基于高斯混合模型的背景建模技术研究

摘　　要

运动目标检测是指将图像序列或视频中发生空间位置变化的物体作为前景提取出并进行标示的过程。它是计算机视觉领域里的一个重要课题，随着计算机视觉的发展，它在智能监控、多媒体应用、航天航空等诸多领域的应用也有了更加深远的应用前景。在实际应用中，具体场景存在着动态变化，包括天气、光源、阴影等干扰因素增加了运动目标检测的难度，限制了其精度。为了保证运动目标检测的精度，相应的方法应运而生。

本文介绍了运动目标检测的课题背景、意义、研究现状，介绍了运动目标检测中的的一些基础概念，分析了运动目标检测的常用方法。光流法、帧间差分法和背景减法是常用的运动目标检测方法。光流法计算过于复杂，实时性较差；帧间差分法对动态背景适应性强，但提取运动目标的完整性欠佳；背景减法受动态场景影响较大，但对于运动目标的完整提取效果更好，本文着重分析。

在对运动物体在背景减法的背景建模中，根据背景图像自身的特征，像素峰值比较单一的归类为单模态背景，适用于单高斯模型；像素峰值比较分散的归类为多模态背景，适用于高斯混合模型，在实际应用中多模态背景更为常见。通过设置相应的参数并实时更新，可建立能较好适应动态场景的背景模型。最终将原始图像与背景模型相减，即可将运动目标以所需的精度提取并标示出来。

关键词： 运动目标检测，背景减法，背景建模，高斯混合模型

Research on Background Modeling Technology Based on Gaussian Mixture Model

Abstract

Motion Detection refers to the process of extracting and labeling the object whose spatial position changes in the image sequence or video as the foreground。It is an important subject in the field of computer vision，With the development of computer vision，It also has a farreaching application prospect in many fields, such as intelligent monitoring, multimedia applications, aerospace and so on。in application，There are dynamic changes in specific scenes，Including weather, light source, shadow and other interference factors increase the difficulty of moving target detection，The accuracy is limited。In order to ensure the accuracy of moving target detection，The corresponding methods came into being。

This paper introduces the background, significance and research status of Motion Detection，Some basic concepts of Motion Detection are introduced，The common methods of Motion Detection are analyzed。Optical flow method, inter frame difference method and background subtraction method are commonly used in Motion Detection。The calculation of optical flow method is too complicated，The realtime performance is poor; The inter frame difference method has strong adaptability to dynamic background，But the integrity of extracting moving objects is not good; Background subtraction is greatly influenced by dynamic scenes，But for the complete extraction of moving objects, the effect is better，This paper focuses on the analysis。

In the background modeling of background subtraction for moving objects，According to the characteristics of the background image itself，The pixel peak value is relatively single, which is classified as singlemode background，It is suitable for single Gaussian model; Those with scattered pixel peaks are classified as multimodal background，It is suitable for Gaussian Mixture Model，In practical application, multimodal background is more common。By setting the corresponding parameters and realtime update，It can build a background model which can adapt to the dynamic scene。Finally, the original image is subtracted from the background model，The moving object can be extracted and marked with the required accuracy。

Key Words: Motion Detection, Background Subtraction, Background Modeling, Gaussian Mixture Model

目　　录

1 绪　论 1
1.1 引言 1
1.2 研究背景及意义 1
1.3 国内外研究现状 2
1.4 论文研究内容及结构 4
2 运动目标检测基础及光流法 5
2.1 图像颜色空间 5
2.2 图像预处理 5
2.2.1 图像灰度化 5
2.2.2 图像滤波 6
2.3 光流法 6
3 帧间差法及背景差分法 9
3.1 二帧间差法原理 9
3.2 二帧间差法实验 10
3.3 三帧间差法原理 16
3.4 三帧间差法实验 18
3.5 背景差分法原理 23
4 基于高斯混合模型的运动目标检测 26
4.1 高斯分布基本概念 26
4.2 单高斯模型 26
4.3 高斯混合模型 28
4.4 建立背景模型以进行运动目标检测 32
4.5 简单背景差分法实验 33
4.6 基于高斯混合模型的背景差分法实验 38
结　论 46
致　谢 47
参考文献 48
附　录 49
附录 A 49

1 绪　论

1.1 引言

当今我们所身处的社会，是个科技飞速发展的社会，其中就包括了计算机、网络通信技术等现代化技术的飞跃提升。人类是通过接收并正确识别外界的各种信息来认识和改造世界的，而现如今人类社会已经经过了长期的探索发展，对于信息的处理能力已经有了非常大的进步，对于从图像这一直观的、易于获取并理解消化的信息来源，人们已经可以借助现代计算机科技进行更加深入地分析，获取到原先仅凭肉眼无法获取的更加完整的信息。

随着计算机处理能力的不断提升，计算机对视频、图像的处理能力也不断提高，基于图像、视频处理技术的智能视频监控技术也日益成熟，并广泛地应用于医学、军事、天气预报、交通、安保等诸多领域。对于这项技术，运动目标检测是实现其智能化的重要基础。以智能视频监控技术的应用为例，对于监控获取到的视频图像信息，相较于所得到的背景图像信息，显然前景图像信息，也就是监控拍摄到的例如大型商场等公共场所里走过的各式各样的人、重要交通路口经过的车辆等运动物体所包含的信息是我们更感兴趣，也是我们更加需要的信息。运动目标检测所要实现的功能正是将前景图像，也就是目标运动物体图像从原始图像中与背景图像分离开来，单独提取出来，以用于后续的处理分析，其在包括智能视频监控技术等领域有着广泛应用。由此可见，运动目标检测技术应用广泛，研究意义重大。

现如今，随着人类社会经济飞速发展，生产效率不断提高，经济全球化的进程不断推进，人们对于安保、交通、环境监测等诸多方面的需求日益增加，这也就使得人们越来越依赖于对于监控技术等现代计算机技术的使用。而监控技术的发展至今，要实现智能化，运动目标检测这一学科的发展和完善是必不可少的。

1.2 研究背景及意义

当今我国发展迅速，科技的革新促进了经济的发展，经济的发展反过来推动科技的革新。其中，计算机技术发展迅猛，包括智能视频监控、智能天气识别等技术的智能化也在不断提高。随着生产力的不断发展，人们对生产效率也不断提出了更高的要求，效率低下的技术不断被淘汰，相应地，如基于图像、视频处理技术的智能视频监控技术这

一高度智能化的计算机视觉技术的应用就日益广泛。

传统的视频监控系统在智能化和自动化方面有着严重缺陷，它要求监控人员长时间持续性地盯着监控的显示器屏幕看，用肉眼来对监控探头拍摄到的内容进行观察并分析，再根据人工分析的结果进一步地采取相应措施。显然，传统的视频监控系统过于依赖低效的、繁重的人力工作。一方面，人力劳动不可避免带来的过低的工作效率已经日渐无法满足越来越多需要高效处理的实际情景；另一方面，相较于机械化的自动处理，人工分析存在准确性难以保障的不可避免的风险。比如当单个监控人员需要同时对较多的监控场景进行监控时，首先以人力来应对就存在因一时疏忽而漏掉关键信息的风险，要做到全面地、完整地对所有监控场景进行监控就过于理想化，并且因为是人力劳动，就存在因工作繁重且枯燥导致效率进一步下降的情况[1]。

相应的，智能视频监控系统作为在传统视频监控系统的基础上诞生并发展的一门新技术，有着精确度更高、可靠性更好的优势，并且能够执行人力无法实现的复杂运算，实现对视频图像信号进行复杂处理的功能。以大型商场等人流量较大的场景应用为例，智能视频监控系统能够做到自动对拍摄到的相关人员进行检测与追踪，有选择地进行分析和行为理解，在发生诸如非法行为或其他危险行为时，能及时分析反馈给相关人员，完成应急处理。让计算机进行智能化处理，相较于人力处理，一方面，能更精确地对预设的特殊情况进行识别，能更迅速地作出反应；另一方面，计算机能避免人工服务存在的因疲劳等原因产生的疏漏，保证整个视频监控系统的可靠性[2]。

对于智能视频监控系统等需要智能分析处理视频、图像的技术而言，有一个可靠的运动目标检测算法非常重要。运动目标检测是将所需的前景图像，也就是目标运动物体图像从原始图像中与背景图像分离开来，单独提取出来，以用于后续的处理分析的理论基础。近些年来，运动目标检测技术日渐成熟，并广泛地应用于医学、军事、天气预报、交通、安保等诸多领域。

1.3 国内外研究现状

在上个世纪七十年代，世界各国就已经在智能视频监控领域投入了大量的资源来进行开发和研究；而早从上个世纪六十年代开始，美国许多高校和研究机构就从事对智能视频监控系统中的运动目标进行检测，跟踪，和异常行为识别等领域展开了研究。以美国国防高级研究项目署于 1997 年启动的项目 VSMA(Visual Surveillance And Monitoring)

为例，该监视与监控系统能检测、识别并分析人类的异常行为，且能够通过算法实现对人类的异常行为进行预测，并给出相应的预警，适用于民用场景以及军事战场环境；另外，美国美国国防高级研究署通过机载视频监控(AVS)项目利用运动目标检测技术实时地获取视频中静态或动态运动目标信息，然后再结合运动目标跟踪和识别技术来获得运动目标的速度、位置、轨迹等信息，在无人机的智能视频监控领域取得了进展；ObjectVideo、NICE、IBM、Microsoft、GE 和 UTRC 等外国公司也相继推出了各自在智能视频监控方面的产品，当前一些国际权威期刊如 IJCV、IVC、CVIU 等以及一些重要的学术会议 ICCV、CVPR、ECCV、IWVS 等也都将智能视频监控作为其主题内容之一，为同领域的研究人员提供了更多交流的机会[1,3]。

我国对于智能视频监控领域的起步较晚，直到上个世纪九十年代，国内高校和重点研究院所才逐渐投入对智能视频监控技术的研究。在 2005 年之前，我国的视频监控行业长期处于模拟视频监控时代，自主研发能力严重匮乏，生产效率低下，产品单一落后，过于依赖对于国外同类型产品的进口。现如今，我国的视频监控技术的智能化逐渐成熟，自主创新能力不断提升，越来越多的智能化产品被投入到市场中。以中国科学院自动化研究所自主研发的一套交通监控原型系统为例，它能对从输入设备获取的视频序列进行运动车辆检测，通过对车牌号的检测，可以自动地定位，跟踪和识别车辆，并由解析程序对车辆的跟踪和识别的结果进行分析，对车辆的异常行为进行预测，具有实时性强，能较好地适应动态场景的环境光线变化，对遮挡现象具有很强的鲁棒性的优点。智能视频监控的发展已成为大势所趋，相较于传统视频频控，智能视频监控能利用计算机算法对监控拍摄到的画面进行自动识别、分析处理，对可能存在的异常状况提前发出预警，这完全扭转了传统视频监控技术只能进行事后进行监控回放的被动局面，是视频监控行业技术发展的一次重大革新[3]。

光流法、帧间差分法和背景减法是常用的运动目标检测方法。光流法计算过于复杂，实时性较差，且易受动态背景环境中的噪声干扰；帧间差分法计算速度快，对动态背景适应性强，但提取运动目标的完整性欠佳；背景减法受动态场景影响较大，需要对建立背景模型的参数实时更新以适应动态背景环境如光照等各项条件的变化，但对于运动目标的完整提取效果更好。受限于具体应用场景中复杂多变的环境干扰，运动目标检测在实际应用中会遇到各种各样的困难，国内外研究者对于运动目标检测的算法也在不断更新。例如，帧间差分法从二帧差分法改良到三帧差分法，改善了所得到的运动区域的重

影现象，再对三帧差分法应用内部孔洞填充算法、腐蚀操作及滤波操作来改善运动目标区域存在空洞而不完整的问题；再例如基于高斯混合模型进行背景建模的背景减法，传统的方法不能很好地对高斯混合模型中涉及到的三类需要仅凭借输入的未标记数据点来进行估计的参数进行有效地求解，于是研究人员开发了期望最大算法等算法来尽可能得到各个参数的最优解[4-6]。

1.4 论文研究内容及结构

本篇论文围绕着运动目标检测这一应用需求展开研究。本文首先对视频图像处理的理论基础进行探讨，在图像颜色空间这一理论基础中，RGB 模型是本文研究主要用到的颜色模型，而在图像预处理这一理论基础中，本文研究主要用到了灰度化处理以及中值滤波等滤波去噪手段。接着，本文对光流法、帧间差法、背景减法这三种常见的运动目标检测算法进行研究，其中着重对帧间差法和背景减法进行了实验研究。最后，本文着重对基于高斯混合模型的背景建模技术进行了研究，背景建模算法是使用背景减法进行运动目标检测的核心，本文着重研究了基于高斯分布的背景建模算法，并进行了实验研究。

