脸及头部，但是该方法也有一点缺点，就是当有其它人脸干扰被跟踪人脸目标导致跟踪失败后，很难再重新跟踪到被跟踪目标，原因在于当目标姿态变化后，目标的参考模型也自适应的随之改变，无法再还原到原跟踪目标的情况。虽然如此，但结合下文颜色和纹理结合的目标模型表示的改进方法后，即便有其它人脸干扰，改进方法仍能准确的跟踪回原被跟踪目标。



## 第三章 颜色-纹理结合的 Mean Shift 算法

### 3.1 引言

目前大多数的跟踪算法都是基于单个视觉特征,然而在实际的复杂的跟踪场景中,基于单个视觉特征的目标描述很难对所有的场景变化都足够鲁棒,因此基于单个视觉特征的跟踪算法往往变得不可靠。如果能将多个视觉特征融合起来一起加入跟踪算法中,则可以利用不同特征之间的互补性更好的适应场景的变化,更鲁棒的跟踪场景变化中的目标。有一些研究者的研究结论可以证明这一点。例如:Perez<sup>[34]</sup>等提出了一种组合目标颜色、运动及声音等特征的粒子滤波跟踪方法,其中颜色作为跟踪的主要特征,而运动和声音特征用于克服颜色特征的缺点;XU<sup>[52]</sup>等人在对人的头部跟踪时结合了目标的颜色及梯度特征。这些方法都利用了目标的多个视觉特征,但是它们都是基于贝叶斯概率框架且缺乏对场景变化的适应性,而且运算复杂度高使得实时跟踪困难。

与粒子滤波的跟踪方法相比,Mean Shift 方法具有运算简单快捷的优点。因此将多个特征融合到 Mean Shift 算法中,既可以增强算法的鲁棒性还可以达到实时性的要求。

而 Mean Shift 作为一种基于核密度估计的无参快速模式匹配的算法,能否准确的跟踪到目标很大程度上取决于目标模式的表示方法。大多数现有的 Mean Shift 及其改进算法也都是基于单个视觉特征,如颜色特征等,但在实际应用环境中,由于场景变化较大,很难保证跟踪目标算法的鲁棒性。当亮度变化过于剧烈或跟踪目标和背景颜色相似时,很容易跟丢目标。因此有不少研究者在 Mean Shift 算法中引入了多特征融合的目标跟踪,例如:Haritaoglu<sup>[26]</sup>等结合目标的 RGB 颜色特征和边缘特征,将其用于基于核的跟踪方法中,但是该方法对目标和场景的变化较敏感,不能鲁棒地处理这种情况。Comaniciu<sup>[7]</sup>也提出结合目标的颜色和梯度的方法,并成功应用在 Mean Shift 人脸跟踪中。但是这些方法在目标和背景颜色差异大时有效,当目标和背景的相似程度较大或者目标运动快速等复杂运动时,颜色信息就不足以表示目标模式。因此,如何将目标的多个不同的视觉特征以一种自适应的方式嵌入到 Mean Shift 算法框架中,实现复杂场景和目标变化下的鲁棒跟踪,仍然是一个有待深入研究的课题。

考虑到目标的纹理通常不会受到光照和背景颜色的影响,是较为稳定的特征,而且当跟踪目标和背景相似程度大时,它们一般仍具有不同的纹理,因此把纹理特征融合到目标模型中表示目标是很好的选择。之前常用的纹理模型有灰度共生



矩阵和小波纹理模型等,但是它们不是点样本估计,很难融合颜色等其它特征,而近年来流行的 LBP 纹理模型与颜色模型一样是基于点样本估计的,而且具有尺度不变性、旋转不变性和计算复杂度低等优点,这使它成为当前主流的纹理模型,也有越来越多的学者将其应用到视频跟踪中。例如:Nguyen<sup>[53]</sup>等就把 LBP 纹理模型和图像的灰度特征模型结合起来组成二维的目标模板,并将该模板应用到 Mean Shift 视频跟踪中,结果证明该方法对单色温度录像仪所录的视频序列具有较好的跟踪结果,但如果目标灰度受噪声干扰或与背景难以区分时还是容易丢失目标。王永忠等<sup>[25]</sup>人把目标的颜色特征和 LBP 简单纹理特征分别作为 Mean Shift 跟踪算法的模板,并利用 Fisher 准则对两种特征进行了可分性分析,以调节各个模板的系数。结果证明该方法能对单目标稳定跟踪,但计算复杂度大,实时性较差。Ning<sup>[54]</sup>等通过分析 LBP 的纹理模型各个模式,选取 LBP 纹理模型中的 5 个关键模式与颜色特征中的红色分量、绿色分量共同组成了三维的目标模型,并将该模型嵌入到 Mean Shift 跟踪算法中。实验证明该方法具有很好的实时性,并能实现稳定跟踪。

本章主要介绍了颜色和纹理结合的目标模型表示方法,并将该方法嵌入到 Mean Shift 跟踪算法中实验。章节安排如下:3.2 节介绍改进的 LBP 纹理模型,3.3 节介绍了颜色-纹理结合的目标模型表示方法,并给出了改进后的实验结果和结果分析,3.4 本章小结。

### 3.2 改进的 LBP 纹理模型

LBP(Local Binary Pattern),即局部二值模式,是一种点样本纹理估计方法,是当前纹理研究中的热点。LBP 纹理模型具有灰度和旋转不变性且计算复杂度低等优点,在工业可视化检测、图像检索、场景分析和人脸识别等领域中都得到了广泛的应用<sup>[55,56]</sup>。

点的纹理通常是通过该点灰度值与其邻域点的灰度值做减法运算而得,对于坐标不是正式位置的采样点的灰度值,可以通过双线性插值法得到。图像的 LBP 纹理特征计算公式如下<sup>[56]</sup>:

$$LBP_{p,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (3-1)$$

其中:  $s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$  为阈值函数,  $R$  是中心像素与其邻域像素的距离,反映了

纹理在空间上的分辨率。 $P$  为邻域像素的数目,  $g_c$  表示中心点的灰度值,  $g_p$  表示以  $g_c$  为中心点、以  $R$  为半径的圆环上的第  $p$  个等分点的灰度值。

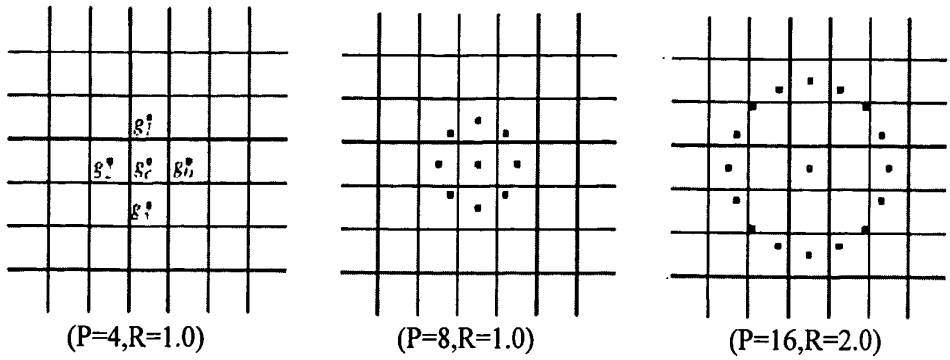


图 3.1 P、R 取不同值的 LBP 模式

上图 3.1 就显示了三种  $P$  和  $R$  取不同值时的 LBP 模式。不同的  $P$  和  $R$  值对应着不同的局部二值模式，当  $P$  较大时，LBP 的计算复杂度将会快速增加。考虑到时视频跟踪的实时性要求，本文选择  $P=8, R=1$ ，此时， $LBP_{8,1}$  纹理模式的值共 256 种，值的范围是 0~255。下图举例显示了  $LBP_{8,1}$  的纹理模型，根据式(3-1)得到该纹理模型的 LBP 值为 205。

