封面页

# 摘要

社会经济不断增长，人们的生活水平逐渐提高，不断促进着监控技术和网络技术的革新，高清网络监控摄像头也被广泛应用在各个场所中，以满足社会各方面的安防需求。如今网络远程监控已经被普及，摄像头数量也日益增长，伴随而来的是海量监控视频，但是却无法找到足够的人力资源和时间资源对这海量视频数据进行处理。将视频监控智能化、自动化成为当前监控领域的迫切需求。

本文针对上述海量监控视频的问题，开发基于视频摘要的视频内容检索系统，给出了一种快速的生成视频摘要方法，同时支持实时监控视频和本地监控视频处理。视频摘要是监控视频的浓缩，提取视频中有意义的部分，然后浓缩成一个很短的摘要，让监控人员能用最短的时间看完一个长的监控视频。视频内容检索是在提取视频摘要的基础上，对浓缩的监控视频的事件进行简单的特征搜索，避免了对整个视频的分析，大大缩小了视频内容检索时间。

视频摘要算法包括运动物体检测、运动物体跟踪以及摘要合成三个部分，先用帧间差分法检测出运动物体的矩形轮廓，然后用矩形轮廓匹配对每个运动物体进行轨迹跟踪，最后遍历每一个运动事件生成视频摘要。视频内容检索主要是对一些简单特征进行检索，如事件发生时间，对象的运动方向，入侵区域，对象的颜色等。

本文先对基于视频摘要的视频内容检索系统进行功能介绍，描述其适用场景范围和存在的问题；然后对系统整体流程和各部分算法进行详细介绍；最后对系统的实验结果进行分析，提出不足之处和可以改进的地方。

**关键字：视频摘要，视频内容检索，图像特征提取，运动物体检测与跟踪**

# Abstract

With the growing social-economic and and the improving of people’s lives, the monitoring technology and the network technology is developing fast, and the HD network surveillance cameras have been widely used in various places to meet the security needs of all sectors of society. While the number of cameras is also growing, producing a large number of surveillance video, but we do not have so many people and time to deal with this massive video data. Intelligent and automatic video surveillance has become a urgent need.

To solve such problems, this paper develop a video content search system base on video abstract. This system provides a method for generate video abstract quickly, and supports real-time video or lacal video processing. Abstract is the inspissation of a video, we extract the meaningful parts of the video, and then condensed them into a short summary, so that the monitoring personnel can spend less time to watch a long video. Video content search is based on the video abstract, it just search the meaningful part of the video, thus we can save a lot of time to search.

# 目录

1. **绪论**
   1. **背景与意义**

随着社会的迅速进步和科学技术的发展，安防监控在软硬件层面上都在不断地革新，安防监控摄像头也被广泛应用于社会中的每个角落。现在低分辨率低码率的摄像头已经不能满足监控的需求，高清监控摄像头是安防监控的必然趋势，高清监控视频对于视频取证，人物识别，交通监控等有着重大意义。由于监控系统己经基本覆盖社会的各个角落，其所产生的海量视频数据无法有效进行有效的管理和查看。实际的监控任务仍需要较多的人工完成，而且现有的视频监控系统通常只是录制视频图像，也有一些厂家提供了一些自动监控特定事件的软件，但实际应用较少。一般的监控视频只能用作事后取证，而且人工浏览监控视频进行取证需要花费很长时间。

随着高清监控摄像头的不断普及，监控视频将在取证和特定目标识别等方面发挥重大作用。面对海量的监控视频，智能监控也是必然趋势，将智能监控应用到实时高清摄像头中成为了当务之急。

针对上述提到的问题，我提出了对应的解决方案，即基于摘要的监控视频信息检索系统,该系统先在实时监控阶段对实时视频流进行摘要分析（摘要分析也可以对离线视频进行处理），然后在事后取证查找过程再加入自动检索功能。视频摘要是视频内容的一个浓缩，去除无用部分，只保留有意义的部分。先进行视频摘要处理，可以为视频检索过程节省大量时间。当需要浏览监控视频时可以只浏览视频摘要，或者根据对象的特征进行搜索，快速找到监控视频内容。