全文总共分为四个章节，其中每个章节的结构及内容如下：

第一章为绪论，介绍了运动目标检测技术的研究背景和意义、国内外研究现状，以及本文的研究内容及结构。

第二章介绍了视频图像处理的一些理论基础，对图像颜色空间中的 RGB 模型和图像预处理中的图像灰度化处理以及滤波去噪处理进行了研究，并对运动目标检测中的光流法进行了介绍。

第三章对二帧差分法、三帧差分法以及背景差分法这些运动目标检测常用到的方法的原理进行了研究，着重对二帧差分法、三帧差分法做了 MATLAB 实验仿真，并对实验结果进行分析。

第四章介绍了高斯分布的理论基础，研究了单高斯模型和高斯混合模型的原理，着重对基于高斯混合模型的背景建模技术进行了研究，分析了通过运用高斯混合模型进行背景建模过程中的参数更新，对背景差分法在不同场景下的应用进行了 MATLAB 实验仿真，并对实验结果进行分析。

2 运动目标检测基础及光流法

2.1 图像颜色空间

若要使用计算机来对视频图像进行处理，必须先将图像信号量化为数字信号，只有将图像以数据的形式进行描述才能用计算机进行进一步的信号处理，而将图像以数据的形式进行描述的方法就是颜色空间。对每种不同的颜色进行量化得到各个特征量，通过分析这些特征量可以实现对运动目标的检测[3]。

RGB 模型是一种非常典型的颜色模型。RGB 指的是红、绿、蓝三原色，由于所有人类肉眼可以观测到的颜色都是由红、绿、蓝三原色根据不同比例混合而成的，因此 RGB 模型的应用十分广泛。RGB 模型在表示颜色时步骤简单且易于存储，但 RGB 模型也有其缺点，在 RGB 模型中，如需对图像修改亮度或改变颜色，常常要对全部三个分量进行修改，这增加了图像处理的复杂度，并且 RGB 模型对于颜色的空间表示有着与人类肉眼对于颜色的视觉认知冲突的缺陷。本文的视频图像处理研究主要围绕 RGB 模型进行[1-2]。

2.2 图像预处理

在运动目标检测中，需要对图像预先进行灰度化以及滤波去噪，才能有较理想的检测效果。

2.2.1 图像灰度化

在运动目标检测过程中，计算机处理的是视频图像序列。若使用彩色颜色模型表示彩色图像，会大幅度增加计算量。为了保证运动目标检测的实时性，需要将输入的彩色图像转换成为只包含亮度信息的灰度图像进行处理，此过程称为图像的灰度化。

在运动目标检测中，如直接使用 RGB 模型对检测到的图像进行表示，会由于计算量过大而导致运动目标检测很难满足其对于实时性的要求，这是由于 RGB 模型包含了图像的完整颜色信息的缘故。为保证运动目标检测能满足其对于实时性的要求，就需要将得到的彩色图像灰度化，即转化成只包含亮度信息的灰度图像，然后再进行后续的图像处理。对于 RGB 颜色空间上各颜色分量进行加权平均的灰度化公式如下：

$$y = 0.299 * r + 0.587 * g + 0.114 * b \tag{2.1}$$

其中，y 表示 RGB 图像灰度化后的值，r、g、b 分别表示像素点在 RGB 颜色空间

上红、绿、蓝分量的值。由于人的肉眼在感知不同颜色时有着不同的敏感度，因此在进行图像灰度化处理时对于 RGB 颜色空间上红、绿、蓝三个分量的加权的值也不同，其中绿色分量加权的值最高，蓝色分量加权的值最低[2-3]。

2.2.2 图像滤波

由于人们在使用各种摄像设备获取图像信号时难免受到来自不同噪声源的干扰，这会导致图像质量变差，甚至导致图像在压缩和传输的过程中出现错误，因此在图像处理的过程中，图像去噪便成为了最开始的必备工序。在运行如运动目标检测等图像处理任务时，需通过如中值滤波、均值滤波等图像去噪技术针对图像中的噪声进行处理，以保证最终的图像处理结果能达到要求。由于噪声在图像中的表现通常是与其临近的像素点有着巨大的灰度值差异，因此可以通过各种不同的滤波器对图像进行针对性地去噪[2,4]。

以中值滤波为例，中值滤波有着能较好保留图像信号的陡峭边缘且算法简单易于实现的优点，在消除椒盐噪声时效果较好，是本文的实验研究中主要的滤波去噪手段之一。中值滤波的去噪思想如下：对于图像中的某个待确定像素点，首先将该待确定像素点邻域内的所有像素点的灰度值重新进行排序，接着用得到的这些灰度值的中间值来替换掉待确定像素点的灰度值，这样一来，如果待确定像素点是与其邻域内其他像素点有巨大的灰度值差异的噪声像素点，那么在滤波处理后该待确定像素点其邻域内其他像素点巨大的灰度值差异就会被消除，即噪声被消除，达到消除噪声的目的。中值滤波作为一种非线性的滤波去噪手段，相较于其他线性的滤波去噪手段，虽然同样会对图像造成信号衰减，但对于图像的陡峭边缘等细节保留较好，不会过分干扰肉眼对图像的观测，是一种较为理想的用于去除随机噪声的滤波去噪手段[3-4]。

2.3 光流法

光流指的是空间运动物体在观察成像平面上的像素运动的瞬时速度，它是由运动物体，即前景目标在背景场景中的运动，以及摄像装置本身的运动所产生的。在人用肉眼对运动物体进行观测时，被观测的运动物体在人的视网膜上不断形成的连续图像被人脑接收，于是人脑获取到了被观测的运动物体的外形、运动的信息，而这些信息是由运动物体，即前景目标在背景场景中的运动，以及摄像装置本身的运动所产生的，光流由这一系列运动产生，携带了来自于运动物体的关键信息。光流的概念早在 1950 年就已被 Gison 提出，并由光流的概念衍生出了光流场的概念，就是将运动物体投影到二维图像

中，得到的图像中各个特定的坐标点的灰度瞬时变化率即为光流矢量，由这些二维矢量构成的集合即为光流场。利用视频图像序列中每帧像素的差别，通过比对像素随时间发生的规律性变化，由此得出视频图像序列中背景环境与前景运动目标的光流差异，并由此将背景环境与前景运动目标从视频图像序列中分离开来，这就是光流法，在运动物体检测等计算机视觉的研究中具有较高应用价值、研究价值。光流法具有能有效应对同时有多个需要检测的运动目标的应用场景，能有效应对摄像装置本身就处在运动状态下的应用场景的优点，并且由于光流同时包含了前景运动目标的外形、运动信息，这使得光流法可直接进行运动目标检测，而不需提前去获取背景环境的任何有关信息。但相对的，光流法也有其缺点，首先光流法的计算非常复杂，这就意味着其计算耗时较长，实时性较差，并且对设备也有着较高要求。除此以外，在实际应用中，由于真实的背景动态环境常常是复杂多变的，如光线、天气、阴影等环境干扰因素会导致光流约束方程受到背景环境噪声的影响，无法求出正确的解，最终导致无法求出准确的光流场的计算结果。

由于运动物体的运动是相对连续的，运动物体目标不会在时间的连续变化过程中位置突然发生巨大的变化，由摄像设备所获取到的相邻帧图像中运动物体目标不会由特别大的位移，这是光流法必不可少的假定条件，并由此得出了亮度恒定不变的前提条件：对于一个图像序列而言，同一个运动物体目标随时间变化在相邻帧图像中运动时，图像的亮度是不会发生变化的，这也是光流法必不可少的假定条件，光流约束方程的满足离不开这一大前提。这里我们假设图像上的某个像素点在某时刻 t 的瞬时灰度值为 $I(x,y,t)$，并假设光流 $W(u,v)$ 在该像素点的水平偏移分量和垂直偏移分量分别为 $u(x,y)$ 和 $v(x,y)$，这二个分量各自的表达式分别如下：

$$u=\frac{dx}{dt} \tag{3.1}$$

$$v=\frac{dy}{dt} \tag{3.2}$$

该像素点在经过了时间间隔 dt 后其瞬时灰度值变为 $I(x+dx,y+dy,t+dt)$，假设在 $dt\to 0$ 的情况下，由于对于一个图像序列而言，同一个运动物体目标随时间变化在相邻帧图像中运动时，图像的瞬时灰度值是不会发生变化的，于是便可得到基本的光流方程如下：

$$I(x,y,t)=I(x+dx,y+dy,t+dt) \tag{3.3}$$

将式(3.3)进行泰勒展开并且忽略其二阶无穷小，便得到基本的光流约束方程，方程表达式如下：

$$\frac{\partial I}{\partial x}u+\frac{\partial I}{\partial y}v+\frac{\partial I}{\partial t}=0 \tag{3.4}$$

由于在式(3.4)中，方程有二个未知量，要想求得 u 和 v 的确定值，定解光流 $W(u,v)$，我们还需要额外引入的其他的约束条件。根据引入约束条件的角度不同，计算光流场的方法也不同，按照理论基础与数学方法的区别把它们分成四种：基于梯度的方法、基于匹配的方法、基于能量的方法、基于相位的方法。

以 LK 光流法为例，LK 光流法作为一种两帧差分的光流估计算法，它多了对于空间一致这一假定条件的要求，即在图像的 x-y 空间内，要求得像素点在 x、y 轴上各自的速度，这就需要相邻的像素点在前后两帧图像中也是相邻的。LK 光流法有着能将图像中像素点的瞬时速度表示出来的优点，该方法能利用光流对运动物体目标的运动状态进行估计，并将相应运动信息以较好的效果表示出来。然而 LK 光流法由于要对整幅图像的光流特征进行计算，因此也有计算量较大，运动目标检测的实时性较差，并且在图像上有其他较小的无关物体在运动时，会影响到 LK 光流法对于图像的光流特征提取，存在这些缺点[7]。

3 帧间差法及背景差分法

3.1 二帧间差法原理

由于运动物体的运动是相对连续的，运动物体目标不会在时间的连续变化过程中位置突然发生巨大的变化，由摄像设备所获取到的相邻帧图像中运动物体目标不会由特别大的位移，并且在时间间隔十分细微的情况下相邻两帧图像的光照也基本是一致的。基于这一前提条件，我们可以利用在相邻帧图像中对应像素点之间的灰度值变化这一信息来对运动物体目标进行检测，这就是帧间差法。

二帧差分法是一种十分典型的帧间差法，它是根据待检测图像序列的具体情况设置合适的阈值，对每相邻二帧原始图像进行检测：当相邻二帧图像中对应像素点之间的灰度值变化大于预先设置的阈值时，即判定该像素点对应的即是图像中的运动目标前景部分；当相邻二帧图像中对应像素点之间的灰度值变化小于预先设置的阈值时，即判定该像素点对应的即是图像中的背景部分。二帧差分法计算量很小，计算速度快，实时性强且对硬件要求不高，并且由于在时间间隔非常短的情况下相邻两帧图像的背景基本不会发生变化，二帧差分法基于其原理，可以较好地排除动态背景环境的光照等变化带来的干扰，也有着对动态背景适应性强这一优点。

使用二帧差分法来对视频图像进行运动目标检测的流程简述如下：第一步，将整个图像序列的每相邻二帧图像进行灰度化处理并进行差分运算，得到差分处理后的图像；第二步，根据原始图像的具体特征，选择合适的阈值，对在第一步得到的差分处理后的图像进行二值化处理，基于图像上各像素点灰度值与阈值的比较来将运动目标前景与环境背景分离开来：当相邻二帧图像中对应像素点之间的灰度值变化大于预先设置的阈值时，即判定该像素点对应的即是图像中的运动目标前景部分；当相邻二帧图像中对应像素点之间的灰度值变化小于预先设置的阈值时，即判定该像素点对应的即是图像中的背景部分；第三步，对于第二步中得到的二值图使用中值滤波等方法消除其噪声，并通过膨胀、腐蚀等方法进行进一步处理，最终达到较好的运动目标检测效果。