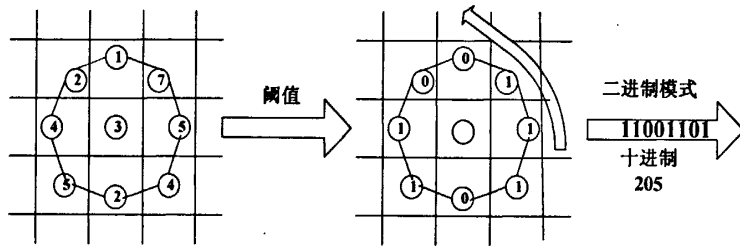


图 3.2  $LBP_{8,1}$  纹理模型示意图

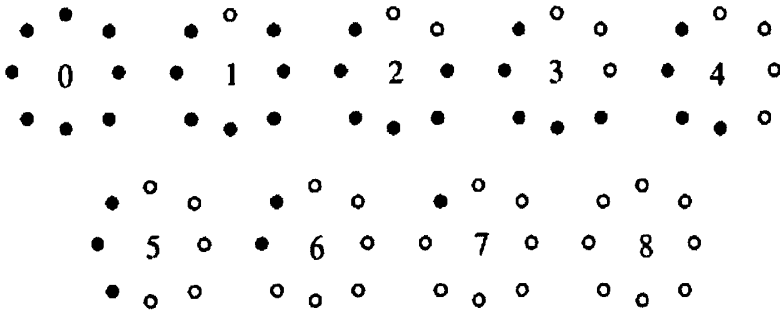
从上图和式(3-1)中可以看出，LBP 值受  $P$  开始位置的影响特别大。在图像有轻微的旋转时，LBP 值可能相差很大。因此它只是具有尺度不变性，而没有旋转不变性。为了解决这种由旋转带来的纹理统计错误问题，很多研究者提出不同的 LBP 旋转不变性的方法，主要有最小值法和 Uniform 模式。

Uniform 模式是当前主流的方法。文献[56]中对上述二值模式按一定的规律进行旋转，这样可把原来的 256 种纹理模式合并为 36 种。其中，将出现频率最高的 9 种模式称为 Uniform 模式，对应的模式值为 0~8。记为  $LBP_{8,1}^u$ ，具体定义如下：

$$LBP_{8,1}^u = \begin{cases} \sum_{p=0}^7 s(g_p - g_c), & U(LBP_{8,1}) \leq 2 \\ 9, & otherwise \end{cases} \quad (3-2)$$

$$U(LBP_{8,1}) = |s(g_7 - g_c) - s(g_0 - g_c)| + \sum_{p=1}^7 |s(g_p - g_c) - s(g_{p-1} - g_c)| \quad (3-3)$$

式(3-3)表示下图 3.3 种所示的环中各点值 0 和 1 的变化次数。

图 3.3  $LBP_{8,1}^n$  的 uniform 模式

当图像中背景光滑区域的灰度值存在微小的波动, 直接使用式(3-2)计算目标纹理有一定的误差, 并且以上得到的 9 中模式根据出现的频率也有主要次要之分, 如图 3.3 所示: 模式 0 和 1 表示噪音点; 模式 7 和 8 表示暗点和光滑区域; 模式 2 和 6 表示线端; 模式 3 和 5 表示角点; 模式 4 表示边界。一般情况下, 在目标模式的表示中, 边界和角比光滑区域具有更重要的作用。基于此文献[57]和[54]分别针对这两点提出了相应的改进。

为了使 LBP 纹理对微小的灰度波动具有鲁棒性, 在式(3-2)基础上做了改进, 具体改进如下<sup>[57]</sup>:

$$LBP_{8,1}^n = \begin{cases} \sum_{p=0}^7 s(g_p - g_c - a), & U(LBP_{8,1}) \leq 2 \\ 9, & otherwise \end{cases} \quad (3-4)$$

$$U(LBP_{8,1}) = |s(g_7 - g_c) - s(g_0 - g_c)| + \sum_{p=1}^7 |s(g_p - g_c) - s(g_{p-1} - g_c)| \quad (3-5)$$

其中,  $a$  是关键, 它表示光滑区域灰度波动的阈值, 本文选择  $a$  的值为 3。

在式(3-4)和(3-5)的基础上提取了 9 中 Uniform 纹理模式中的主要的 5 种。代表图像光滑区域和噪音点的模式 0、1、7 和 8 属于目标模式的次要部分, 而与目标边界和角有关的模式 2、3、4、5 和 6 构成了目标的主要纹理边界。这 5 种基本模式不仅提取目标的边界纹理, 而且同时有去除了光滑背景和抑制噪音的能力。最终得到的纹理模型<sup>[54]</sup>为:

$$LBP_{8,1}^n = \begin{cases} \sum_{p=0}^7 s(g_p - g_c - a), & U(LBP_{8,1}) \leq 2 \text{ 且 } \sum s(g_p - g_c - a) \in \{2, 3, 4, 5, 6\} \\ 0, & otherwise \end{cases} \quad (3-6)$$

本文是用式(3-6)构造的纹理模型。下面介绍颜色和纹理结合的目标表示方法。

### 3.3 颜色-纹理结合的目标模型表示

由于光照及目标的运动等原因, 在目标跟踪过程中目标与背景时刻在发生变化, 目标跟踪应该适应这种变化以增强目标跟踪的鲁棒性。近年来, 许多研究者注意到在目标跟踪中进行自适应特征选择与融合的重要性。

传统的基于核的跟踪方法中, 利用多特征描述目标的方法是直接将目标的多个特征组合在一起, 并在目标的高维特征空间用其特征直方图来描述目标的。但是该方法最大的缺点就是随着特征空间维数的增加, 运算复杂度会急剧增加而使实时跟踪变的不可行。因此, 使用多个线索的同时如何更高效的融合不同线索的问题吸引了不少学者的关注。

目前, 主流的多线索融合的方法是将多个线索分别开来, 然后通过一种融合策略将其结合起来。常用的融合策略有加权融合<sup>[25, 58]</sup>和乘性融合<sup>[49, 59]</sup>。

#### 3.3.1 加权融合

加权融合是基于对不同的场景和不同的线索的可靠性的不同, 通过对不同线索给出的评价值加权的方法来调整对它们的依赖程度, 而权重调整的依据是单个线索给出的结论与所有线索给出的综合结论的差别大小。因此, 加权融合方式仅对各个线索按照一定的线性加权, 它并不放大噪声, 但同时场景和目标复杂变化的情况下并不能较大程度上提高融合跟踪的可信度, 不便于长时间使用。

文献[25]中王永忠等人把目标的颜色特征和 LBP 简单纹理特征分别作为 Mean Shift 跟踪算法的模板, 利用目标特征子模型集合构造了目标的多特征描述, 通过线性加权的方法将目标的多个特征集成在核跟踪方法中, 而且根据各个子模型与当前目标及背景的相似性, 提出了一种基于 Fisher 可分性度量的权值自适应更新机制, 实验结果证明改进的融合方法可以在光照不断变化的情况下能对单目标稳定跟踪, 但计算复杂度大, 实时性较差。

#### 3.3.2 乘性融合

乘性融合是将特征权值以乘积的形式给出多个特征的联合分布情况, 它假设各个线索间独立从而方便的构造出多线索的联合似然分布。但事实上这一假设并不是总成立的, 不过这一假设简化了问题本身。从贝叶斯观点来看的融合结果是最好的, 因此乘性融合得到了广泛的应用, 它可以有效的提高跟踪的精度, 但是很容易受噪声的影响, 会放大噪声。

两种融合策略各有优劣, 因此有些人就将两种策略结合应用。文献[60]中, 李等人将这两种融合策略结合, 在粒子滤波过程中采用乘性融合放大有效粒子的

权值以保证大权值粒子的数量,防止粒子退化,在对各粒子的均值迁移迭代的过程中采用加权融合的策略抑制噪声,该方法得到了稳定的跟踪结果,但是很明显,文中的方法计算复杂度较高,而且该方法只在静态场景中实验成功,动态复杂场景中的效果无从得知。