视频摘要的目的是提取视频中有意义的内容（即运动事件），因此此系统不适用于人来人往的闹市，或者有很多目标固定存在的场所（比如说大型办公室等）。本系统可以应用于出入口（比如说公司大门）监控，小区道路监控，非塞车时的交通道路监控，别墅监控等。

* 1. **国内外相关研究和产品概况**
     1. **相关研究概况**

本文研究主要涉及到的关键技术有运动物体检测，运动物体跟踪，图像特征提取等。

1. **运动物体检测**

运动物体检测就是在视频中将感兴趣的运动前景和背景分割开来，提取我们最感兴趣的运动目标。经过人们的多年研究，至今也有很多种运动物体检测的方法，但主要可以分为三大类：背景差分法，帧差法和光流法。

**背景差分法**[4,7]是在背景静止的前提下进行运动物体检测的方法，先对视频进行背景建模，然后用当前帧图像与背景图像进行相减运算，从而得到运动物体。这种方法的关键是背景建模。背景差分法操作简单，其检测结果依赖于背景模型的质量，受环境干扰（光照，摄像头抖动等）的影响较大，需要添加背景更新机制才能得到较好的检测结果。

在文献[4]中，Jin Chengjun等人采用了一种基于YCbCr颜色空间的直方图背景提取方法，减少了光照和阴影带来的影响；在文献[7]中，[Bouttefroy, P.L.M.](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=p_Authors:.QT.Bouttefroy,%20P.L.M..QT.&searchWithin=p_Author_Ids:37402872100&newsearch=true)则是对混合高斯建模法的单一学习速率进行改进，区分两种学习率，对方差施加约束，改进背景建模的效果。

**帧差法**[5,10]是对视频图像序列相邻帧进行差分，利用连续图像序列之间的强相关性进行运动物体检测。帧差法计算简单，而且受环境的干扰很小，能适应多种复杂的环境，但是往往不能够提取完整的运动物体，只能够检测到运动物体的边缘轮廓，在运动物体内形成“空洞”。

在文献[5]中，Yang Shu-Ying等人采用帧差法对运动物体进行初步定位；在文献[10]中，提出一种基于三帧差分法和边缘信息的余地弄物体检测方法，能准确检测目标，有较好的鲁棒性。

**光流法**能够很好的处理可动背景中运动物体检测与跟踪，目前经典的光流算法有Lucas-Kanade算法[6]和Horn-Schunck算法[8]，但是光流法的计算量都很大，不能用于实时视频的分析中。

1. **运动物体跟踪**

运动物体跟踪是在运动物体检测的基础上，然后根据目标的某个或者某些特征，使用合适的匹配算法，在连续图像序列中寻找相同的运动物体，最后得到物体的完整运动过程。运动物体跟踪算法也有很多种，最简单的有基于模板匹配的运动物体跟踪[3]，但是这种方法效率低下，不适合用在实时监控跟踪中；基于颜色直方图的算法有Camshift算法[14]，文献[14]运用Camshift算法来实现了人脸跟踪，这种算法的缺点是要手动选取被跟踪的区域；也有基于区域的跟踪算法等。一般来说，现在还没有一个算法能运用于每一种场景的运动物体跟踪中，算法选择非常关键。

1. **图像特征提取**

图像特征提取是一个比较范的领域，图像特征包括形状、颜色、纹理、场景、类别等等，本文主要研究颜色和类别两种特征。颜色特征提取又有颜色直方图[17]、颜色集、颜色矩等方法

* + 1. **相关产品概况**

1. **BriefCam**

BriefCam是第一个成功商业应用的视频摘要软件，由耶路撒冷希伯来大学凭借其技术实力研发。BriefCam实现了以前全球对于视频查看未能满足的需求，即快速地调查和查明事件并采取行动。BriefCam视频摘要技术使用户可以用一分钟的时间回顾24小时内发生的事件，将数小时的事件压缩成“摘要”，仅用几分钟来查看。其客户包括警察、军队、边检及其他国土安全机构，以及政府安全部门、机场、铁路、海运及其他运输部门、银行、办公楼管理、零售商及更多。