这里我们用 $F_k(x,y)$ 来表示第 k 帧图像，用 $F_{k-1}(x,y)$ 来表示第 $k-1$ 帧图像，用 $D_k(x,y)$ 来表示第 k 帧图像和第 $k-1$ 帧图像这两帧相邻的图像进行差分处理后得到的结果，则 $D_k(x,y)$ 的表达式如下：

$$D_k(x,y)=\left|\ F_k(x,y)-F_{k-1}(x,y)\ \right| \tag{3.5}$$

根据待检测图像序列的具体情况，取阈值为 T，对由式(3.5)求解所得出的 $D_k(x,y)$进行二值化处理，通过将得到的差分图像与阈值进行比较，得到相应的二值化图像。设该二值化图像 $T_k(x,y)$，则 $T_k(x,y)$的表达式如下：

$$T_k(x,y)=\begin{cases}0, & D_k(x,y)<T\\ 1, & D_k(x,y)\geq T\end{cases} \tag{3.6}$$

得到的二值化图像中像素值为 1 的像素点即为运动目标前景部分，像素值为 0 的点即为背景部分，对得到的二值化图像进行滤波去噪、膨胀腐蚀等处理，最终得到质量能达到要求的运动目标检测结果。

二帧差分法虽然有着计算简单，计算速度快，实时性强，在动态背景环境中有着较好的鲁棒性，能较好地适应动态背景环境中光线等条件的变化的优点，但是二帧差分法也有其不可忽视的缺点。二帧差分法对于运动目标的检测效果过于受目标的移动速度影响，如果运动目标的移动速度过快，将会导致检测到的运动目标面积过大，而如果运动目标的移动速度过慢，则会导致运动目标无法被检测到；由于原始图像序列中存在着运动目标的阴影、背景环境中存在噪声等干扰，二帧差分法最终的检测效果过于依赖阈值的确定，阈值过大会导致图像中部分不属于运动目标的区块也被检测为运动目标，阈值过小会导致最终检测到的运动目标不完整；当多个运动目标在原始图像上贴在一起时，二帧差分法无法将这些不同的运动目标区分开来；并且如果运动目标有着大片颜色分布较均匀的区域，那么最终的运动目标检测结果就很容易出现“空洞”，最终检测到的运动目标轮廓很可能会不完整，并且其连通性也无法保证在较好的水平。综上，二帧差分法作为简单而又经典的运动目标检测方法，有着较大的改良空间[3,8]。

3.2 二帧间差法实验

这里要处理的原始图像为 1 组 165 帧的图像序列，该图像序列截取取自一段画面中有 1 个人走过的监控录像，画面中走过的这个人便是本次实验所要提取的运动目标前景。输入的待处理图像序列中的第 33 帧、第 63 帧、第 93 帧、第 123 帧图像分别如图 3.1，图 3.2，图 3.3，图 3.4 所示：

图 3.1　第 33 帧原始图像

图 3.2　第 63 帧原始图像

图 3.3　第 93 帧原始图像

图 3.4　第 123 帧原始图像

通过用第 k 帧图像减去第 k-1 帧图像，得到差分图像，再选取一个合适的阈值，对得到的差分图像进行二值化处理，基于图像上各像素点灰度值与阈值的比较来将运动目标前景与环境背景分离开来：当相邻二帧图像中对应像素点之间的灰度值变化大于预先设置的阈值时，即判定该像素点对应的即是图像中的运动目标前景部分；当相邻二帧图像中对应像素点之间的灰度值变化小于预先设置的阈值时，即判定该像素点对应的即是图像中的背景部分。对于得到的二值图，这里使用中值滤波来消除其噪声。

以第 94 帧原始图像与第 93 帧原始图像相减的差分图像处理后得到的第 93 帧二值化图像为例为例，这里取阈值为 30，得到处理后的二值化图像如图 3.5 所示：

图 3.5　阈值为 30 时的第 93 帧二值化图像

如果阈值取 10、20 或 40，则得到的二值化图像分别如图 3.6，图 3.7，图 3.8 所示：

图 3.6　阈值为 10 时的第 93 帧二值化图像

图 3.7　阈值为 20 时的第 93 帧二值化图像

图 3.8　阈值为 40 时的第 93 帧二值化图像

通过对比可以发现，当阈值取过大时，算法对于运动目标前景的轮廓获取的完整性会变差，而阈值取过小时则可能导致原始图像中的环境噪声对处理结果产生干扰，这里的噪声主要是人物的阴影。也就是说，在使用二帧间差法进行运动目标检测时，阈值的选择对于最终的检测结果起到了至关重要的作用。本次实验中，阈值的合适取值为 30。

将得到的各帧二值化图像进行膨胀、腐蚀处理，改善这些二值化图像中运动目标前景的轮廓的连通性，再基由这些处理后的二值化图像获取到运动目标的轮廓边界，在相应帧的原始图像中标记出来。还是以第 33 帧、第 63 帧、第 93 帧、第 123 帧图像为例，最终得到的运动目标前景标记结果分别如图 3.9，图 3.10，图 3.11，图 3.12 所示：

图 3.9　第 33 帧标记后的图像

图 3.10　第 63 帧标记后的图像

图 3.11　第 93 帧标记后的图像

图 3.12　第 123 帧标记后的图像

3.3 三帧间差法原理

由于使用二帧差分法进行运动目标检测存在最终检测到的运动目标精确度不足的问题，可能出现最终检测到的运动目标不完整或运动目标区域过大的问题，针对这一缺点，在二帧差分法的基础上改进，通过从图像序列中选取连续的三帧图像，对这三帧图像中的每相邻二帧分别按照二帧差分法的方法进行差分运算，得到二幅差分图像，再将这二幅差分图像合并处理，最终得到精确度较好的运动目标，这就是三帧差分法。

使用三帧差分法来对视频图像进行运动目标检测的流程简述如下：第一步，将整个图像序列的每相邻三帧图像进行灰度化处理并进行差分运算，具体是选取其中每相邻二帧图像进行差分运算，再将得到的二幅差分图像相加，得到该相邻三帧图像最终的差分处理后的图像；第二步，根据原始图像的具体特征，选择合适的阈值，对在第一步得到的差分处理后的图像进行二值化处理，基于图像上各像素点灰度值与阈值的比较来将运动目标前景与环境背景分离开来：当差分图像中对应像素点之间的灰度值大于预先设置的阈值时，即判定该像素点对应的即是图像中的运动目标前景部分；当差分图像中对应像素点之间的灰度值小于预先设置的阈值时，即判定该像素点对应的即是图像中的背景部分；第三步，对于第二步中得到的二值图使用中值滤波等方法消除其噪声，并通过膨胀、腐蚀等方法进行进一步处理，最终达到较好的运动目标检测效果。

这里我们用 $F_k(x,y)$ 来表示第 *k* 帧图像，用 $F_{k-1}(x,y)$ 来表示第 *k-1* 帧图像，用 $F_{k+1}(x,y)$ 来表示第 *k+1* 帧图像，用 $D_k(x,y)$ 来表示第 *k* 帧图像和第 *k+1* 图像、第 *k-1* 帧图像这三帧相邻的图像进行差分处理后得到的结果，则 $D_k(x,y)$ 的表达式如下：

$$D_k(x,y)=\left|\ F_k(x,y)-F_{k-1}(x,y)\ \right|+\left|\ F_k(x,y)-F_{k+1}(x,y)\ \right| \tag{3.7}$$

根据待检测图像序列的具体情况，取阈值为 *T*，对由式(3.7)求解所得出的 $D_k(x,y)$ 进行二值化处理，通过将得到的差分图像与阈值进行比较，得到相应的二值化图像。设该二值化图像为 $T_k(x,y)$，则 $T_k(x,y)$ 的表达式如下：

$$T_k(x,y)=\begin{cases}0, & D_k(x,y)<T\\ 1, & D_k(x,y)\geq T\end{cases} \tag{3.8}$$

得到的二值化图像中像素值为 1 的像素点即为运动目标前景部分，像素值为 0 的点即为背景部分，对得到的二值化图像进行滤波去噪、膨胀腐蚀等处理，最终得到质量能达到要求的运动目标检测结果。

三帧差分法作为二帧差分法的改良，有效地解决了最终检测到的运动目标精确度不足的问题，并继承了二帧差分法计算量小，计算速度快，实时性强且对硬件要求不高，可以较好地排除动态背景环境的光照等变化带来的干扰，对动态背景适应性强的优点。但三帧差分法依旧存在着如对于动态背景环境中的噪声的抗干扰能力不足，过于依赖阈值的确定，无法将多个贴在一起的运动目标有效区分开，无法在运动目标有大片颜色较为均匀的区域时有效检测运动目标等原先二帧差分法也存在的不足之处。

3.4 三帧间差法实验

与上文中二帧间差法实验相同，这里要处理的原始图像仍为同样的 1 组 165 帧的图像序列，该图像序列截取取自一段画面中有 1 个人走过的监控录像，画面中走过的这个人便是本次实验所要提取的运动目标前景。输入的待处理图像序列中的第 33 帧、第 63 帧、第 93 帧、第 123 帧图像分别如图 3.13，图 3.14，图 3.15，图 3.16 所示：

图 3.13　第 33 帧原始图像

图 3.14　第 63 帧原始图像

图 3.15　第 93 帧原始图像

图 3.16　第 123 帧原始图像

通过用第 k 帧图像分别减去第 k-1 帧图像、第 k+1 帧图像，得到二幅差分图像，将这二幅差分图像相加得到处理后的差分图像，再选取一个合适的阈值，对得到的差分图像进行二值化处理，基于图像上各像素点灰度值与阈值的比较来将运动目标前景与环境背景分离开来：当相邻三帧图像中对应像素点之间的灰度值变化大于预先设置的阈值时，即判定该像素点对应的即是图像中的运动目标前景部分；当相邻三帧图像中对应像素点

之间的灰度值变化小于预先设置的阈值时，即判定该像素点对应的即是图像中的背景部分。对于得到的二值图，这里使用中值滤波来消除其噪声。

以通过第 93 帧、第 94 帧、第 95 帧图像进行处理得到的差分图像再次处理后得到的第 93 帧二值化图像为例为例，这里取阈值为 30，得到处理后的二值化图像如图 3.17 所示：

图 3.17　阈值为 30 时的第 93 帧二值化图像

如果阈值取 10、20 或 40，则得到的二值化图像分别如图 3.18，图 3.19，图 3.20 所示：

图 3.18　阈值为 10 时的第 93 帧二值化图像

图 3.19　阈值为 20 时的第 93 帧二值化图像

图 3.20　阈值为 40 时的第 93 帧二值化图像

通过对比可以发现，当阈值取过大时，算法对于运动目标前景的轮廓获取的完整性会变差，而阈值取过小时则可能导致原始图像中的环境噪声对处理结果产生干扰，这里的噪声主要是人物的阴影。也就是说，与使用二帧间差法进行运动目标检测时相同，在使用三帧间差法进行运动目标检测时，阈值的选择对于最终的检测结果起到了至关重要的作用。本次实验中，阈值的合适取值为 30。另外，相较于使用二帧间差法进行运动目标检测时的效果，使用三帧间差法进行运动目标检测时提取到的运动目标轮廓显然要完整的多，这是三帧间差法作为二帧间差法改良后的算法的进步之处。

将得到的各帧二值化图像进行膨胀、腐蚀处理，改善这些二值化图像中运动目标前景的轮廓的连通性，再基由这些处理后的二值化图像获取到运动目标的轮廓边界，在相应帧的原始图像中标记出来。还是以第 33 帧、第 63 帧、第 93 帧、第 123 帧图像为例，最终得到的运动目标前景标记结果分别如图 3.21，图 3.22，图 3.23，图 3.24 所示：

图 3.21　第 33 帧标记后的图像

图 3.22　第 63 帧标记后的图像

图 3.23　第 93 帧标记后的图像

图 3.24　第 123 帧标记后的图像

3.5 背景差分法原理

背景差分法是当下较为常用的一种目标检测的方法，其原理是从视频或图像序列中选取需要进行运动目标检测的那一帧图像，使用背景建模的方法以得到与当前待检测的

那一帧原始图像所适配的背景图像，再用所选取的那一帧原始图像与背景图像进行差分运算，得到差分图像，并设置合适的阈值进行二值化处理，对得到的二值图进行滤波去噪等处理，最终得到那一帧原始图像中所需要检测到的运动目标前景。