还有一种跟踪方法是基于协方差矩阵的跟踪方法,它是2006年F. Porikli<sup>[61]</sup>等人提出的,应用基于协方差的目标描述。它是综合多个特征的跟踪方法,可以认为是多特征的融合方法,但是关键的一点是模型匹配问题。模型间的相似准则几篇文献中是用李群结构、黎曼几何等方法求的,这种方法的时间复杂度较高,不能达到实时跟踪的要求。

### 3.3.3 本文融合方法

从上文介绍的两种融合策略可以看出,加权融合并不能提高融合跟踪的可信度,而乘性融合虽然可以提高跟踪性能,但是容易受噪声影响,而且假设各个线索独立。因此,基于多线索融合跟踪方法并不成熟,没有很好的融合策略达到既提高跟踪可信度,同时具有稳定跟踪的性能。

本文方法还是利用基本的将多个特征直接结合在一起的方法,但该方法不但没有增加特征空间的维数,反而降低了计算复杂度,用更小的特征空间维数描述目标,即提高了跟踪的可信度也加快了跟踪速度。

上面也提到,肤色能够很好的被聚类并可以很好的与其他颜色区分,所以被很多的研究者用于目标跟踪中。然而利用肤色的方法对光照条件敏感,也无法处理类似肤色的背景区域。而纹理信息恰好对光照的敏感性低,通常不会受到光照和背景颜色的影响,是较为稳定的特征,而且当跟踪目标和背景相似程度大时,它们一般仍具有不同的纹理,本文就结合颜色特征和纹理特征共同描述目标模型。

颜色特征,本文利用RGB颜色空间中的R分量和G分量,去掉了对光照敏感的B分量,按照直方图的方式将R和G分量分别分为8个Bins,然后加上利用式(3-6)得到的LBP纹理模型,共同组成了三维的特征空间,特征数量是 $8*8*5$ 共320个,而一般的颜色空间是 $16*16*16$ 共4096的特征数。就特征数量上而言,本文方法是RGB颜色直方图表示法的7.81%<sup>[54]</sup>。下节的实验结果证明,这样少量的特征数量表示的目标模型仍能得到更好的跟踪效果。

### 3.3.4 改进后的 Mean Shift 算法步骤

本章同样将颜色纹理结合的目标模型嵌入到基本的Mean Shift算法中去,假定初始帧中手动选取目标区域,并获得颜色纹理结合的目标模型为 $\{\hat{q}_\beta\}_{\beta=1,\dots,m}$ (步骤中所有的目标模型都是基于颜色纹理结合表示的),对上章中的算法步骤略作改进,可以获得如下的目标跟踪定位算法步骤:

- 1) 将上一帧目标的位置定为当前帧目标的初始位置  $y_0$ , 利用公式(2-20)计算当前帧的目标候选直方图  $\{\hat{p}_\beta(\hat{y}_0)\}_{\beta=1,\dots,m}$  和 Bhattacharyya 系数

$$\hat{\rho}(y_0) = \rho[\hat{p}(y_0), \hat{q}] = \sum_{\beta=1}^m \sqrt{\hat{p}_\beta(y_0) \hat{q}_\beta}$$

- 2) 利用公式(2-26)计算图像像素的加权  $\{\omega_i\}_{i=1,\dots,n_k}$ , 利用(2-27)计算候选目标的下一个新位置  $y_1$ ;
- 3) 循环判断  $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$ , 若满足则  $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_0 + \hat{y}_1)$ ;
- 4) 判断循环终止条件  $\|\hat{y}_1 - \hat{y}_0\| < \varepsilon$ , 若满足则终止循环, 否则  $\hat{y}_1 \leftarrow \hat{y}_0$ , 并返回第 2)步继续寻找满足条件的候选目标位置;

### 3.4 实验结果及分析

视频 seq-dk 和实验环境同上一章一致, 本实验将基本 Mean Shift 与本章加入颜色纹理结合的目标模型的改进后的 Mean Shift 方法进行比较。下图是两种方法比较的结果显示。

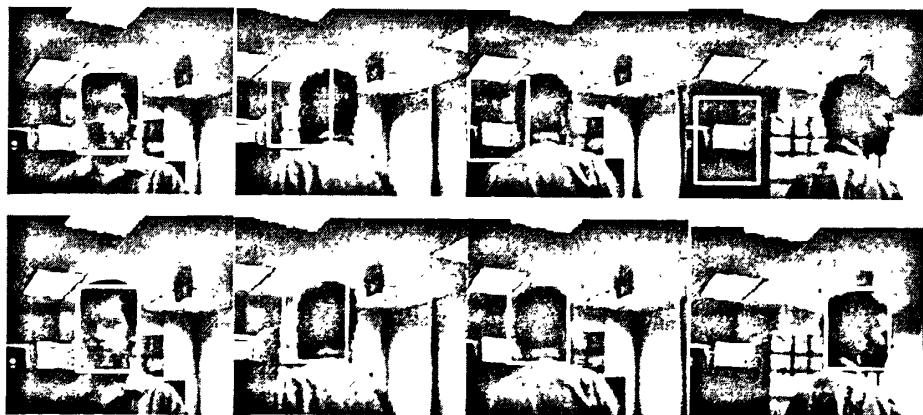


图 3.4 两种方法的结果, 分别对应视频的 1、38、43、50 帧

上行是基本 Mean Shift 方法的结果, 下行是加入颜色纹理结合的目标模型表示的改进方法后的结果。首帧是手动选定的, 两种方法首帧目标区域近似, 基本方法从 38 帧开始丢失目标, 直到后来完全丢失, 再也没找回目标, 而改进后的方法在人脸旋转不可见的情况下照样可以很好的跟踪目标到最后。

同样, 实验结果证明颜色纹理结合的目标模型表示的改进方法对人脸旋转变化情况下的跟踪也是很有效的, 这得益于加入的纹理模型。当人脸旋转幅度过大

甚至人脸不可见的情况下，虽然颜色模型发生很大的变化，但是加入纹理信息后，颜色和纹理共同决定目标模型的情况下，Mean Shift 方法仍然可以很好的跟踪到头部，以便在人脸转回正面后可以马上准确的定位人脸。

### 3.5 本章小结

本章首先介绍了改进的 LBP 纹理模型，然后叙述了颜色和纹理结合的目标模型表示方法，最后给出基于颜色和纹理结合的目标模型改进 Mean Shift 方法的实验结果及分析。

实验结果表明，在视频序列的人脸跟踪中，颜色和纹理结合的目标模型比单一的颜色模型更为有效和鲁棒，在人脸大幅度旋转甚至人脸不可见的情况下，人脸目标颜色发生很大变化时仍然可以利用纹理信息准确的跟踪到人的头部，在人脸转为正面后也能马上准确的定位人脸目标。

## 第四章 自适应调节核窗大小的 Mean Shift 算法

在传统的 Mean Shift 算法跟踪过程中,核窗宽的大小是固定不变的,但是被跟踪的目标大小却可能时刻变化着。核窗宽的大小不但决定了参与 Mean Shift 迭代的样本的数量,而且也反映了跟踪窗口和目标的大小,因此它在 Mean Shift 跟踪中也起了非常重要的作用。

在目标运动中,目标的姿态大小、环境的光照都存在变化,原始的 Mean Shift 算法缺乏必要的模板更新算法,利用本文的模板更新算法时,如果核窗宽大于目标的大小,会对目标过大估计,在进行目标模板更新时就会包含进背景,产生目标模型更新误差,进而影响跟踪性能。此外,当目标的尺寸大小超过核窗宽范围时,固定不变的核窗宽常常会对目标过小估计,导致目标定位不准甚至跟踪失败,也同样会影响更新的目标参考模板。因此我们需要在跟踪过程中的不断调整核函数的带宽大小,使之适应目标尺度变化,尽可能保证被跟踪目标始终在核窗范围内。准确的估计目标的大小也是目标跟踪中一个关键的问题。

### 4.1 自适应调节核窗大小

近年来,同样也有许多的研究者在研究目标尺度更新问题,主要的方法有以下几种:

1) 文献[7]中提到的采用核函数带宽重定位来适应目标尺度的变化。设前一帧中的目标带宽是  $h$ , 使用  $0.9h$ 、 $h$ 、 $1.1h$  的核窗大小进行三次跟踪循环,然后取其中产生最大 Bhattacharyya 系数的一次循环作为最终的跟踪结果。但是这种方法没有从根本上解决目标尺度的问题,而且三次循环增大了算法的复杂度,极大地降低了跟踪的速度,无法达到实时性要求。此外,该算法有尺度收敛过拟合的问题,窗宽有可能越来越小无法正确描述目标大小。

2) 彭宁嵩在文献[62]中证明了目标在不同尺度下核直方图基于 Bhattacharyya 系数相似性的基础上,在核窗宽固定的条件下,目标在其窗宽范围内进行缩放、平移运动并不影响 Mean Shift 跟踪算法空间定位的准确性。在此基础上提出了一种基于后向跟踪、形心配准的核窗宽自动选取的算法,并实验证明算法的有效性。但是对于非刚性的目标来说,目标旋转,形变等变化是经常发生的,因此形心配准的方法就失效了。

3) Collins 在文献[44]中增加一个额外的尺度核,在定义的离散尺度空间进行 Mean Shift 迭代,从而找到最佳的核窗宽,解决了核函数带宽实时变化时目标跟



踪问题, 但计算量增加很大, 算法速度不好。

4) 根据目标的形状或轮廓信息来估计目标的大小。在人脸跟踪中, 椭圆拟合人脸来获得人脸大小时最常用的方法。

总结以上方法, 还是椭圆拟合的方法最快速有效。而 CAMSHIFT 算法中就是利用椭圆拟合的方法对目标尺度变化进行自适应调整的, 该方法简单有效, 但是有一个缺点, 就是容易受周围与目标颜色相似的背景或其它物体的干扰。针对这一缺点, 本文中利用上章的颜色纹理结合的目标模型表示方法对其有一定的限制作用, 此外还用限制长轴和短轴比例的方法来限制目标的大小。

CAMSHIFT 算法是 Bradski<sup>[63]</sup>于 1998 年提出来的, 是计算局部最优解的一个实用算法, 具有快速高效的特点。它在每一帧中寻找目标位置的过程是通过 Mean Shift 算法完成的, 能够自动调节搜索窗的大小和位置, 定位被跟踪目标的中心位置和大小。

本文是在 Mean Shift 算法迭代过程结束后, 最后一次迭代得到的权重图像的基础上利用 CAMSHIFT 中的矩计算估计目标的大小的。具体的算法公式<sup>[63]</sup>为:

a) 在权重图像上, 目标迭代的最后一次位置的核窗区域内, 利用下式计算零阶矩:

$$M_{00} = \sum_x \sum_y I(x, y) \quad (4-1)$$

b) 计算一阶矩和目标中心位置:

$$M_{10} = \sum_x \sum_y xI(x, y), M_{01} = \sum_x \sum_y yI(x, y) \quad (4-2)$$

$$x_c = \frac{M_{10}}{M_{00}}; y_c = \frac{M_{01}}{M_{00}} \quad (4-3)$$

c) 计算二阶矩获得长轴和短轴大小以及目标的方向信息:

$$M_{20} = \sum_x \sum_y x^2 I(x, y), M_{02} = \sum_x \sum_y y^2 I(x, y), M_{11} = \sum_x \sum_y xyI(x, y) \quad (4-4)$$

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}}; w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \quad (4-5)$$

$$\theta = \frac{1}{2} \arctan \left( \frac{2 \left( \frac{M_{11}}{M_{00}} - x_c y_c \right)}{\left( \frac{M_{20}}{M_{00}} - x_c^2 \right) - \left( \frac{M_{02}}{M_{00}} - y_c^2 \right)} \right) \quad (4-6)$$

其中:

$$a = \frac{M_{20}}{M_{00}} - x_c^2, b = 2 \left( \frac{M_{11}}{M_{00}} - x_c y_c \right), c = \frac{M_{02}}{M_{00}} - y_c^2 \quad (4-7)$$

当有与肤色相似的背景或人手等干扰时,利用上述公式计算得到的人脸大小往往比实际的人脸大小大好多,这是 CAMSHIFT 的弱点,本文除了有核窗大小限制这种情况发生外,还采取了一个措施,即通过限制上述的长轴和短轴的比例来达到限制目标大小过大估计。本文的实验数据中有大量人脸旋转和侧面等情况,人脸的长宽比例变化较大,因此限制人脸长宽比例的范围也相对较大。假设在首帧选定目标区域时获得的人脸目标的长宽比例为  $H$ ,则本文设定在跟踪过程中,人脸目标的长宽比例为  $0.8H \sim 1.5H$ ,当长宽比例超过这个范围时,则人为的调整核窗大小为上一帧的核窗大小。

通过自适应调节核窗不仅可以减少搜索的迭代系数,加快跟踪的速度,而且还可以有效的防止类肤色物体或背景对跟踪效果的干扰,提高跟踪的准确性。

#### 4.2 综合方法的 Mean Shift 算法步骤

综合上两章和本章的改进方法,将自适应目标模型更新、颜色-纹理结合的目标模型表示和自适应调节核窗宽方法全部嵌入到 Mean Shift 算法中,假定初始帧中手动选取目标区域,并获得颜色纹理结合的目标模型为  $\{\hat{q}_\beta\}_{\beta=1\dots m}$  (步骤中所有的目标模型都是基于颜色纹理结合表示的),上一帧的最终的目标模型  $\{\hat{r}_\beta\}_{\beta=1\dots m}$  可以获得如下的目标跟踪定位算法步骤:

- 1) 将上一帧目标的位置定为当前帧目标的初始位置  $y_0$ , 将上一帧目标大小乘以一个固定因子作为该帧中的目标跟踪带宽,利用公式(2-20)计算当前帧的目标候选直方图  $\{\hat{p}_\beta(\hat{y}_0)\}_{\beta=1\dots m}$  和 Bhattacharyya 系数

$$\hat{\rho}(y_0) \equiv \rho[\hat{p}(y_0), \hat{q}] = \sum_{\beta=1}^m \sqrt{\hat{p}_\beta(y_0) \hat{q}_\beta}$$

- 2) 利用公式(2-26)计算图像像素的加权  $\{\omega_i\}_{i=1\dots n_i}$ ;
- 3) 计算核内所有像素的权重的统计平均值,当它大于一定阈值(本文给定经验值),则利用(2-27)计算候选目标的下一个新位置  $y_1$ , 否则利用式(2-34)、和  $\{\hat{r}_\beta\}_{\beta=1\dots m}$  重新计算各像素的权重和候选目标的新位置  $y_1$ ;
- 4) 循环判断  $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$ , 若满足则  $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_0 + \hat{y}_1)$ ;

- 5) 判断循环终止条件  $\|\hat{y}_1 - \hat{y}_0\| < \varepsilon$ ，若满足则终止循环，否则  $\hat{y}_1 \leftarrow \hat{y}_0$ ，并返回第 1) 步计算当前位置的目标候选模型继续寻找满足条件的候选目标位置；
- 6) 确定最终的目标位置后，利用(4-3)和(4-5)计算目标的中心位置和大小，在该位置和大小核窗上利用公式(2-17)更新当前帧的目标模型  $\{\hat{y}_\beta\}_{\beta=1\dots m}$ ，以备后用。

### 4.3 跟踪算法评估指标

跟踪算法的性能评估是计算机视觉中必不可少的重要环节，目前国际上多数研究都是针对算法准确性的，对算法的鲁棒性、灵敏度和稳定性的评估的研究几乎没有。本文也是针对算法准确性进行评估，主要采用 Viper 平台对算法进行评估。