1. **华尊视频摘要**

华尊视频摘要分析技术，采用基于视频对象的技术原理。追踪和分析视频中的活动对象，并提取对象的运动区域、颜色、大小等信息，从而建立与原始视频的索引关系。通过构建视频摘要剪辑，显示完整的视频内容，可将24小时内发生的所有事件以浓缩短片的形式，在短短几分钟内完整显示出来。华尊视频摘要最大的特点就是增加了一些对象特征提取的功能。

1. **QBIC系统**

QBIC（Query By Image Content）由IBM Almaden研究中心所开发，是“基于内容”视频检索的典型代表。此检索系统利用形状、纹理、颜色和对象运动等特征来描述视频内容，并在此基础上实现视频内容检索。QBIC同时提供了基于内容的视频信息检索手段和静止图像信息检索手段，用户可以使用颜色特征、纹理特征、例子图像、草图和对象运动等信息对视频或者图像进行检索。在视频内容分析方面运用了运动估计、代表帧生成、镜头检测、层描述等多种分析技术。

1. **Visual Seek 系统**

Visual Seek是美国哥伦比亚大学实现的在互联网上使用的基于内容的检索系统。它实现了互联网上的基于内容的视频或者图像检索系统，提供了一个供用户能在互联网上搜索和检索视频以及图像的工具。

* 1. **智能监控系统的市场预测和发展趋势**
     1. **智能监控技术不断革新**

从软硬件技术方面来看，首先是半导体工艺的发展，根据摩尔定律，处理器性能逐年升高，即是是计算量很大的算法也能很快的处理完；其次是视频图像处理技术也在不断的完善和革新，能够处理更为复杂和多变的场景，能够分析和识别更多的异常事件和行为，处理的时间复杂度也不断的优化降低；再次是监控设备也不断得到更新，高清视频监控也已经得到很好的应用。

* + 1. **不断发展的市场**

其实，对智能监控系统的需求一直存在，只是智能监控技术还不够成熟，目前智能视频监控的应用场景还比较有限，主要应用于一些特定的场合，但随着软硬件技术的不断进步，智能视频监控必将覆盖到每一个监控摄像头，甚至走进千家万户。

根据现在的发展趋势，安防监控将会在未来几年内基本实现城市高清监控，高清视频使得智能监控技术发挥更多的用武之地，之前由于视频分辨率太低图像质量太差等原因而导致无法分析的情况不复存在。

总之，智能安防监控的市场需求在不断扩大，应用领域也越来越广，甚至可以作为一个模块集成到数字家庭系统中。

* 1. **开发环境和开发工具简介**

本系统使用Qt Creator作为IDE和UI开发工具进行开发，同时用到了Opencv和FFmepg两个开源跨平台的工具。

* + 1. **Qt Creator**

Qt Creator是一个跨平台的IDE，同时支持Windows、Linux和Mac OS X系统，开发人员能用QT更方便快捷的完成开发任务。Qt Creator整合了跨平台的自动化构建系统：qmake与CMake，使用Qt Designer可以很方便快捷的设计和构建图形界面。此外，Qt Creator还具有对C++语言的完整表达式检查，上下文关联，代码不全，键入代码时的行间错误即时指示等功能。

* + 1. **Opencv**

OpenCV的全称是：Open Source Computer Vision Library。它是一个基于BSD许可证授权（开源）发行的跨平台计算机视觉库，可以运行在Linux、Windows和Mac OS操作系统上。它轻量级而且高效地实现了图像处理和计算机视觉方面的很多通用算法。

目前OpenCV广泛运用在人机互动、物体识别、图像分割、人脸识别、动作识别、运动跟踪、机器人等领域。

因为OpenCV中提供了很多方便高效的图像处理算法找到，特别是运动物体的检测与跟踪，故采用OpenCV 进行辅助开发。

* + 1. **FFmpeg**

FFmpeg是一个开源免费跨平台的视频和音频流方案。其包含了非常完善的视频解码技术，可以运行音频和视频多种格式的录影、转换、流功能，尤其还支持网络RTSP视频流的解码。使用FFmpeg工具可以很方便解决主流监控视频的解码问题。

* 1. **主要研究内容和论文主要结构**

本论文主要研究视频摘要和视频内容检索，视频摘要又可分为视频中运动物体检测与跟踪，视频摘要生成等；视频内容检索主要是在视频摘要的基础上根据运动对象的颜色、大小、轨迹、时间等特征进行检索。