使用背景差分法进行运动目标检测有着算法简单易懂，实时性强的优点，并且相较于帧间差法，背景差分法能更精确地检测到较为完整的运动目标前景，对于运动目标移动速度不受控制以及运动目标存在大片颜色均匀的区域的情况，背景差分法都能做到有效应对。但是真实的动态背景环境通常也是复杂多变的，获取到的原始图像中的背景部分通常存在诸如光线变化等干扰因素，这就需要有合适的算法来保证得到的背景模型质量并能够有效地进行实时的更新，排除动态背景环境中存在的干扰，使得背景环境中的干扰因素不会影响到运动目标前景的检测，保证运动目标检测的精度。也就是说，背景差分法的算法核心就是背景建模算法的设计。

使用背景差分法来对视频图像进行运动目标检测的流程简述如下：第一步，对获取到的原始图像序列中的一帧图像进行滤波去噪等处理，基于合适的背景建模原理进行背景建模，得到背景模型，并设计合适的训练模型使得背景模型能以合适的速率进行更新，以应对动态背景环境中存在的如光线等变化所带来的干扰；第二步，将原始图像序列中的每帧图像与对应的背景图像进行差分运算，得到差分处理后的图像；第三步，根据原始图像的具体特征，选择合适的阈值，对在第二步得到的差分处理后的图像进行二值化处理，基于图像上各像素点灰度值与阈值的比较来将运动目标前景与环境背景分离开来：当差分图像中对应像素点之间的灰度值大于预先设置的阈值时，即判定该像素点对应的即是图像中的运动目标前景部分；当差分图像中对应像素点之间的灰度值小于预先设置的阈值时，即判定该像素点对应的即是图像中的背景部分；第四步，对于第三步中得到的二值图使用中值滤波等方法消除其噪声，并通过膨胀、腐蚀等方法进行进一步处理，最终达到较好的运动目标检测效果。

这里我们用$F_k(x,y)$来表示第 k 帧图像，用$B_k(x,y)$来表示第 k 帧图像所对应的背景建模得到的背景图像，用$D_k(x,y)$来表示第 k 帧图像和第 k 帧图像所对应的背景建模得到的背景图像进行差分处理后得到的结果，则$D_k(x,y)$的表达式如下：

$$D_k(x,y)=\left| F_k(x,y)-B_k(x,y) \right| \tag{3.9}$$

根据待检测图像序列的具体情况，取阈值为 T，对由式(3.7)求解所得出的$D_k(x,y)$进行二值化处理，得到的相应二值化图像设为$T_k(x,y)$，则$T_k(x,y)$的表达式如下：

$$T_k(x,y)=\begin{cases}0, & D_k(x,y)<T\\ 1, & D_k(x,y)\geq T\end{cases} \tag{3.10}$$

得到的二值化图像中像素值为 1 的像素点即为运动目标前景部分，像素值为 0 的点即为背景部分，对得到的二值化图像进行滤波去噪、膨胀腐蚀等处理，最终得到质量能达到要求的运动目标检测结果。

背景差分法作为当前常用的一种用于运动目标检测的视频图像处理方法，在视频监控技术等诸多领域有着广泛的应用，其应用前景相当可观。由于摄像设备获取到的原始图像序列通常存在由动态背景环境带来的各种因素的不可预测的变化，如光线的变化、树叶及塑料袋等无关物体的飘过、水面的波纹、树木枝叶随风摆动、以及其他原始图像上的噪声干扰，这使得背景建模变得复杂困难，于是就有了对于背景建模算法改良的需求。目前市面上的背景建模算法大致存在以下几个缺点：第一，原始图像的背景环境中的一些干扰条件可能会被算法视为运动目标前景，如运动目标自身的阴影、原始图像中其他飘过的无关物体等；第二，背景模型的更新频率可能会与运动目标的移动速度产生冲突，如背景模型更新频率过快会导致移动速度较慢的目标被算法视作图像的背景部分从而不被检测到，背景模型更新速率过慢则会导致移动速度较快的目标在当前帧之前的帧当中所处的位置也被视为运动目标前景从而导致检测结果中存在虚影。当前市面上有许多各不相同的背景建模算法，其中基于高斯混合模型的背景建模技术是一个经典的背景建模算法，本文将对其进行介绍及研究。

4 基于高斯混合模型的运动目标检测

4.1 高斯分布基本概念

高斯分布又称正态分布，这一概念最早由法国数学家 Abraham de Moivre 于 1733 年提出，后被德国数学家 Gauss 首先应用于天文学的研究。概率函数指的将事件概率表示成有关于事件变量的函数；概率分布函数指的是假设一个随机变量 y 并取一个确定的数值 x，y 小于 x 的概率是有关于 x 的函数，则该函数称为随机变量 y 的分布函数，可由其来确定随机变量落在任意范围内的概率；概率密度函数指的是假设一个随机变量 y 并取一个确定的数值 x，概率密度函数可以描述 y 取值在 x 附近的概率，概率密度则指的是假设一个随机变量 y，y 在一个区间(事件的取值范围)的总的概率除以该段区间的长度。正态分布指的是假设一个连续型随机变量 X，若该连续型随机变量 X 的概率密度 y 等于以下公式：

$$y = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \tag{4.1}$$

其中，μ、σ 都是常数，μ 为均值，σ^2 为方差，称 X 服从参数为 μ、σ 的正态分布，即高斯分布，记作 $X \sim N(\mu,\sigma^2)$。

由中心极限定理我们可以得知，当同一个事物受到多种因素的影响时，不论这些因素本身是何种分布，分布情况具体如何，在将这些因素进行加和汇总后，所得结果的平均值一定是服从正态分布的；即便由于各种因素对结果的影响并非简单加法，而是符合乘法规律，最终的结果不服从正态分布，其对数值也是服从正态分布的。经统计，在自然现象和社会现象中，正态分布是最为常见的，大量的随机变量都服从于正态分布，或者近似服从正态分布。比如说一片区域内所有成年女性的身高、一片区域内所有同种鸟类的寿命、对同一个物件多次测量所得到的误差、海洋波浪的高度变化等，诸如此类的大量随机变量都是服从正态分布的。

高斯分布在统计学、医学、计算机科学等诸多领域有着广泛应用。由于在视频图像处理中图像的背景像素值也服从于高斯分布，因此在进行运动目标检测的背景建模时就可以使用高斯分布来对背景像素进行描述，实现背景建模的目的。

4.2 单高斯模型

单高斯模型是高斯混合模型等其他模型的基础。由于在视频图像处理的应用中，图像的背景像素值服从于高斯分布，我们可以使用高斯分布来对图像中的背景进行描述，用于运动目标检测。用高斯模型来描述视频或图像序列中的某 1 个像素点在某时刻的状态，以此来进行背景建模，得到我们所需要的背景图像，这种模型就是单高斯模型。单高斯模型适用于背景图像的像素点分布有着像素峰值比较单一这一特征，即视频或图像序列中的背景部分为背景单模态背景时的情形。假设视频图像中在时刻t的某个像素点的像素值为Y_t，μ_t为该像素点在时刻t的均值，$\sum_t$为该像素点在时刻t的方差矩阵，单高斯分布模型表达式如下所示：

$$\eta\left(Y_t,\mu_t,\sum_t\right)=\frac{1}{\sqrt{2\pi\left|\sum_t\right|}}e^{-\frac{1}{2}\left(Y_t-\mu_{i,t}\right)^T\sum_t^{-1}\left(Y_t-\mu_t\right)} \tag{4.2}$$

用单高斯模型进行背景建模，首先需要对高斯分布的参数进行初始化。对于待检测的视频或图像序列，通过对于图像的各个像素点进行计算，得到图像的像素值的均值为μ_0，方差为σ_0^2，由得出的这二参数来初始化高斯分布函数的参数，我们所要求得的背景图像便可以通过高斯分布函数来估计得出。μ_0和σ_0^2各自的表达式如下式所示：

$$\mu_0(x,y)=\frac{1}{T}\sum_{i=0}^{T-1}Y_i(x,y) \tag{4.3}$$

$$\sigma_0^2(x,y)=\frac{1}{T}\sum_{i=0}^{T-1}\left[Y_i(x,y)-\mu_0(x,y)\right]^2 \tag{4.4}$$

考虑到实际检测到的原始图像中可能存在光线变化、无关物体飘过等干扰因素，这会干扰到运动目标检测的运动目标前景判断，这里我们需要不断地对高斯分布的参数进行更新，用于修正背景模型使其能适应原始图像序列中动态背景环境的变化，避免动态背景环境中干扰因素导致运动目标检测产生累积性错误。这里引入一个学习率α用于表示高斯分布参数的更新速率，高斯分布的各个参数更新公式如下：

$$\mu_t=(1-\alpha)\mu_{t-1}+\alpha Y_t \tag{4.5}$$

$$\sigma_t^2=(1-\alpha)\sigma_{t-1}^2+\alpha\left(Y_t-\mu_t\right)^T\left(Y_t-\mu_{t-1}\right) \tag{4.6}$$

$$\alpha=K\eta\left(Y_t\middle|\mu_{t-1},\sigma_{t-1}^2\right) \tag{4.7}$$

假设更新后的背景图像为$H_t=\left[\mu_t,\ \sigma_t^2\right]$。式(4.7)中$\eta\left(Y_t\middle|\mu_{t-1},\sigma_{t-1}\right)$表示均值$\mu_{t-1}$、方差为$\sigma_{t-1}^2$的高斯分布概率密度函数，式(4.7)中的$K$为设定的常数。

学习率α是一个介于 0 到 1 之间的常数。根据实际获取到的原始图像序列中动态背景环境的变化情况，学习率α的取值也相应地需要进行改变。如果原始图像序列中动态背景环境的变化较为剧烈，运动目标的移动速度较快，则学习率α就应该取较大的值以适应动态背景环境的变化及运动目标的快速移动；如果原始图像序列中背景环境的变化并不大，运动目标的移动速度较慢，则学习率α就应该取较小的值以避免将运动目标判定为背景部分，保证运动目标检测的完整性。学习率α的取值通常为 0.005。

在使用单高斯模型建立了背景模型后，即可继续对获取到的图像序列进行运动目标检测。首先，对当前帧图像 f_t 和计算得到的背景图像 H_t 进行差分运算，得到差分图像，再根据原始图像序列的具体特征，选取一个合适的阈值 T，将得到的差分图像与阈值 T 进行比较，即可得到相应的二值化图像。这里设该二值化图像为$T_k(x,y)$，则$T_k(x,y)$的表达式如下：

$$T_k(x,y)=\begin{cases}0, & D_k(x,y)<T\\ 1, & D_k(x,y)\geq T\end{cases} \tag{4.8}$$

得到的二值化图像中像素值为 1 的像素点即为运动目标前景部分，像素值为 0 的点即为背景部分，对得到的二值化图像进行滤波去噪、膨胀腐蚀等处理，最终得到质量能达到要求的运动目标检测结果。

4.3 高斯混合模型

背景图像像素点的像素峰值比较单一时，适合使用单高斯模型，算法较为简单。然而实际拍摄到的原始图像中常常有着较为复杂的动态背景环境，背景中可能存在如光线的变化、树叶及塑料袋等无关物体的飘过、水面的波纹、树木枝叶随风摆动等干扰因素，这使得背景图像像素点的像素值呈现多峰分布，这种情况下，单高斯分布背景模型就不能较好地对背景模型进行描述，于是就需要使用多个高斯模型来进行背景建模，即使用高斯混合模型来进行背景建模[2]。

这里我们将背景图像中某个像素点在 RGB 空间上的的颜色分布用 M 个高斯分布来进行表示，并让高斯分布的均值、方差这二个参数以及权值的更新能够适应背景图像的像素更新。设原始图像序列中在某一时刻 t 图像的某个像素点的像素值的值为 Y，如果该像素值与 M 个高斯分布中的某一个高斯分布能够相匹配，就取这个相匹配的高斯分布的均值作为当前帧图像的背景的像素值，并对高斯分布的各项参数进行更新，而如果

该像素值无法与任意一个高斯分布相匹配，则该时刻该像素点即为运动目标前景部分。通常情况下，M 的取值为 3 到 5。使用高斯混合模型进行背景建模的流程简述如下：

首先要定义像素模型及初始化参数。用多个单高斯模型组成的线性集合来描述随着时间的变化每个像素的像素值的历史变化记录 $\{Y_1,....,Y_t\}$，假设各个高斯分布的概率密度函数为：

$$\eta\left(Y_t,\mu_{i,t},\sum_{i,t}\right)i=1,2,...,M \tag{4.9}$$

其中，下标 t 表示观测时刻，Y_t 表示在该观测时刻 t 图像的某个像素点的像素值。

这里我们假设方差矩阵的表达式如下：

$$\sum_{k,t}=\sigma_k^2 I \tag{4.10}$$

其中，I 表示单位矩阵，以这种方式来表示方差矩阵有着不会对结果产生过多影响且计算简单，实时性强的优点。在 RGB 颜色模型中，根据表达式，图像中各个像素点的各个分量相互独立且有着一样的方差。使用这种算法来进行运算，降低了计算的复杂度，提高了计算效率。尽管这可能存在与真实状况有出入，会牺牲掉部分运动目标检测的精确性的缺陷，但这并不会对计算结果造成太大的不利影响。