Viper 平台是一个可重新配置的视频性能评估的软件，由美国 Maryland 大学的 LAMP (Language and Media Processing) 实验室开发的。其中的软件包提供一个可以生成基准数据 (Ground Truth) 的界面、算法性能评估的准则和使结果可视化的工具。本文用到的基准数据就是通过 Viper 手动标定产生的，下面详细叙述一下算法性能评估的准则。

算法性能评估是将算法的输出结果与基准数据从不同的评价指标进行比较，基准数据被认为是先验的正确的目标属性的集合，该评估的流程如下图所示：

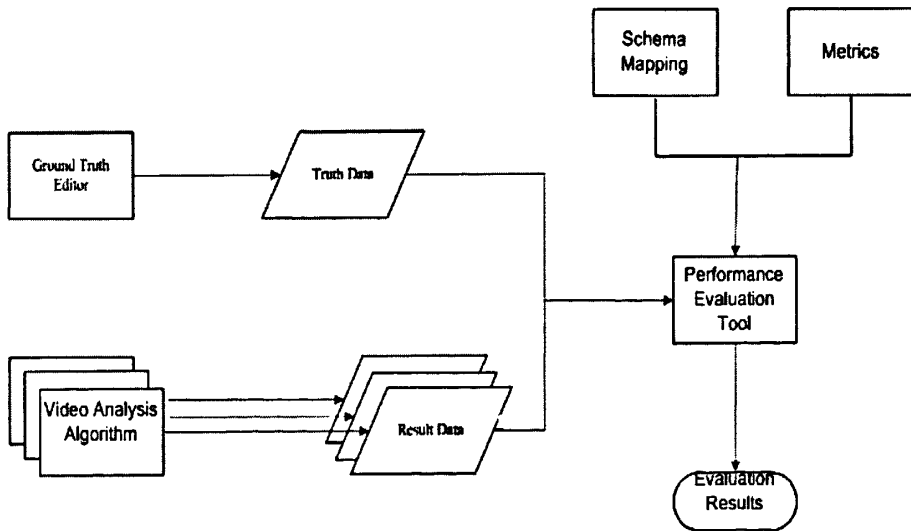


图 4.1 算法性能评估流程

在后续公式中， $G(t)$  代表第  $t$  帧的基准标定， $D(t)$  表示第  $t$  帧的算法输出框， $U_{G(t)}$ 、 $U_{D(t)}$  分别表示  $G(t)$ 、 $D(t)$  的空间并集，针对本文是单目标的视频人脸跟踪，

主要用到的评估指标有以下几种。

### 1. 戴斯系数

该指标是一种对离散数据的相似性度量的指标，例如用于信息重建中的关键字集合  $X$  和  $Y$ ，两者的戴斯系数可以被定义为：

$$S = 1 - \frac{2|X \cap Y|}{|X| + |Y|} \quad (4-8)$$

在实际图像应用中，戴斯系数可以很好的对目标和候选目标进行识别、帧检测和目标数目统计等需求分析。比如计算一帧中的目标数目时，不需要核对这些目标所处位置，戴斯系数就可以满足应用的需求。戴斯系数越小表示离散数据的相似度越高。该指标求解方法如下图所示：

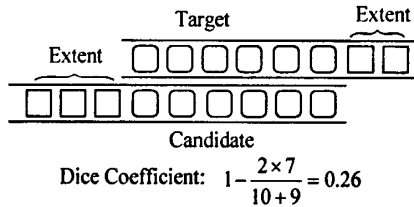


图 4.2 戴斯系数求解图示

### 2. 基于区域的查全率 (Object Area Recall)

该指标是基于对区域像素点数目进行统计的指标，度量算法得出检测区域与基准区域重合度的好坏。对于某一帧  $t$ ，定义  $Rec(t)$  为基准区域中被检测到的区域和整个基准区域的比值：

$$Rec(t) = \begin{cases} \text{undefined} & \text{if } U_G(t) = 0 \\ \frac{|U_D(t) \cap U_G(t)|}{|U_G(t)|} & \text{otherwise} \end{cases} \quad (4-9)$$

OverallRec 是整个视频序列中所有帧的  $Rec$  的加权平均值。

$$OverallRec = \begin{cases} \text{undefined} & \text{if } \sum_{i=1}^{N_f} |U_G(t)| = 0 \\ \frac{\sum_{i=1}^{N_f} |U_G(t)| * Rec(t)}{\sum_{i=1}^{N_f} |U_G(t)|} & \text{otherwise} \end{cases} \quad (4-10)$$

其中， $N_f$  表示基准数据的总帧数， $||$  运算符表示取这一区域的像素点数。

### 3. 基于区域的查准率 (Box Area Precision)

该指标也是对区域像素点数目进行统计，查准率越高则不属于目标的像素点越少。对一单帧  $t$ ，定义  $Prec(t)$  为算法得到检测区域和基准标定区域的重合区域

与总的算法检测区域的比值:

$$Prec(t) = \begin{cases} \text{undefined} & \text{if } U_D(t) = 0 \\ \frac{|U_D(t) \cap U_G(t)|}{|U_D(t)|} & \text{otherwise} \end{cases} \quad (4-11)$$

同上, OverallPrec 是整个视频序列中所有帧的 Prec 的加权平均值:

$$Overall\ Prec = \begin{cases} \text{undefined} & \text{if } \sum_{i=1}^{N_f} |U_D(t)| = 0 \\ \frac{\sum_{i=1}^{N_f} |U_D(t)| * Prec(t)}{\sum_{i=1}^{N_f} |U_D(t)|} & \text{otherwise} \end{cases} \quad (4-12)$$

其中,  $N_f$  表示基准数据的总帧数,  $\|\cdot\|$  运算符表示取这一区域的像素点数

#### 4. 成功定位目标个数的查全率 (Localized Object Area Recall)

如果一个基准目标被输出框覆盖的区域超过了一个比例阈值, 则其被视为检测到。定义  $Loc\_Obj\_Recall(t)$  是第  $t$  帧中被检测到的目标的数目。

$$Loc\_Obj\_Recall(t) = \sum_{G_i(t)} ObjDecect(G_i(t)) \quad (4-13)$$

其中:

$$ObjDecect(G_i(t)) = \begin{cases} 1 & \text{if } \frac{|G_i(t) \cap U_D(t)|}{|G_i(t)|} > OverlapMin \\ 0 & \text{otherwise} \end{cases} \quad (4-14)$$

$OverlapMan$  是判断条件成立的最小阈值,  $Overall\_Loc\_Obj\_Recall$  是所有帧中被检测到的目标数目与总的基准目标数的比值:

$$Overall\_Loc\_Obj\_Recall = \frac{\sum_{i=1}^{N_f} Loc\_Obj\_Recall}{\sum_{i=1}^{N_f} N_G(t)} \quad (4-15)$$

综上, 指标 1 是将跟踪过程中的目标位置信息与基准进行数据上的相似性比较, 指标 2 和 3 把基准中的像素点看做目标或非目标, 把算法输出结果中的像素点看作检测到或未检测到, 指标 4 则对已测量的区域的查准率和查全率加了一个阈值来决定基准目标是否被检测到和检测区域是否准确的覆盖了基准区域。

## 4.4 算法的综合性实验及性能比较

本文将上几章中介绍的改进方法应用到人脸跟踪过程中, 采用了 MATLAB R2010a 在由斯坦福大学的视觉实验室提供的专门为检测人脸跟踪算法的人脸序列上进行人脸跟踪。视频中包含人脸姿态变化、人脸快速运动、人脸被遮挡、人脸大小不断变化等情况。本文中颜色直方图的目标模型的构造是利用 RGB 颜色空间, 特征数量为  $16*16*16$ , 而颜色-纹理结合的目标模型直方图是利用 R、G 和 LBP 共同构成, 特征数量为  $8*8*5$ 。基于以上的改进方法, 实验中用以下四种实验方法进行性能比较。