本论文共分为六章：

第一章主要讨论了基于视频摘要的视频内容检索。主要是描述其背景和意义，对目前现有的相关技术和商业产品进行简单的介绍，简单的展望了智能监控领域的前景，对本文所采用的开发环境和开发工具进行介绍。

第二章是介绍基于视频摘要的视频内容检索系统中两大模块的主要流程，系统的性能指标，另外还给出了具体的开发环境和实验数据。

第三章主要研究了视频摘要的实现。针对实时监控视频，提出了一种快速生成视频摘要的方法。该方法可以应用于多种监控场景中，事件检出率高，速度快，满足高清网络摄像头的实时监控需求。

第四章主要研究了一些图像特征提取技术，包括颜色、形状、类别等。在利用特征进行视频内容检索的时候，采用逐个特征过滤的方式，从最易提取的特征开始过滤，使检索的效率最大化。

第五章给出了整个系统的架构的实现，然后对系统的运行测试结果进行详细的分析，指出不足之处和改进的方向。

第六章对论文作出了总结和展望。总结了论文的主要研究工作与创新之处，并指出其中的不足；对基于云处理的视频摘要和视频内容检索的发展方向进行了展望。同时描述了本人的下一步工作内容。

1. **系统主要流程和系统指标**
2. **系统主要流程**

基于视频摘要的视频内容检索系统分为两大模块，一个是视频摘要，另一个是视频内容检索。视频摘要模块是可以实时处理的（当然也支持本地视频流处理），就是说在监控过程中实时检测并跟踪每一个运动事件，然后每隔一定时间形成一个视频摘要文件（视频文件），并保存每一个运动事件的详细信息（发生时间，运动路径等）。视频内容检索模块顾名思义就是根据一定的特征检索视频中感兴趣的事件，由于视频内容检索往往都是事后取证进行，所以此模块并不是实时进行的。在视频摘要的基础上进行视频内容检索，可以过滤掉一大部分的无用信息，大大减少检索的时间。

**视频摘要模块**中需要先对监控视频流（或本地视频流）进行解码，但本文不对监控视频流的解码进行研究，其编解码过程可以非常简单的用开源工具ffmpeg和opencv来实现。视频摘要模块的流程如图2-1所示：

监控视频流解码

解码后的视频图像序列

运动物体检测

运动物体跟踪

记录运动事件详细信息

定时生成视频摘要文件

视频内容检索模块

视频摘要查看

查看单个事件

查看整个视频摘要

由ffmpeg和opencv实现

图2-1 视频摘要模块的流程图

运动物体检测是视频摘要模块中最关键的一步，因为这会直接影响到后面运动物体跟踪的效果，本文将在第三章详细介绍运动物体检测算法，第四章是运动物体跟踪和视频摘要生成。

**视频内容检索模块**需要视频摘要作为基础，先是用户输入特定的检索特征（可以是一个也可以是多个），然后系统根据检索条件便利每一个运动事件，分析并提取运动事件的特征，如果与用户输入的特征相匹配，则作为检索结果输出此运动事件。如果用户输入的特征是组合特征，那就采用逐个特征过滤的方式，从耗时最少的特征条件开始过滤，逐级筛选，去除一定数量的运动事件，进一步缩短视频内容检索的时间。视频内容检索模块的主要流程如图2-2所示：