高斯混合模型中每个单高斯模型各自的概率密度表达式如下：

$$\eta\left(Y_k\left|\mu_{k,t},\sigma_{k,t}\right.\right) \tag{4.11}$$

Y_t 的概率密度函数由 M 个单高斯模型的函数组合表示，其表达式如下：

$$P(Y_t)=\sum_{k=1}^{M}W_{k,t}\times\eta\left(Y_t\left|\mu_{k,t},\sigma_{k,t}\right.\right) \tag{4.12}$$

其中，$\mu_{k,t}$ 表示单高斯模型的均值，$\sigma_{k,t}^2$ 表示单高斯模型的方差，$W_{k,t}$ 表示单高斯模型的权值，$W_{k,t}=p(k,t)$，$k=1,2,...,M$,且有 $\sum_{k=1}^{M}W_{k,t}=1$，$p(k,t)$ 表示在观测时刻 t 图像的某个像素点的像素值是由第 k 状态产生的先验概率，$W_{k,t}$ 表示采用第 k 个高斯分布来表示该像素点的像素值时其真实程度。

用高斯混合模型进行背景建模，首先需要对 M 个高斯分布的参数进行初始化。对于待检测的视频或图像序列，通过对于图像的各个像素点进行计算，得到图像的像素值的均值为 μ_0，方差为 σ_0^2，由得出的这二参数来初始化 M 个高斯分布函数的参数，我们

所要求得的背景图像便可以通过这 M 个高斯分布函数来估计得出。μ_0 和 σ_0^2 各自的表达式如下式所示：

$$\mu_0 = \frac{1}{N}\sum_{t=0}^{N-1} Y_t \tag{4.13}$$

$$\sigma_0^2 = \frac{1}{N}\sum_{t=0}^{N-1} (Y_t - \mu_0)^2 \tag{4.14}$$

由于计算平均像素值 μ_0 和像素值的方差 σ_0^2 需要用到多帧图像，这对计算机内存容量大小要求较高。如果对高斯混合模型参数初始化的效率以及速率要求不高，则可将方差 σ_0^2 初始化为一个较大的值。

设第 i 个单高斯分布的权重为 W_i，均值为 μ_i，各自表达式如下所示：

$$W_i = \frac{1}{M} \tag{4.15}$$

$$\mu_i = 255 \times \left(\frac{i}{M}\right), \ i = 1,,2,...,M \tag{4.16}$$

对于输入的视频或图像序列，如果第一帧原始图像中不存在运动目标，或运动目标在图像中的区域占比不大，即认为该帧图像就是背景图像的概率很大，那么就可以直接利用第一帧原始图像，通过对其各个像素点进行计算，得到图像的像素值的均值和方差，由得出的这二参数来对高斯混合模型中的其中一个高斯分布进行初始化，并对于该高斯分布的权重取最大的值，对其他高斯分布的权重则取各自相等的较小值，且这些其他高斯分布的均值都置为零。这种算法设计让权重取最大的值的高斯分布的均值在后续高斯混合模型参数的更新时能有更大的概率被视作背景像素，有着降低计算量，加快背景建模速率，改善算法实时性的好处。

由于实际检测到的原始图像中可能存在光线变化、树枝飘动、水面波纹扩散等干扰因素，这会对运动目标检测的运动目标前景判断造成干扰，因此我们需要不断地对高斯分布的参数进行更新，用于修正背景模型使其能适应原始图像序列中动态背景环境的变化，避免动态背景环境中干扰因素导致运动目标检测产生累积性错误。高斯混合模型参数的更新也是基于高斯混合模型的背景建模技术中的核心算法。高斯混合模型参数的更新主要是针对高斯混合模型中各高斯分布的均值、方差以及各高斯分布的权重进行更新。高斯混合模型参数的更新流程简述如下：

第一步，取一个判断值 $f(x)$，如果将在某个时刻观测到的图像的某个像素点的像素值与高斯混合模型中的某个高斯分布函数的均值代入以下判断公式：

$$f(x)=\begin{cases}1, |Y_t - \mu_{i,t}| < \lambda \sigma_{i,t} \\ 0, |Y_t - \mu_{i,t}| \geq \lambda \sigma_{i,t}\end{cases} \tag{4.17}$$

若 $f(x)$的值为 1，则认为该时刻观测到的该图像的该像素点的像素值与该高斯分布相匹配；若 $f(x)$的值为 0，则认为高斯混合模型中不存在与该时刻观测到的该像素点的像素值相匹配的高斯分布。其中 $\mu_{i,t}$ 是第 i 个高斯分布函数在观测时刻 t 的均值，$\sigma_{i,t}$ 是第 i 个高斯分布函数在观测时刻 t 的均值，λ 是我们引入的一个自定义参数，这里 λ 的取值通常为 2.5。

第二步，经过第一步的算法处理，可能出现的情况有以下二种：一种是高斯混合模型中存在与该时刻观测到的该图像的该像素点的像素值相匹配的高斯分布，另一种是高斯混合模型中不存在与该时刻观测到的该图像的该像素点的像素值相匹配的高斯分布。这二种不同的情况有着各自相对应的高斯混合模型参数的更新方式，以下是对这二种不同的具体情况各自相应的高斯分布的参数更新方式的说明：

如果高斯混合模型中存在与该时刻观测到的该图像的该像素点的像素值相匹配的高斯分布，我们要从高斯混合模型中找到最为匹配的那个高斯分布，然后对找到的这个高斯分布进行参数更新。对这个高斯分布进行参数更新的流程简述如下：首先要将与该时刻观测到的该图像的该像素点的像素值所不匹配的其他高斯分布的权重降低，设这些不匹配的其他高斯分布的权重为$W_{i,t}$，并按以下式子降低权重：

$$W_{i,t}=(1-\alpha)W_{i,t-1} \tag{4.18}$$

这样一来，该时刻观测到的图像的背景的整体像素分布的变化就能被完整且准确地被表示出来。综上，随着观测时刻的推移，如果图像上像素点的像素值在新的时刻存在与之向匹配的来自高斯混合模型中的高斯分布，则应选取高斯混合模型中与该像素值最为匹配的那个高斯分布，对该高斯分布的权重进行适当地增加，同时相应地也要降低其他与该像素值所不匹配的高斯分布的权重。

除此之外，我们还需要对于所选取的与该像素值最为匹配的那个高斯分布的均值 $\mu_{i,t}$ 和方差 $\sigma_{i,t}^2$ 这二个参数进行更新，对于这二个参数各自的更新公式如下式所示：

$$\mu_{i,t} = (1 - \rho)\mu_{i,t-1} + \rho Y_t \tag{4.19}$$

$$\sigma_{i,t}^2 = (1-\rho)\sigma_{i,t-1}^2 + \rho(Y_t - \mu_{i,t})^T(Y_t - \mu_{i,t}) \tag{4.20}$$

$$\rho = \alpha\eta(Y_t|\mu_{i,t-1},\ \sigma_{i,t-1}) \tag{4.21}$$

其中，ρ为参数学习率，学习率α则表示高斯分布参数的更新速率。学习率α是一个介于 0 到 1 之间的常数，与上个小节中单高斯模型的参数更新类似，根据实际获取到的原始图像序列中动态背景环境的变化情况，学习率α的取值也相应地需要进行改变。如果原始图像序列中动态背景环境的变化较为剧烈，运动目标的移动速度较快，则学习率α就应该取较大的值以适应动态背景环境的变化及运动目标的快速移动；如果原始图像序列中背景环境的变化并不大，运动目标的移动速度较慢，则学习率α就应该取较小的值以避免将运动目标判定为背景部分，保证运动目标检测的完整性。至于高斯混合模型中其他与该像素值所不匹配的高斯分布，则不必对它们的均值和方差也进行参数的更新。

如果高斯混合模型中不存在与该时刻观测到的该图像的该像素点的像素值相匹配的高斯分布，则需要在高斯混合模型中选出权重最小的一个高斯分布，将其除去，然后引入一个新的高斯分布。对于新引入的这个高斯分布，将其均值设置为与该时刻观测到的该图像的该像素点的像素值的均值相同，将其方差设置为一个较大的值，并保证其权重依然为最小。与此同时，令高斯混合模型中其它高斯分布的均值与方差保持不变，并按照之前提过的“如果高斯混合模型中存在与该时刻观测到的该图像的该像素点的像素值相匹配的高斯分布”这种情况中的方法来降低它们的权重。

第三步，在完成了对高斯混合模型中各个高斯分布各自的参数的更新后，还需要对高斯混合模型中的这些高斯分布的权重进行归一化处理，归一化处理公式如下所示：

$$W_{i,t} = \frac{W_{i,t}}{\sum_{j=1}^{K} W_{j,t}} \tag{4.22}$$

4.4 建立背景模型以进行运动目标检测

设高斯混合模型中各个高斯分布的相对值为$W_{i,t}/\sigma_{i,t}$，然后计算得出这些相对值，并对这些相对值进行降序排序，便可以得到高斯混合模型中各个高斯分布的优先级。对于这些高斯分布来说，其相对值越大，其被视作背景像素的概率也越大，方差则越小。确定背景像素的模型的方法流程简述如下：第一步，根据各个高斯分布的权重和标准差，

相除得到相对值$W_{i,t}/\sigma_{i,t}$；第二步根据各个高斯分布相对值$W_{i,t}/\sigma_{i,t}$的大小来对各个高斯分布进行优先级排列，相对值$W_{i,t}/\sigma_{i,t}$越大的高斯分布排列在越前面，它描述背景像素的可能性越大，否则，它描述背景像素的可能性越小；第三步，从 M 个高斯分布中选出 N 个高斯分布作为背景模型，用 Z 来表示描述背景像素的高斯分布在高斯混合模型中的比重，则背景模型由下式表示：

$$U = \arg\min_N \left(\sum_{k=1}^{N} W_{i,k} > Z \right) \tag{4.23}$$

其中，Z 是一个全局的先验概率，它的作用是对图像中的像素最可能成为背景像素的概率进行描述，它的值决定了高斯混合模型中高斯分布的数量。在对 Z 进行取值时需保证 Z 的取值大小合适，过大或过小都会导致实际应用时出现问题。当 Z 的取值过大时，由于高斯分布的数量变多，这会导致计算量也随之增加，并且 Z 的取值过大会导致权重过小的高斯分布也被用于背景建模，这可能会使得部分原本应该是运动目标前景的区域也被视作背景的一部分；而如果 Z 的取值过小，则会导致高斯混合模型中高斯分布的数量变为 1，高斯混合模型的实际背景建模效果只能达到单高斯模型的程度，导致背景建模时算法对动态背景环境的适应能力不足[2]。

最后，将在观测时刻 t 观测到的图像的某个像素点的像素值Y_t和选出来的与Y_t相匹配的高斯分布的均值$\sigma_{i,t}$进行相减，再将得到的差值的绝对值与该高斯分布的标准差的 λ 倍进行比较，若得到的差值的绝对值大于该高斯分布的标准差的 λ 倍，则该像素点就被视作运动目标前景的一部分，若得到的差值的绝对值小于该高斯分布的标准差的 λ 倍，则该像素点就被视作背景的一部分。一般地，我们对于 λ 的取值为 2.5。

4.5 简单背景差分法实验

这里要处理的原始图像仍为与前文帧间差法实验相同的 1 组 165 帧的图像序列，该图像序列截取取自一段画面中有 1 个人走过的监控录像，画面中走过的这个人便是本次实验所要提取的运动目标前景。输入的待处理图像序列中的第 33 帧、第 63 帧、第 93 帧、第 123 帧图像分别如图 4.1，图 4.2，图 4.3，图 4.4 所示：