- A. 基本的 Mean Shift 跟踪算法: 目标参考模型固定不变; 单一的颜色特征构建目标模型; 核窗宽大小固定不变;
- B. 自适应更新目标模型的 Mean Shift 跟踪算法: 加入自适应更新目标模型; 单一颜色特征构建目标模型, 核窗宽大小固定不变;
- C. 颜色-纹理结合的 Mean Shift 跟踪算法: 目标参考模型固定不变; 颜色-纹理结合的目标模型; 核窗宽大小固定不变;
- D. 综合的改进 Mean Shift 跟踪算法: 自适应更新目标模型; 颜色-纹理结合的目标模型; 自适应调节核窗大小。

### 4.4.1 单个改进方法的性能比较

本节实验分别将方法 B 和 C 与 A 进行比较, 说明各个改进方法的有效性。利用到两个视频, 视频一 seq-dk 是第二、三章用过的视频数据, 各章节中已经显示了结果图, 以下主要用 viper 评估各个改进算法, 分别获得各方法与基本算法的性能比较。视频二 seq-dp 共 95 帧, 帧率为 16 帧/S, 除了包含人脸目标的旋转和不可见情况外, 还包括人脸目标被其他人脸遮挡的情况, 以下将显示方法 A、B 和 C 的图示结果和用 Viper 评估的参数结果。

#### A. 视频一 seq-dk 上的结果比较:

前面两章的图示结果显示, 在人脸发生旋转甚至不可见时, 方法 A 会跟丢目标, 而方法 B 和 C 都能很好的跟踪目标到最后。下表利用 Viper 各参数显示了各个跟踪方法的准确的跟踪性能及比较。

表 4.1 方法 A、B、C 在视频一上的对比实验结果

视频, 算法		评价 指标	戴斯 系数	目标区域 查准率	目标区域 查全率	成功定位目标 个数查全率	算法平均迭 代次数	算法运 行时间 (S)
seq-dk	A	0.2861	0.7130	0.7183	0.78	4.7	11.2	
	B	0.1178	0.8655	0.9245	0.98	4.5	8.9	
	C	0.0983	0.8845	0.9245	1.0	4.5	7.9	

从图表显示结果可以看出, 不论是从戴斯系数、查全率、查准率、成功定位的查全率, 还是从算法的迭代次数和运行时间上, 单独比较方法 A 和 B、方法 A 和 C, 方法 B 和 C 都比方法 A 有更好的跟踪性能, 因此, 不论是在跟踪过程中利用姿态变化信息来自适应更新目标模型的改进方法, 还是剔出应用特征单一的缺点而加入目标纹理信息的多特征表示的目标模型改进方法, 两者都可以有效的处理人脸目标大幅度旋转甚至不可见的情况, 获得鲁棒且准确有效的跟踪效果。

#### B. 视频二 seq-dp 上的结果比较:

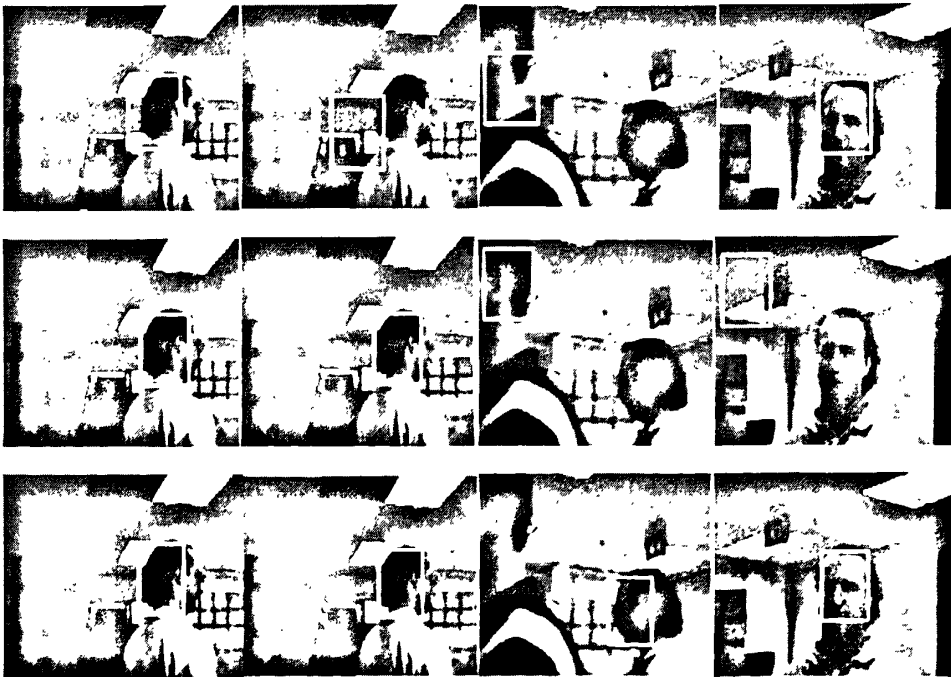


图 4.3 视频二自上至下方法 A、B、C 的结果, 对应视频的 65、66、84、95 帧

从图示结果可以看出: 利用方法 A 在人脸发生旋转后会跟踪失败, 但是基于 Mean Shift 本身的特性, 后来又重新跟上目标; 而方法 B 在人脸发生旋转后仍能很好的跟踪目标, 但是当有其他人脸遮挡被跟踪人脸而导致跟踪失败后无法再重新跟回被跟踪人脸, 这是因为方法二在人脸姿态发生变化后, 人脸目标的参考

模型也自适应的发生改变,当错误的跟到其他人脸目标时,目标参考模型变为该人脸的模型,所以后面无法再跟踪原先的目标,这正是方法 B 的缺陷;方法 C 利用了人脸目标的 LBP 特征,因此,即便有其他人脸干扰被跟踪人脸,当两个人脸分开后,方法 C 还是可以立刻跟踪到正确的人脸目标。下表准确的比较了各方法在视频二上的跟踪性能。结果显示,方法 C 还是从各个方面都比方法 A 优秀,而方法 B 虽然在跟踪失败后无法重新跟踪正确的目标,但从所有视频序列上评价还是优于方法 A 的。

表 4.2 方法 A、B、C 在视频二上的对比实验结果

视频, 算法		评价 指标		目标区域 查全率	成功定位目标 个数查全率	算法平均 迭代次数	算法运 行时间 (S)
		戴斯 系数	目标区域 查准率				
seq-dp	A	0.4259	0.5624	0.6032	0.70	5.8	12.2
	B	0.3819	0.5796	0.6811	0.74	5.2	11.4
	C	0.2414	0.6945	0.8938	0.91	4.7	11.6

#### 4.4.2 综合方法的性能比较

上节实验中,将两个主要的改进方法分别与基本 Mean Shift 算法进行比较,取得了良好的效果,证明了每种改进方法都是有效的。本节实验将综合所有的改进方法与基本的 Mean Shift 算法进行比较,即方法 D 和方法 A 进行比较。实验视频有两个,视频三 seq-fast 共 31 帧,帧率 16 帧/S,视频主要是人脸目标快速运动的情况,视频四 seq-sb 共 533 帧,帧率 16 帧/S,本文选取前 200 帧进行实验,该视频中包含人脸目标旋转、人脸不可见、人脸大小不断变化等情况。以下是两种方法的跟踪结果图示和用 Viper 评估的结果。

##### A. 视频三 seq-fast 上的结果比较:



图 4.4 视频三自上至下方法 A 和 D 的结果,对应视频的 2、3、10、31 帧

从图示结果上可以看出,在人脸目标快速运动时,基本 Mean Shift 算法 A 有



明显的滞后性, 当目标更快速的运动时, 方法 A 很有可能跟丢目标, 而方法 D 不仅可以很好的跟到目标, 而且可以准确的描述人脸目标的大小。从下表的各个参数评估结果也很明显的看出, 虽然每一帧两种方法都可以成功得定位目标, 但是综合改进后的方法具有更加准确和鲁棒的定位。在以上的各个改进方法的结果中, 每种改进方法都缩短了算法的迭代次数和运行时间, 因此, 本文的综合改进方法可以处理目标快速运动的情况, 更准确的跟踪目标。