视频摘要模块

用户输入检索特征

特征1筛选

特征2筛选

特征n筛选

特征提取计算复杂度低的排前面，复杂度高的在后面

结果列表

**……**

图2-2 视频内容检索模块流程图

各种特征的筛选方式将在第五章详细介绍。

1. **系统指标**
2. **视频摘要模块**
3. 高清监控：能用在高清视频监控中，具体的说就是能支持分辨率为1280\*720，帧率为15帧/秒的高清监控视频流处理。
4. 实时性：要求视频摘要模块能够应用在实时高清视频监控中，能实时的检测和跟踪运动物体，记录运动事件，在一段监控视频保存后能快速生成视频摘要。
5. 准确性：运动事件的检出率在90%以上，运动事件是指一个物体从静止开始运动（或者视野外进入）到物体再次静止（或者离开视野）的过程。这里的检出率计算方法是：假设一个视频中的运动事件有n个，视频摘要模块中算法检测到的运动事件总数是k个，对同一个运动事件重复的记录有a条，则 。
6. 应用场景：视频摘要不可能应用到每一种监控场景中，视频摘要的宗旨就是提取监控视频中有意义的部分（运动事件），去除没意义的静止部分，达到快速查阅的效果，如果是闹市中人山人海或者道路塞车的监控视频，一来运动物体跟踪和检测难以做到，二来就算生成了视频摘要也可能比源监控视频大。因此，本文所研究的视频摘要主要应用在运动物体相对不多的安防监控场景，比如公司大门，非塞车公路路段，小区道路，小巷和楼梯间等相对不拥挤的地方。
7. **视频内容检索模块（未完待续）**
8. 多特征支持
9. **实验数据介绍**

本文研究的视频摘要和视频内容检索主要是以交通道路、公司大门、小区人行道等作为实验场景，主要运动对象是行人和车辆。数据来源主要是一些实际监控视频和少量自主拍摄视频，由于来源不同，视频分辨率都不尽相同。如下图2-3是交通部门拍摄的高清道路视频中的一帧，图2-4是某公司门口大堂的监控视频中的一帧。

图2-3 交通监控视频



图2-4 公司门口监控视频

1. **视频中运动物体检测**

目前检测视频中运动区域比较流行的方法有背景差分法、帧间差分法和光流法。每一种方法都有其各自的优点和局限性，本章节先对光流发进行简单介绍，然后通过实验对比背景差分法和帧差法之间的优劣。本章所述算法都是基于灰度图的，由RGB彩色图到灰度图的转换将在4.1节讲述。

1. **光流法简介**

光流是空间运动物体在观测成像面上的像素运动的瞬时速度。光流法检测运动物体的基本原理是：对平面图像中的每一个像素点赋予一个速度矢量（包含x，y方向的分量），这就形成了一个图像运动的速度矢量场，在运动的一个特定时刻，图像上的点与三维物体上的点一一对应，这种对应关系可由投影关系得到，根据各个像素点的速度矢量特征，可以对图像进行动态分析，检测图像中的运动区域。如果图像中没有运动物体，则光流矢量在整个图像区域是连续变化的。当图像中有运动物体时，目标和图像背景存在相对运动，运动物体所形成的速度矢量必然和邻域背景速度矢量不同，从而检测出运动物体及位置。

光流法大致可以分为三类：基于梯度的方法、基于匹配的方法和基于频域的方法。光流法的优点在于光流不仅有物体的运动信息，还有关于物体三维结构的丰富信息，但是光流算法大多数计算比较耗时，实时性和实用性都较差，较难应用于高清实时监控中，因此，本文不对光流法进行详细讨论分析。

1. **基于背景差分法的运动物体检测**

背景差分法简单的说就是用当前视频帧与背景图像对应的像素点做差从而得到运动物体。用背景差分法进行运动物体检测的性能取决于背景建模的技术，在取得背景图像的情况下，背景差分法速度快，运动物体检测准确，实现简单。但是现实中的监控视频由于环境光变化、场景情况复杂、摄像机抖动等因素影响，使得背景建模通常需要一定的时间，也使得背景建模变得比较困难。

背景建模算法也很多，目前比较常用的背景建模方法有均值法背景建模、中值法背景建模、高斯背景建模等。

1. **中值法背景建模**

中值法背景建模是最简单的背景建模之一，它是一种基于排序的背景建模方法，原理是对连续N帧图像相同位置的像素点灰度值进行从小到大排序，如果数组长度为奇数则此位置像素点的背景灰度值就是有序数组的中间值，如果数组长度为偶数则取数组中间的两个数的平均值作为背景灰度值。即：

式中代表对第k帧到第k+N帧建模后的背景图像，代表在位置为的像素点数组的中间值。此方法虽然简单，但是缺点也是显而易见的，它需要额外的内存来存储N帧图像的灰度值，并且N的取值也直接影响到背景建模的质量，更新背景模型也需要用到二分插入法，造成了额外的时间开销。