图 4.1　第 33 帧原始图像

图 4.2　第 63 帧原始图像

图 4.3　第 93 帧原始图像

图 4.4　第 123 帧原始图像

由于该组原始图像序列背景环境无明显变化，且该组原始图像序列的前 22 帧图像中人物没有出现，即不存在运动目标前景，这里直接取第 1 帧原始图像作为背景图像，将其它帧图像分别与背景图像相减，得到差分图像，然后设置阈值为 70，对得到的差分图像进行二值化处理，基于图像上各像素点灰度值与阈值的比较来将运动目标前景与环境背景分离开来：如果当前帧图像与背景图像中对应像素点之间的灰度值差大于预先设置的阈值，即判定该像素点对应的即是图像中的运动目标前景部分；如果当前帧图像与背景图像中对应像素点之间的灰度值差小于预先设置的阈值，即判定该像素点对应的即是图像中的背景部分。对于得到的二值图，这里使用中值滤波来消除其噪声。

第 1 帧原始图像如图 4.5 所示：

图 4.5　第一帧原始图像

第 93 帧二值化图像如图 4.6 所示：

图 4.6　阈值为 10 时的第 93 帧二值化图像

很明显，相较于帧间差法，在背景环境没有明显变化的情况下，使用简单的背景差分法就可以提取到完整得多的运动目标前景。

将得到的各帧二值化图像进行膨胀、腐蚀处理，改善这些二值化图像中运动目标前景的连通性，再基由这些处理后的二值化图像获取到运动目标的轮廓边界，在相应帧的原始图像中标记出来。还是以第 33 帧、第 63 帧、第 93 帧、第 123 帧图像为例，最终得到的运动目标前景标记结果分别如图 4.7，图 4.8，图 4.9，图 4.10 所示：

图 4.7　第 33 帧标记后的图像

图 4.8　第 63 帧标记后的图像

图 4.9　第 93 帧标记后的图像

图 4.10　第 123 帧标记后的图像

4.6 基于高斯混合模型的背景差分法实验

本次实验需处理 1 组 200 帧的图像序列。这组图像序列截取自一段某公路上的监控录像，画面中经过的车辆是本次实验要检测的运动目标前景。输入的待处理图像序列中的第 33 帧、第 63 帧、第 93 帧、第 123 帧图像分别图 4.11，图 4.12，图 4.13，图 4.14 所示：

图 4.11　第 33 帧原始图像

图 4.12　第 63 帧原始图像

图 4.13　第 93 帧原始图像

图 4.14　第 123 帧原始图像

由于该组图像序列中存在“车来车往”的情况，无法从中选取一帧直接作为背景图片，且该组图像序列中存在光线变化、行人路过等动态背景环境中的干扰因素，这里选择使用高斯混合模型进行背景建模。

要进行背景建模，首先要对高斯分布的各项参数进行设置。本次实验中，所使用的高斯混合模型由 3 个高斯分布构成，将高斯混合模型中第一个高斯分布的均值设置为图像序列中第一帧图像的像素值，权重设置为 1，其它 2 个高斯分布的均值设为 0，初试的权重值也为 0；式(4.17)中 λ 的取值为 2.5，$\sigma_{i,t}$ 的取值为 30，式(4.18)中的 α 设置为 0.005，式(4.23)中的 Z 取值为 0.6。

建立背景模型得到的背景图像如图 4.15 所示：

图 4.15　背景图像

以第 33 帧、第 63 帧、第 93 帧、第 123 帧图像为例，接下来基于建立好的背景模型进行背景差分法获取二值化图像，结果分别如图 4.16，图 4.17，图 4.18，图 4.19 所示：

图 4.16　第 33 帧二值化图像

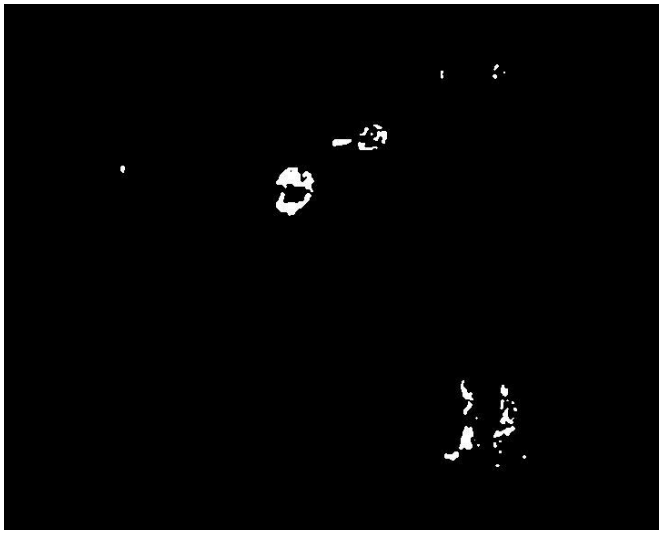

图 4.17　第 63 帧二值化图像

图 4.18　第 93 帧二值化图像

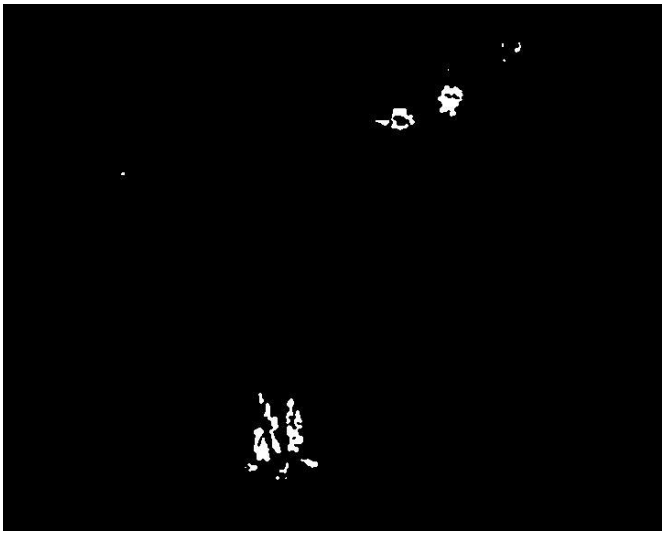

图 4.19　第 123 帧二值化图像

将得到的各帧二值化图像进行膨胀、腐蚀处理，改善这些二值化图像中运动目标前景的连通性，再基由这些处理后的二值化图像获取到运动目标的轮廓边界，在相应帧的原始图像中标记出来。还是以第 33 帧、第 63 帧、第 93 帧、第 123 帧图像为例，最终得到的运动目标前景标记结果分别如图 4.20，图 4.21，图 4.22，图 4.23 所示：

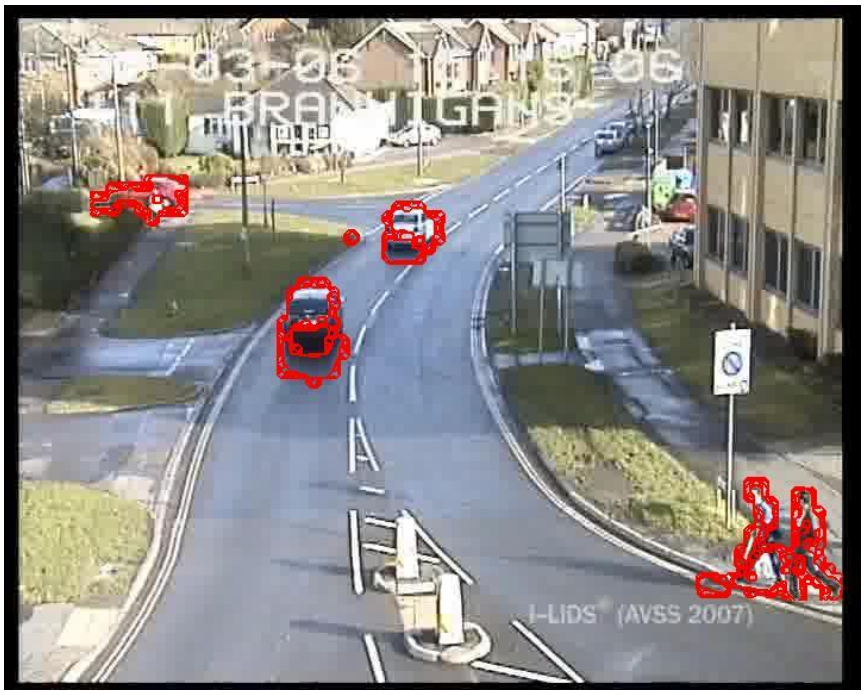

图 4.20　第 33 帧标记后的图像

图 4.21　第 63 帧标记后的图像

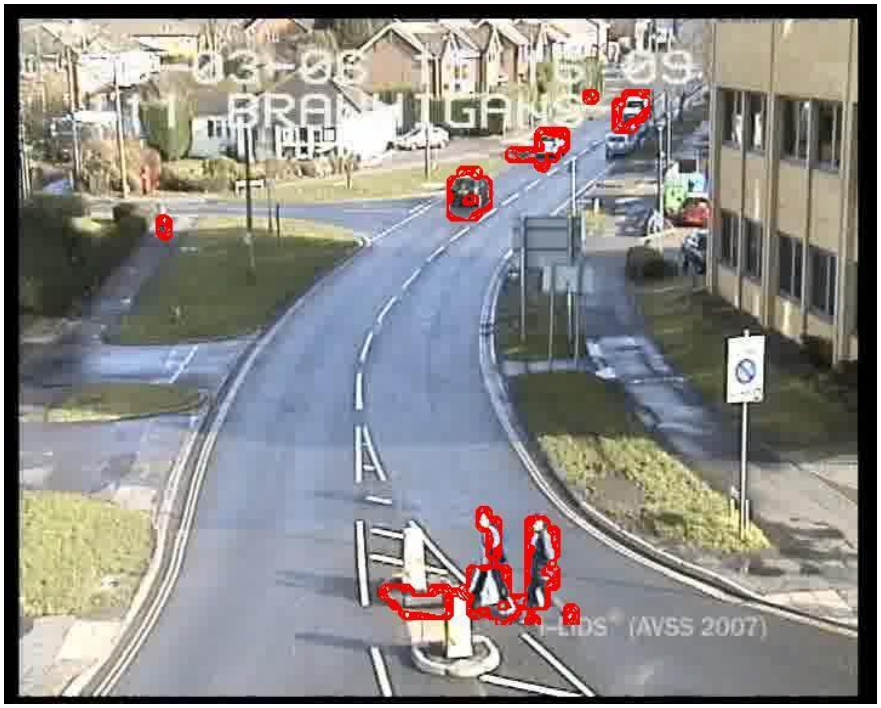

图 4.22　第 93 帧标记后的图像

图 4.23　第 123 帧标记后的图像

从实验结果可以看出，用基于高斯混合模型的背景差分法来进行运动目标检测能较好地应对较为复杂的动态背景环境，检测到的车辆轮廓也相对完整。本次实验的检测结果中，车辆的阴影和行人也被视作运动目标，说明传统的基于高斯混合模型的背景差分法的算法仍需进一步优化。

结　论

由于在当今社会，基于图像、视频处理技术的智能视频监控技术日益成熟，运动目标检测作为实现其智能化的重要基础也越来越被人们所重视。运动目标检测有着深远的研究意义和广泛的应用，是当前的一个热门研究课题。

本文介绍了光流法、帧间差分法、背景差分法这三个常见的运动目标检测方法，对它们的实现方式进行了研究，并分析了它们各自的优缺点。光流法通过对图像序列中光流场的计算，得到图像中运动目标前景的位置。使用光流法能够无需提前去获取背景环境的任何有关信息，直接就可以进行运动目标检测，但光流法计算复杂耗时，实时性差，且对光流场的精确计算十分容易受到动态背景环境中各种干扰因素的影响。帧间差分法利用在相邻帧图像中对应像素点之间的灰度值变化这一信息来对运动物体目标进行检测，该方法计算简单，不过于依赖设备性能，实时性强，能一定程度上适应动态背景环境中的变化，但在适应背景环境中的噪声干扰时过于依赖阈值的设置，且对于运动目标物体的移动速度和颜色分布有着较为苛刻的限制，最终检测到的运动目标轮廓常常不够完整。背景差分法则是通过使用背景建模的方法以得到与当前待检测的那一帧原始图像所适配的背景图像，再用该帧原始图像与背景图像进行差分运算，最终得到运动目标前景。背景差分法的算法也相对比较简单，有着实时性好的优势，根据具体的应用场景，选择合适的背景建模方法以应对各种动态背景环境，保证运动目标检测的精度。

在进行背景建模时，单高斯模型适用于背景部分为背景单模态背景时的情形，而在背景图像像素点的像素值呈现多峰分布，实际应用时背景环境较为复杂的情况下，则应该使用高斯混合模型。通过进行 MATLAB 仿真实验发现，对于背景建模而言，高斯分布的参数更新是重中之重，参数更新的正确性将对于建立能实时且正确地对当前帧图像的背景进行表示的背景模型起到重要影响。

参考文献

[1] 谭文明. 复杂背景建模方法及其在运动目标检测中的应用[D]. 安徽: 中国科学技术大学, 2009.