表 4.3 方法 A、D 在视频三上的对比实验结果

评价 指标 视频, 算法		戴斯 系数	目标区域 查准率	目标区域 查全率	成功定位目 标个数查全 率	算法平均迭 代次数	算法运 行时间 (S)
seq-fast	A	0.1271	0.8370	0.9128	1.0	7.7	7.9
	D	0.0778	0.9530	0.8971	1.0	6.9	6.5

B. 视频四 seq-sb 上的结果比较:



图 4.5 视频四自上至下方法 A 和 D 的结果, 对应视频的 16、25、35、44 帧



图 4.6 视频四自上至下方法 A 和 D 的结果, 对应视频的 58、90、135、179 帧

上述两幅图是比较了方法 A 和 D 的实验结果, 从其中的 25、58 和 135 帧可以看出方法 A 对目标人脸过大估计, 跟踪核窗中包含进很多的背景, 而 90 帧则对目标过小估计, 不能准确的定位目标, 这两种情况都是跟踪带宽固定的缺陷, 而对应的方法 D 可以自适应调节跟踪核窗的大小, 更准确的描述目标, 得益于椭圆拟合人脸来自适应调节核窗的方法。此外, 在目标人脸发生旋转后, 方法 A 从 35 帧开始丢失目标, 而方法 D 即便在人脸不可见的情况下也能很好的跟踪目标。这同样说明了自适应模板更新和颜色纹理结合的多特征描述人脸目标模型的改进方法的有效性。

表 4.4 方法 A、D 在视频四上的对比实验结果

视频, 算法		评价指标	戴斯系数	目标区域查准率	目标区域查全率	成功定位目标个数查全率	算法平均迭代次数	算法运行时间 (S)
seq-sb	A		0.2941	0.8479	0.6332	0.92	4.3	29.5
	D		0.1587	0.8562	0.8397	1.0	4.2	19.8

上表也同样从各个方面评估比较了两种方法的跟踪性能, 结果显示在人脸目标发生旋转、人脸不可见和人脸大小不断变化的情况下, 综合的改进方法在各项指标上都优于基本的 Mean Shift 方法。

综合以上所有实验, 自适应更新目标模型方法、颜色-纹理结合目标模型表示方法和自适应调节核窗大小的方法都能有效的改进 Mean Shift 算法的跟踪性能, 在跟踪定位的准确性和算法的迭代系数和运行时间上都有很好的改善。

#### 4.5 本章小结

本章首先介绍了利用 CAMSHIFT 椭圆结合来自适应调节核窗宽大小的方法, 然后介绍了 Viper 软件评估跟踪算法所用的评估参数, 最后针对本文的各种改进方法列举了四种实验方法, 通过实验得到各种改进方法的性能比较。

实验结果证明, 每种改进方法都可以提高算法的跟踪性能, 将各种方法综合到一起也可以很好的跟踪大小和姿态不断变化的人脸, 改进方法不仅提高了跟踪的准确性, 而且缩短了算法的迭代次数和运行时间, 在一定程度上提高了算法的跟踪速度。



## 第五章 总结与展望

### 5.1 总结

近年来,随着计算机技术的快速发展和人类物质水平的提高,数字视频监控技术日益成熟并得到了广泛的应用。目标跟踪技术是智能视频监控中关键和核心的研究课题之一,而其中的人脸跟踪技术的研究也同样具有十分重要的现实意义。

本文对动态复杂背景下人脸目标的姿态、大小等快速变化的情况下的 Mean Shift 人脸跟踪作了比较深入的研究,取得了相应的研究成果,主要贡献如下:

1. 针对 Mean Shift 跟踪算法中目标参考模型固定不变的缺点,提出了自适应更新目标模型的改进算法,使得目标的参考模型随着动态背景和目标的变化而变化,实验结果表明,在人脸目标姿态和大小动态变化,即便人脸不可见的情况下,本文方法仍能有效的跟踪到人脸目标。
2. 针对 Mean Shift 算法中仅仅使用单一的颜色信息丢失目标空间信息等其它特征,而在光照、人脸姿态和背景变化等引起目标颜色变化大时无法鲁棒跟踪人脸目标的缺陷,引入颜色和纹理结合的目标模型表示方法,实验证明,该方法更为鲁棒的表示人脸目标,在背景和人脸复杂变化的情况下也能快速有效的跟踪人脸目标。
3. 针对 Mean Shift 算法在跟踪过程中核窗宽大小固定不变,无法适应人脸目标的大小变化而可能导致目标定位不准的缺点,利用人脸椭圆的形状进行椭圆拟合获取人脸目标的准确的大小,更准确的更新目标模型和跟踪带宽,增强了跟踪算法的准确性和鲁棒性。

综上所述,本文较全面的分析了 Mean Shift 跟踪算法及其缺陷,并针对各种缺陷提出相应的改进方法,取得相应的效果,为下一步的工作打下良好的基础。

### 5.2 未来工作展望

本文虽然取得了一定的成果,但是,同时也存在着一些不足,需要在接下来的研究工作中加以研究和改进。

首先,本文主要针对单个人脸目标进行跟踪,而现实世界中,需要同时跟踪的目标往往不止一个,其次,本文用人脸椭圆拟合的方法获得人脸目标大小,方法中用的矩的计算,提高了算法的计算复杂度,降低了算法的速度,这一点也有待改进,再者,本文并未在有强烈光照变化的视频上进行实验,不知该算法是否

对强烈的光照变化同样具有鲁棒性，而这也是下一步工作的重点。相信随着科技的发展和相关领域研究的不断深化，越来越多的学者将会对人脸等目标的跟踪技术投入更多的关注，可以预见，在不久的将来，人脸等目标跟踪技术将会得到长足的发展。

## 致 谢

值此论文成稿之际，我的研究生生活即将结束，回首两年多的实验室生活的点点滴滴，我从中不仅学到了导师们严谨的治学态度，还感受到了同学们的拼搏向上，不屈不挠的钻研精神。

我想首先感谢我的导师田捷教授和梁继民教授！他们广博的专业知识，丰富的实践经验，平易近人的处世作风给我留下了深刻的印象。还要特别感谢赵恒老师，从我的论文的构思，实验结果的验证到最后论文的完成都给予了无微不至的指导，并提出了许多宝贵的意见。借此机会，向老师们的苦心栽培致以深深的谢意！

感谢智能生物信息中心胡海虹老师等，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。

感谢实验室的诸位同窗，与我一起度过这段难忘的时光，他们是邹亚茹、崔佳、简小明、唐余波、汪小龙、宋建强、杨华春、毕萍、张春晖、尹雪聪、鲁会娟等。与他们的友好交往增添了我学习的乐趣，有意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。

感谢我的父母，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] 高文, 陈熙霖. 计算机视觉—算法与系统原理. 北京: 清华大学出版社, 2000.
- [2] <http://trend.cnki.net/>
- [3] 李源, 艾海舟. 视频中的人脸检测与头部跟踪. 清华大学综合论文训练, 2005.
- [4] Chuan-xu W., Zuo-yong Li. A new face tracking algorithm based on local binary pattern and skin color information. *Computer Science and computational Technology*, 2008, pp. 657-660.
- [5] Tianrui W., Yuexian Z, Wei W. Skin-Color based Particle Filtering for Human Face Tracking. *Cybernetics and Intelligent Systems*, 2008, pp. 728-733.
- [6] Stern H. Adaptive color space switching for face tracking in multi-colored fighting environments. *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, 2002, pp. 236-241.
- [7] Comaniciu D., Ramesh V. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, pp. 564-575.
- [8] Lerdsudwichai C., Abdel-Mottaleb M. Algorithm for multiple faces tracking. *Proceedings of International Conference on Multimedia and Expo*, 2003.
- [9] Gao J., Wu Z., Wang Y. Face tracking algorithm based on mean shift and ellipse fitting. *ICONIP*, 2006, pp. 270-277.
- [10] Bradski G. Computer vision face tracking for use in a perceptual user interface. *Intel. Technology Journal*, 1998.
- [11] 章毓晋. 图像工程(中册)图像分析, 2版 ed. 北京: 清华大学出版社, 2005.
- [12] Ojala T., Pietikinen M., Harwood D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 1996, vol. 29, pp. 51-59.
- [13] Wu X., Li L., Lai J. A Framework of Face Tracking with Classification Using CAMShift-C and LBP. *IEEE Fifth International Conference on Image and Graphics*, 2009, pp. 217-222.
- [14] 邢昕, 汪孔桥, 沈兰荪. 基于器官跟踪的人脸实时跟踪方法. *电子学报*, 2000, vol. 28, pp. 29-31.
- [15] Strom J. Model-based real-time head tracking. *EURASIP Journal on Applied*