实验效果如下图所示：

1. **均值法背景建模**

均值法背景建模也是最简单的背景建模之一，它的原理跟中值法背景建模有点相似，只不过是将中值替换成均值。由经验很容易得出，在固定摄像头拍摄下，运动物体在画面中同一个地方停留的时间很短，例如在视频样本中抽取一帧如图xxx所示，车辆即是运动物体，我们追踪每一帧某一点的像素灰度值变化，可以得到图xxx

从图xxx可以看到，这一点的像素灰度值在通常情况下总是在一个很小的区间范围波动，只有出现运动物体的时候才会有较大的波动，均值法背景建模的原理就是将连续图像序列同一位置的像素灰度值累加后求平均值，并将此平均值作为背景图像的灰度值。即：

使用此方法的建模速度会比较快，因为累加求和可以用于每一帧，而且不需要额外的存储空间来保存图像序列，但是随着时间的增加，这种方法不能很好的适应复杂背景变化，需要重新从0开始建模。

实验效果如下图所示：

1. **高斯背景建模法**

而其中背景建模效果最好，最能适应背景变化的就是高斯背景建模方法。

单高斯背景模型的中心思想是，认为一个背景图像，其特定位置的像素亮度（灰度值）的分布满足高斯分布，即对背景图像B，坐标为(x,y)的点的灰度值I

满足：

其中u代表平均值，d代表方差，这是背景模型每个像素点的属性。背景建模过程也就是将图像中每一个像素点进行高斯分布建模，其样本就是视频连续图像中的相同位置像素点的灰度值。每读入一帧新的视频图像，就用像素点的高斯模型进行匹配，判断其是否属于运动像素点；同时，随着时间的推移，背景图像也会不断发生微小的变化，这时候就要不断更新每个像素点高斯模型的参数，以适应背景变化。

处理多模态场景一般使用混合高斯建模的方法。混合高斯背景模型是采用n（3到5）个高斯模型来对图像中每个像素点进行建模，这里实验采用文献[x]中所描述的混合高斯背景建模方法，检测效果如下图所示：

1. **基于帧差法的运动物体检测**

帧差法是通过对视频图像序列相邻两针或者隔几帧进行一次差分运算，从而获取运动物体区域。与背景差分法相比帧差法少了背景建模的过程，速度是最快的运动物体检测算法，而且帧差法基本不会受到环境光的影响，能够适应各种复杂的环境，稳定性高。但是帧差法也有显而易见的缺点，由于两帧相减，相同的区域差值变为0，因此通过这种方法获得的差图像往往是运动对象的轮廓，不能够提取完整的运动对象，也就是会形成“空洞”，此外差图像中会留下上一帧运动对象的“拖影”。目前也有一些帧差法的改进算法，接下来本文实现并对比了3种帧差算法的实际效果。

1. **传统帧差法**

用相邻两帧直接做差是最简单的方法，其算法步骤如下：

1. 采集视频图像的第k帧和第k-1帧
2. 将和进行灰度化和中值滤波操作，得到和
3. 计算差图像：
4. 对进行阈值化操作，将中灰度值大于阈值T的像素点标记为运动像素点，反之标记为背景，最后得到二值差分图像，如下所示：

实验结果如下图：

1. **三帧差分法**

三帧差分法主要为了解决差图像的“拖影”现象，更精确的得到运动对象的轮廓，与传统帧差法相比多了一步“与”的操作，其算法步骤如下：

1. 采集视频图像序列的连续三帧：第k-1帧，第k帧和第k+1帧
2. 将图像转化为灰度图后进行中值滤波操作得到，和，然后分别计算相邻帧之间的差图像：
3. 对和分别进行二值化操作，得到两个二值和图像和，将这两个图像按像素点进行“与”操作，即可得到第k帧图像的三帧差分结果，公式如下：

实验结果如下图：

1. **与运动历史图结合的三帧差分法**

三帧差分法确实能有效消除“拖影”现象，但是还是没能解决运动物体检测过程中的“空洞”问题。在实际运用过程中，“拖影”的影响只是会造成检测到的矩形轮廓偏大，这对后续的运动物体跟踪没的影响较小，但是“空洞”问题会使得一个运动对象分为两块或者更多，使得运动物体跟踪不准确，影响较大。为了解决“空洞”问题，这里采用了一种与运动历史图像相结合的三帧差分法进行运动区域检测算法。