[2] 余启明. 基于背景减法和帧差法的运动目标检测算法研究[D]. 江西: 江西理工大学, 2013.

[3] 何银飞. 基于改进的帧差法和背景差法实现运动目标检测[D]. 河北: 燕山大学, 2016.

[4] 白一帆. 智能视频监控中运动目标检测识别方法研究[D]. 山西: 太原理工大学, 2018.

[5] 杨旗, 程鹏. 一种融合改进帧间差的 vibe 算法[J]. 信息技术与信息化, 2020, (9): 68- 70.

[6] 邱藤. 基于高斯混合模型的EM算法及其应用研究[D]. 四川: 电子科技大学, 2015.

[7] 刘宏飞, 杨耀权, 杨雨航. 基于改进光流特征的运动目标跟踪[J]. 计算机与现代化, 2021, (3): 115- 121.

[8] 李岩. 固定场景中运动目标检测与运动估计跟踪[D]. 河南: 郑州大学, 2016.

附　　录

附录 A

Improved Gaussian Mixture Models for Adaptive Foreground Segmentation

Abstract

Adaptive foreground segmentation is traditionally performed using Stauffer and Grimson’s algorithm that models every pixel of the frame by a mixture of Gaussian distributions with continuously adapted parameters. In this paper we provide an enhancement of the algorithm by adding two important dynamic elements to the baseline algorithm: The learning rate can change across space and time, while the Gaussian distributions can be merged together if they become similar due to their adaptation process.We quantify the importance of our enhancements and the effect of parameter tuning using an annotated outdoors sequence.

1 Introduction

Adaptive foreground segmentation is the cornerstone of many computer vision applications, since it allows a system to concentrate on the important segment of a visual scene. A typical example is tracking systems that begin by segmenting the foreground in order to alleviate scene clutter.

Foreground segmentation can be based on current motion as expressed by frame-byframe difference, but occasional motion stops of foreground objects render it useless. Static foreground segmentation can be used in very controlled environments, where the foreground is determined by subtracting a pre-calculated background image from the current frame. However, foreground-free initial segments and static environments are rarely found in practice. To overcome these problems, an adaptive algorithm that models every pixel of the frame by a mixture of Gaussian distributions is commonly used [7]. The Gaussians are learnt from the evolving scene using a learning rate. While this approach shows improved performance, it requires a selection of the learning rate. Small rates cannot cope with fast illumination changes, while large rates fail when slow moving foreground patches are present.

To overcome these difficulties spatio-temporal adaptation of the learning rate has been proposed. Rate reduction for slowly moving foreground has been considered in [5]. Rate increase for illumination changes has been used in [3].

More flexibility in the description of the scene has been added in [10] by adaptively selecting the number of Gaussians used to model each pixel: the mixture elements are updated recursively.

We propose an extension of the adaptation of the rate and the number of Gaussians, that performs well when there is slowly moving foreground, camera flicker and sudden changes in the lighting conditions. We employ a learning rate that changes (both increases and decreases) on a per-pixel basis (space) and a per-frame basis (time), reducing the negative effects of the above conditions. Furthermore, we propose a Gaussian merging algorithm that improves the dynamic modelling capability of GMMs, offering a considerable boost to the performance.

2 Adaptive Foreground Segmentation

The chosen foreground segmentation algorithm is a variant of Stauffer and Grimson's [7] adaptive foreground estimation algorithm. According to this, a model is built for every pixel in the frame. Gaussian mixture models (GMMs), comprising nGMM Gaussians each, model the different colours every pixel can receive in a video sequence. We work on the YCbCr colour-space, assuming the three components independent for simplicity. The weight of each Gaussian in the mixture is proportional to the time that particular Gaussian models best the colour of the pixel. By manipulating these weights we estimate the foreground patches in the image and we reconstruct a background frame. In the following sub-sections the algorithm is detailed.

2.1 Initialising Gaussians

The first frame of the video is used to initialize the first Gaussian for every pixel. The mean value of the Gaussian is the vector describing the colour (YCbCr) triplet of the pixel. The variance is initialised to some moderate initial value σ_{init}^2 t (the same for every colour component), to allow colour triples with subtle differences occurring in a given pixel, as it appears in the upcoming frames, to still be modelled by the same Gaussian.

Gaussians are also initialised when a colour triple does not match the models for the pixel. In this case the next empty slot of the model is used. If the maximum number of Gaussians in the model is already reached, then the newest Gaussian (the one with the smallest weight) is deleted and the new Gaussian is initialised in its place.

2.2 Matching and Updating Gaussians

For every new frame, the colour triple of every pixel is checked against the GMM of that pixel for possible matching. For this the Mahalanobis distance between every one of the

nGMM Gaussians in the model and the triple is calculated and compared to a threshold value dGMM. The Gaussian with the smallest distance that is below d_{GMM} is selected as the matching one (the one describing the new triple well enough and certainly better than any other in the GMM) and is updated.

Gaussian updating involves the increase of its weight based on the current learning rate of the pixel, and the decrease of all other weights in the pixel GMM. The learning rate $a(x,y,t)$ depends on the pixel coordinates (x,y) and the frame time t. The weight update for a matched Gaussian is then:

$$w(x,y,t)=[1-a(x,y,t)]w(x,y,t-1)+a(x,y,t) \tag{1}$$

and for a non-matched Gaussian:

$$w(x,y,t)=[1-a(x,y,t)]w(x,y,t-1) \tag{2}$$

The above equations are similar to the ones presented in [7], adding our proposed spatiotemporal adaptation of the learning rate. After weight update, the Gaussians in every pixel GMM are sorted in descending order based on their weights. The weights are then renormalised to unity.

The mean and variance of the matched Gaussian are also updated similar to the corresponding parameters presented in [7]:

$$\mu(t)=(1-\rho)\mu(t-1)+\rho X(t) \tag{3}$$

$$\sigma^2(t)=\min\left\{\sigma_{\min}^2,(1-\rho)\sigma^2(t-1)+\rho[X(t)-\mu(t)]^T[X(t)-\mu(t)]\right\} \tag{4}$$

where ρ is the learning factor for updating existing Gaussians given by [4]:

$$\rho=\alpha(x,y,t)\eta(X(t)|\mu_k,\sigma_k) \tag{5}$$

Mean update ensures that Gaussians are following slow changes in the appearance of the pixel. Variance update leads to narrow models for triples that do not vary significantly,while allowing wider models for triples that vary a lot but in small steps. Note that, in addition to [7], the variances are not allowed to drop below a threshold $\sigma_{\min}^2$.

The learning rate $a(x,y,t)$ is adapted per pixel and time step. While in [5] it is only decreased to protect slowly moving foreground and in [3] it is increased to cope with sudden illumination changes, here we adapt the rate as follows:

- Increase rate to learn background flicker.
- Increase rate globally for global illumination changes.

•Increase rate locally for local illumination changes, like moving clouds.

•Decrease rate locally for slowly moving foreground.

Flicker adaptation is straightforward: very small foreground patches at pixels (x, y) are considered flicker, in which case the rate is scaled by $a_f > 1$:

$$a(x, y, t) = a_f a(x, y, t-1) \tag{6}$$

Adaptation for global illumination changes is initiated when the foreground suddenly covers most of the scene, or the frame-by-frame difference is excessive, in which case the rate of all pixels is scaled by $a_i > 1$.Local illumination changes are handled by scaling the rate of only the affected pixels, which are detected as foreground blobs that increase in area very fast.

Adaptation for target protection is only possible if the system has a tracker that validates the foreground patches as targets and provides their speed. In that case the rate is scaled by a factor $a_p(s, v)$ that depends on the target's size s and speed v as detailed on [5]. In short, fast targets employ the normal learning rate, while slow targets have a much reduced learning rate. Furthermore, if the target is too large the rate is reduced further by a factor of 0.25, to handle large vehicles, where their speed can be large, but their uniform colours can make them fade into the background.

The modified learning rates persist across time: At every frame after the rate scaling, if there is no reason for further scaling at any particular pixel, its learning rate is gradually changed towards the default value.

2.3 Merging Gaussians

In the original Gaussian mixture algorithm, the number of Gaussians in the per pixel GMMs is fixed to nGMM. More Gaussians allow for finer modelling, but slower operation.We add a dynamic modelling capability by merging Gaussians which, although at initialisation were quite apart, have been drifting closer in the colour-space via the Gaussian adaptation process. Gaussians whose means have Euclidean distance less than a threshold dmerge are merged together. The new Gaussian has a weight equal to the sum of the weights of the two merged Gaussians. The mean and variance of the new Gaussian is the weighted average of the individual means and variances of the merged Gaussians.

The merged Gaussian has an increased weight, so two similar versions of the background are now actually tagged by the algorithm as such much faster. Additionally, many Gaussians with low weight that actually belong to the background may be merged with the closest

background Gaussian, reducing the false positives. Finally the released Gaussian slot allows the introduction of a new member in the mixture, without deleting the one with the least weight, effectively increasing the number of available Gaussians.

2.4 Foreground Detection from the Gaussian Mixture Pixel Model

Given the GMM for all the pixels, the Pixel Persistence Map (PPM) IPPM can be built, in which every pixel is represented by the weight of the Gaussian from its GMM that best describes its current colour triple. Regions of the map with large values correspond to pixels that have colours that appear there for a long time, hence belong to the background. On the contrary, regions with small values correspond to pixels that have colours that appear there for a short time, hence belong to the foreground. The unfiltered foreground pixels are those with accumulated model weights for the Gaussians with larger weights than the matching one that is above a threshold s. These foreground pixels are subjected to shadow removal [9] and morphological clean-up to obtain the foreground image Ifrg.

The shadow removal employed is performed only on the pixels that have been classified as foreground in the previous step. Each unfiltered foreground pixel is compared to the background estimation at the same location. The current background pixel in these locations is estimated as the mean of the Gaussian with the largest weight. Shadow removal is controlled by two tolerances: the tolerance to brightness distortion BD and to colour distortion CD. Unfiltered foreground pixels that exhibit too large brightness distortion, i.e. discrepancy from the background image luminance, are checked for colour distortion, i.e. discrepancy from the background image chrominance. If they exceed both distortion tolerances, they are marked as shadow, therefore larger BD and CD tolerances lead to fewer pixels being labelled as shadow.

3 Results

To quantify the performance of the system and the effect of the various parameters introduced in Sect. 2, we have annotated several frames from two video sequences of outdoor cameras overlooking a city square. The first sequence depicts the formation of a demonstration that begins with normal midday occupancy of the square and ends with a large crowd being gathered there. It is captured from a camera fixed on a second floor. The second captures the same square and the street after it during a normal day from a handheld camera at a first floor window. Annotation involves marking every foreground pixel in the sequence. Examples of the annotated frames can be seen in Fig. 1 for the first sequence and Fig. 2 for

the second one. In both figures, a visual comparison to the detected foreground is also performed.

In the following sub-sections we evaluate the performance of our foreground estimation algorithm on a per pixel level. We begin by introducing our evaluation metrics and compare our system to the baseline implementations. We then proceed in evaluating the effect of the various parameters introduced in Sect. 2.

3.1 Evaluation Metrics and Comparison to Baseline

Foreground estimation is actually a two-class classification problem: every pixel is classified in the foreground or the background class. As such, we employ the established metrics of precision, recall, and their harmonic mean in the F1-score as a single-number performance indicator combining both precision and recall [6]. Some definitions are due:

•True positives TP: The pixels that are classified as foreground and are indeed foreground.

•False positives FP: The pixels that are classified as foreground but are actually background. This is one possible error that system can make.

•False negatives FN: The pixels that are classified as background but are actually foreground, or the missing foreground pixels. This is the other possible error that system can make.

•Precision prec (positive predictive value): The ratio of correctly classified foreground pixels over the total pixels classified as foreground

$$prec = \frac{TP}{TP + FP} \tag{7}$$

•Recall rec (true positive rate or sensitivity): The ratio of correctly classified foreground pixels over the sum of the correctly classified and the missing foreground pixels

$$rec = \frac{TP}{TP + FN} \tag{8}$$

•F1-score: Usually when trying to improve one of the precision or recall, the other worsens. It is easier to find a single figure of merit called F1-score, defined as the harmonic mean between precision and recall

$$F1 = \frac{2 * prec * rec}{prec + rec} \tag{9}$$

The baseline system against which we compare our algorithm is the plain Stauffer implementation, where there is no learning rate adaptation, neither Gaussian merging. On top

of that we add the proposed improvements, excluding the reduction of rate for target protection, as the system under consideration did not employ a tracker.