- Signal Processing, 2002, pp. 1039-1052.
- [16] 高建坡. 视频序列中的人脸检测与跟踪算法研究. 东南大学博士学位论文, 2007.
- [17] Bascle B., Blake A. Separability of pose and expression in facial tracking and animation. 2002, pp. 323-328.
- [18] Haritaoglu I., Harwood D., Davis L. Real-time surveillance of people and their activities. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2002, vol. 22, pp. 809-830.
- [19] Decarlo D., Metaxas D. Optical flow constraints on deformable models with applications to face tracking. International journal of computer vision, 2000, vol. 38, pp. 99-127.
- [20] Shen C., Van den Hengel A., Dick A. Probabilistic multiple cue integration for particle filter based tracking. Proc. VIIth Digital Image Computing: Techniques and Applications, 2003, pp. 309-408.
- [21] Seo K., Lee J. Object tracking using adaptive color snake model. 2003, pp. 1406-1410.
- [22] 黄福珍 苏剑波. 基于 Level Set 方法的人脸轮廓提取与跟踪. 计算机学报, 2003, vol. 26.
- [23] Lee H., S. Kim D. Robust face tracking by integration of two separate trackers: Skin color and facial shape. Pattern Recognition, 2007, vol. 40, pp. 3225-3235.
- [24] 庄莉, 徐光祜, 艾海舟. 视频中多线索的人脸特征检测与跟踪. 计算机学报, 2003, vol. 26, pp. 160-167.
- [25] 王永忠, 梁彦. 基于多特征自适应融合的核跟踪方法. 自动化学报, 2008.
- [26] Haritaoglu I. Flickner M., Detection and tracking of shopping groups in stores. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.
- [27] Zhou H., Yuan Y., Shi C. Object tracking using SIFT features and mean shift. Computer vision and image understanding, 2009, vol. 113, pp. 345-352.
- [28] Porikli F., Tuzel O., Meer P. Covariance tracking using model update based on means on Riemannian manifolds. Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, 2006.
- [29] Zheng W., Bhandarkar S. A boosted adaptive particle filter for face detection and tracking. IEEE International Conference on image Processing, 2006, pp. 2821-2824.
- [30] Xu F., Cheng J., Wang C. Real time face tracking using particle filtering and

- mean shift. IEEE International Conference on Automation and Logistics, 2008, pp. 2252-2255.
- [31] Li Y. Tracking in low frame rate video: a cascade particle filter with discriminative observers of different life spans. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, vol. 30, pp. 1728-1740.
- [32] Isard M., Blake A. conditional density propagation for visual tracking. International journal of computer vision, 1998, vol. 29, pp. 5-28.
- [33] Nummiaro K., Koller-Meier E., Van Gool L. An adaptive color-based particle filter. Image and Vision Computing, 2003, vol. 21, pp. 99-110.
- [34] Perez P., Vermaak J., Blake A. Data fusion for visual tracking with particles. Proceedings of the IEEE, 2005, vol. 92, pp. 495-513.
- [35] An K., Yoo D., Jung S. Robust multi-view face tracking. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005, pp. 1905-1910.
- [36] Chang C., Ansari R. Kernel particle filter for visual tracking. Signal Processing Letters, IEEE, 2005, vol. 12, pp. 242-245.
- [37] Yang X., Li H., Zhou X. Nuclei segmentation using marker-controlled watershed tracking using mean-shift and Kalman filter in time-lapse microscopy. Circuits and Systems I: Regular Papers, IEEE Transactions on, 2006, vol. 53, pp. 2405-2414.
- [38] Bai K., Liu W. Improved object Tracking with Particle Filter and Mean Shift. IEEE International Conference on Automation and Logistics, 2007, pp. 431-435.
- [39] Hostetler L. The estimation of the gradient of a density function with applications in pattern recognition. IEEE Transactions on Information Theory, 1975, vol. 21, pp. 32-40.
- [40] Yizong C. Mean shift, mode seeking, and clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, vol. 17, pp. 790-799.
- [41] Comaniciu D. Meer P. Mean Shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, vol. 24, pp. 603-619.
- [42] Comaniciu D. Meer P. Mean shift analysis and applications. IEEE Int. Conference Computer Vision, 1999, pp. 1197-1203.
- [43] 顾凤婵. 基于 Mean Shift 的视频跟踪算法研究. 西安电子科技大学硕士学位论文, 2009.

- [44] Collins R.. Mean-shift blob tracking through scale space. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003.
- [45] 李乡儒, 吴福朝. 均值漂移算法的收敛性. 软件学报, 2005, vol. 16.
- [46] Shieh T., Zhang J., Chiu S. On convergence of the mean shift algorithm. IEEE ISCCSP, 2008, pp. 614-618.
- [47] Comaniciu D., Ramesh V. Real-time tracking of non-rigid objects using mean shift. Proc. Conference on Computer Vision and Pattern Recognition, 2003.
- [48] Soto A. A probabilistic approach for the adaptive integration of multiple visual cues using an agent framework. Robotics Institute, Carnegie Mellon University, 2002.
- [49] Shen C., Van den Hengel A., Dick A. Probabilistic multiple cue integration for particle filter based tracking. 2003, pp. 309-408.
- [50] Yoo T., Oh I. A fast algorithm for tracking human faces based on chromatic histograms. Pattern Recognition Letters, 1999, vol. 20, pp. 967-978.
- [51] Vermaak J. Towards improved observation models for visual tracking: selective adaptation. Computer Vision<sup>†</sup>ECCV 2002, 2002, pp. 645-660.
- [52] Xu X., Li B. Head tracking using particle filter with intensity gradient and color histogram. 2005, pp. 888-891.
- [53] Nguyen Q., Robles-Kelly A. Shen C. Enhanced kernel-based tracking for monochromatic and thermographic video. Advanced Video and Signal Based Surveillance, Sydney, Australia, 2006, p. 28.
- [54] Ning J. Robust Object Tracking using Joint Color-Texture Histogram. International Journal of Pattern Recognition and Artificial Intelligence, 2009, vol. 23, pp. 1245-1263.
- [55] Ojala T. Texture discrimination with multidimensional distributions of signed gray-level differences. Pattern Recognition, 2001, vol. 34, pp. 727-739.
- [56] Ojala T. Pietikinen M., Mäenpää T., Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, pp. 971-987.
- [57] Heikkila M., Pietikainen M. A texture-based method for modeling the background and detecting moving objects. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2006, vol. 28, pp. 657-662.
- [58] Spengler M., Schiele B. Towards robust multi-cue integration for visual tracking. Machine Vision and Applications, 2003, vol. 14, pp. 50-58.
- [59] Li P. Chaumette F. Image cues fusion for object tracking based on particle

- filter. *Articulated Motion and Deformable Objects*, 2004, pp. 99-107.
- [60] 李远征. 基于多特征融合的均值迁移粒子滤波跟踪算法. *电子与信息学报*, 2010, vol. 32.
- [61] Porikli F., Tuzel O., Meer P. Covariance tracking using model update based on lie algebra. 2006, pp. 728-735.
- [62] 彭宁嵩, 杨杰. Mean-Shift 跟踪算法中核函数窗宽的自动选取. *软件学报*, 2005, vol. 16, pp. 1542-1550.
- [63] Bradski G. Real time face and object tracking as a component of a perceptual user interface. 2002, pp. 214-219.