The resulting performance is shown in Table 1. The performance increase over the baseline is 158 %, mainly achieved by reducing the false positives. The results indicate that either rate adaptation (to conditions and flicker) or Gaussian merging can improve performance significantly, with the latter being significantly more effective. Also, the local rate adaptation for learning the flicker pixels is more important than the global rate adaptation for learning the change of imaging conditions. However, the effect of global rate adaptation should not be underestimated, as it is much more important when coping with larger data sets, such as the one available from the Background Models Challenge (BMC2012) [8].

中文译文 A

改进的自适应前景分割高斯混合模型

摘要

自适应前景分割传统上是使用 Stauffer 和 Grimson 的算法进行的，该算法通过具有连续适应参数的高斯分布的混合来模拟帧的每个像素。本文通过在基线算法中加入两个重要的动态元素，对算法进行了改进：学习速率可以在空间和时间上变化，而高斯分布由于其自适应过程而变得相似，则可以合并在一起。我们量化了我们的增强的重要性和使用带注释的户外序列进行参数调优的效果。

1 引言

自适应前景分割是许多计算机视觉应用的基石，因为它允许系统集中在视觉场景的重要部分，一个典型的例子是跟踪系统，它从分割前景开始，以减轻场景的混乱。

前景分割可以基于帧间差分表示的当前运动，但前景对象的偶尔运动停止会使其失效。静态前景分割可用于非常受控制的环境，其中前景是通过从当前帧中减去预计算的背景图像来确定的。然而，无前景的初始段和静态环境在实际中却很少出现。为了克服这些问题，通常使用一种自适应算法，通过混合高斯分布对帧中的每个像素建模[7]。高斯人通过学习率从进化场景中学习。虽然这种方法显示出改进的性能，但它需要选择学习率，小速率无法应对快速的光照变化，而大速率则会在存在缓慢移动的前景面片时失败。

为了克服这些困难，人们提出了学习速率的时空适应性。在文献[5]中考虑了慢速移动前景的速率降低。增加光照变化的速率已在[3]中使用。

自适应地选择用于对每个像素建模的高斯数，文献[10]增加了场景描述的灵活性：更新了混合元素递归地。我们提出了一种扩展的高斯速率和高斯数的自适应算法，在前景缓慢移动的情况下性能良好，相机闪烁和照明条件的突然变化。我们采用每像素（空间）和每帧（时间）变化（增加和减少）的学习速率，减少上述条件的负面影响。此外，我们还提出了一种高斯合并算法，提高了 GMMs 的动态建模能力，大大提高了 GMMs 的性能。

本文的剩余部分如下：第二章简要讨论了自适应前景估计算法，重点讨论了与基线相比的差异。第三章介绍了评价指标，并对每一个改进方案的性能进行了量化。第 3.2 节详细说明了调整各种参数的效果，第四章给出了结论。

2 自适应前景分割

所选择的前景分割算法是 Stauffer 和 Grimson 的[7]自适应前景估计算法的变体。据此，为帧中的每个像素建立一个模型。高斯混合模型（GMMs）由 n_{GMM} 高斯子组成，对视频序列中每个像素可以接收的不同颜色进行建模。我们在 YCbCr 颜色空间中工作，假设这三个分量独立于简单性。混合物中每个高斯的权重与特定高斯模型最佳像素颜色的时间成正比。通过控制这些权重，我们估计出图像中的前景块，并重建出一个背景帧。在下面的小节中详细介绍了算法。

2.1 初始化高斯

视频的第一帧用于初始化每个像素的第一高斯。高斯平均值是描述像素的颜色（YCbCr）三元组的向量。方差被初始化为一些中等的初始值 σ_{init}^2（每个颜色分量相同），以允许在给定像素中出现细微差异的颜色三元组（如在即将到来的帧中出现的）仍然由相同的高斯模型建模。

当颜色三元组与像素的模型不匹配时，高斯也会初始化。在这种情况下，将使用模型的下一个空插槽。如果模型中的高斯数已达到最大值，则删除最新的高斯（具有最小权重的高斯）并在其位置初始化新的高斯。

2.2 高斯的匹配与更新

对于每一个新的帧，每个像素的颜色三元组都会根据该像素的 GMM 进行检查，以便进行可能的匹配。为此，计算模型中每一个 n_{GMM} 高斯子与三个 n_{GMM} 高斯子之间的马氏距离，并与阈值 d_{GMM} 进行比较。选择距离小于 d_{GMM} 的高斯函数作为匹配函数（描述新三重态阱的高斯函数，它比 GMM 中的任何一个都好），并进行更新。

高斯更新包括基于像素的当前学习率增加其权重，以及像素 GMM 中所有其他权重的减小。学习速率 $a(x,y,t)$ 取决于像素坐标 (x,y) 和帧时间 t。然后，匹配高斯的权重更新为：

$$w(x,y,t)=[1-a(x,y,t)]w(x,y,t-1)+a(x,y,t) \tag{1}$$

对于非匹配的高斯：

$$w(x,y,t)=[1-a(x,y,t)]w(x,y,t-1) \tag{2}$$

上述方程与文献[7]中的方程相似，增加了我们提出的学习速率的时空适应性。权值更新后，每个像素 GMM 中的高斯子根据权值按降序排序。然后将权重重新标准化为统一。

匹配高斯的均值和方差也会更新，与[7]中给出的相应参数类似：

$$\mu(t)=(1-\rho)\mu(t-1)+\rho X(t) \tag{3}$$

$$\sigma^2(t)=\min\left\{\sigma_{\min}^2,(1-\rho)\sigma^2(t-1)+\rho[X(t)-\mu(t)]^T[X(t)-\mu(t)]\right\} \tag{4}$$

其中 ρ 是更新由[4]给出的现有高斯数的学习因子：

$$\rho=\alpha(x,y,t)\eta(X(t)|\mu_k,\sigma_k) \tag{5}$$

均值更新可以确保高斯数跟踪像素外观的缓慢变化。方差更新导致三元组的模型变窄，变化不明显，同时允许三元组的模型变窄，变化很大，但步骤很小。注意，除了[7]之外，方差不允许降到阈值 $\sigma_{\min}^2$ 以下。

每像素和时间步调整学习速率 $a(x,y,t)$。在[5]中，它只是为了保护缓慢移动的前景而减小，而在[3]中，它是为了应对突然的光照变化而增大的，这里我们将速率调整如下：

•提高学习背景闪烁的速率。

•提高全局照明变化的全局速率。

•提高局部照明变化的局部速率，如移动的云。

•降低缓慢移动前景的本地速率。

闪烁适应很简单：像素 (x,y) 处的非常小的前景块被视为闪烁，在这种情况下，速率按 $a_f>1$:

$$a(x,y,t)=a_f a(x,y,t-1) \tag{6}$$

当前景突然覆盖了大部分场景，或逐帧差异过大时，开始对全局照明变化的自适应，在这种情况下，所有像素的速率由 $a_i>1$ 缩放。局部照明变化通过仅缩放受影响像素的速率来处理，它们被检测为前景斑点，面积增长非常快。

只有当系统有一个跟踪器来验证前景补丁作为目标并提供其速度时，才有可能对目标保护进行自适应。在这种情况下，速率按系数 $a_p(s,v)$ 缩放，该系数取决于目标的大小 s 和速度 v，详见[5]。简而言之，快速目标采用正常的学习率，而慢速目标则大大降低

了学习率。此外，如果目标太大，则速度将进一步降低 0.25 倍，以处理速度可能较大但颜色均匀的大型车辆，使其褪色到背景中。

修改后的学习速率随着时间的推移而持续：在速率缩放后的每一帧，如果没有理由在任何特定像素上进一步缩放，则其学习速率将逐渐变为默认值。

2.3 合并高斯

在原始的高斯混合算法中，每像素 GMMs 中的高斯数固定为n_{GMM}。高斯数越多，建模越精细，但速度越慢行动。我们通过合并 Gaussians 添加动态建模功能，虽然在初始化时，Gaussians 是完全分开的，已经通过高斯适应过程在色彩空间中漂移得更近了。平均欧氏距离小于阈值d_{merge}的高斯数合并在一起。新高斯函数的权重等于两个合并高斯函数的权重之和。新高斯的均值和方差是合并高斯的个体均值和方差的加权平均值。

合并后的高斯有一个增加的权重，因此两个相似版本的背景现在被算法标记得更快。此外，许多实际属于背景的低权重高斯子可以与最接近的背景高斯子合并，减少误报。最后，释放的高斯时隙允许在混合物中引入新成员，而不删除具有最小权重的成员，有效地增加可用高斯数。

2.4 基于高斯混合像元模型的前景检测

具有较大值的地图区域对应于具有在那里出现很长时间的颜色的像素，因此属于背景。相反，具有较小值的区域对应于具有在那里出现很短时间的颜色的像素，因此属于前景。未过滤的前景像素是那些具有累积模型权重的高斯像素，其权重大于阈值以上的匹配像素。这些前景像素经过阴影去除[9]和形态学清理以获得前景图像I_{frg}。

所采用的阴影去除仅对在上一步骤中被分类为前景的像素执行。将每个未滤波的前景像素与相同位置的背景估计进行比较。这些位置的当前背景像素被估计为具有最大权重的高斯分布的平均值。阴影去除由两个公差控制：亮度畸变公差 *BD* 和颜色畸变公差 *CD*。如果未经过滤的前景像素显示出过大的亮度失真，即与背景图像亮度的差异，则检查颜色失真，即与背景图像色度的差异。如果它们超过两个失真公差，则标记为阴影，因此较大的 *BD* 和 *CD* 公差会导致标记为阴影的像素较少。

3 结果

量化系统的性能以及第 2 节中介绍的各种参数的影响。我们从两个俯瞰城市广场的室外摄像机视频序列中注释了几个帧。第一个序列描述了示威游行的形成，从中午广场

的正常占用开始，到聚集在那里的一大群人结束。它是从固定在二楼的摄像机上拍摄的。第二张照片是在平常的一天里，从一楼窗户的手持式摄像机拍下同一个广场和之后的街道。注释包括标记序列中的每个前景像素。注释帧的示例可以在图 1 中看到第一序列，图 2 可以看到第二序列。在这两个图中，还执行与检测到的前景的视觉比较。

在下面的小节中，我们将在每像素级别上评估前景估计算法的性能。我们首先介绍我们的评估指标，并将我们的系统与基线实现进行比较。然后我们继续评估第 2 节中介绍的各种参数的影响。

3.1 评估指标和与基线的比较

前景估计实际上是一个两类的分类问题：每个像素在前景或背景类中进行分类。因此，我们将 F1 分数中的精确度、召回率及其调和平均值作为一个单一的数字绩效指标，将精确度和召回率结合起来[6]。一些定义如下：

•正片 TP：被分类为前景的像素，实际上是前景。

•误报 FP：分类为前景但实际上是背景的像素。这是系统可能犯的一个错误。

•假阴性 FN：被分类为背景但实际上是前景的像素，或丢失的前景像素。这是系统可能产生的另一个错误。

•精密度 prec（阳性预测值）：正确分类的前景像素与分类为前景的总像素的比率

$$prec = \frac{TP}{TP + FP} \tag{7}$$

•召回率（真阳性率或灵敏度）：正确分类的前景像素与正确分类和丢失前景像素之和的比率

$$rec = \frac{TP}{TP + FN} \tag{8}$$

•F1 分数：通常当试图提高一个精度或回忆时，另一个会恶化。更容易找到一个称为 F1 分数的单一价值指标，定义为精确性和召回率之间的调和平均值

$$F1 = \frac{2 * prec * rec}{prec + rec} \tag{9}$$

我们比较算法的基线系统是普通的 Stauffer 实现，没有学习速率自适应，也没有高斯合并。除此之外，我们还增加了建议的改进，不包括降低目标保护率，因为考虑中的系统没有使用跟踪器。

结果性能如表 1 所示。性能比基线提高了 158%，主要是通过减少误报来实现的。结果表明，无论是速率自适应（条件和闪烁）还是高斯合并都能显著提高性能，后者更有效。此外，用于学习闪烁像素的局部速率自适应比用于学习成像条件变化的全局速率自适应更重要。然而，不应低估全球利率调整的影响，因为在处理更大的数据集时，如背景模型挑战赛（BMC2012）[8]提供的数据集时，全球利率调整更为重要。