运动历史图像（MHI）是一种运动分割方法。运动历史图像是一种保存运动物体运动痕迹的数据结构，可以将其视为一个大小与原图像相同的二维数组，在每帧的运动物体检测后对其进行更新，在有运动像素的对应数组坐标以时间的方式做一个标记，随着帧数的推移，不断对运动历史图进行更新，旧的运动像素点如果超过设置的时间区段时就会被清零。

在3.3.2节中得到视频第k帧的三帧差分图像后，采用如下公式对第k帧的运动历史图像进行更新：

其中*t*是运动像素保留时间，单位是帧。当一个像素点经过*t*帧都没有运动时，就清0。*t*的取值如果太大，运动区域检测结果就会造成很长的“拖影”，相反*t*的取值太小，就会形成较大的“空洞”，当*t*=1的时候就跟三帧差分法的结果一样了。

试验结果如下图：

1. **运动物体检测的实验分析**
2. **运动点团处理与跟踪**

第三章已经详细讨论了几种比较成熟的运动物体检测算法，并对三帧差分法进行了改进，在对各个算法做了实验对比后，本文选择了改进的三帧差分法进行运动物体检测。但是检测结果还没有经过其它处理，还比较“粗糙”，要经过进一步处理才能进行运动物体跟踪。运动物体检测与跟踪过程的主要算法流程如图4-1所示：

图像序列灰度化

运动物体检测

中值滤波去噪

形态学处理

运动点团跟踪

采用第三章3.3.3节所述改进的三帧差分法

图4-1

1. **图像灰度化**

从原始视频获取到的图像是彩色图像，如果直接对彩色图像操作，计算量会非常大，为了提高算法处理速度，要先将彩色图像转化为灰度图。

彩色RGB图像的每一个像素点有三个分量，分别代表红、绿、蓝三个通道的值。由RGB颜色空间的图像转化为灰度图最简单的做法是取R、G、B三个分量的平均值，但是这种做法并不科学，因为人眼对红、绿、蓝三种颜色的感知是不一样的，灰度值其实就是亮度，由RGB像素转化为灰度像素的公式为：

（4.1.1）

在一些处理器中，整数的运算比浮点数运算效率更高，因此将上述公式左右放大1000倍后可以变换为：

（4.1.2）

式中括号最后加的500是作四舍五入用。将公式中的浮点数转化为整型后，运算效率得到一定的提升，唯一耗时的部分在于最后的除法运算。而移位运算可以快速实现除以2的n次方，因此可以考虑将公式中的除法运算替换成最快的位移运算。采用16位精度来计算R、G、B前面的系数：

采用去尾法得到一个整数的系数，最后得到的灰度转换公式为：

（4.1.3）

实验仿真使用一交通道路视频进行不同公式的算法效率测试，视频长度为3623帧，分辨率为1280\*720，图4-2（原始）是视频中的一帧，这里的时间只统计图像由彩色图转化为灰度图的时间。结果如表4-1所示：

表4-1 灰度转化实验数据表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 公式 | 视频长度/帧 | 视频分辨率 | 总时间/ms | 平均时间/ms | 结果图 |
| 4.1.1 | 3623 | 1280\*720 | 15722.8 | 4.34 | 图4-2(a) |
| 4.1.2 | 8406.68 | 2.32 | 图4-2(b) |
| 4.1.3 | 5764.69 | 1.59 | 图4-2(c) |
| Opencv | 7646.59 | 2.11 | 图4-2(d) |



原始

(a)

(b)

(c)

(d)

图4-2 各种算法灰度图

从表4-1和图4-2中可以看出，使用不同的公式算法得到的灰度图是一样的，不同在于时间效率，这里除了上文提到的的三个公式（4.1.1，4.1.2和4.1.3）之外，还与opencv中的灰度转换函数做了对比，可以看出最快的是采用公式4.1.3进行灰度图转化。

1. **中值滤波去噪**
2. **中值滤波基本原理**

在图像的生成、传输或者变换过程中，由于受到多种因素的影响，如视频录制设备失真，传输噪声，曝光条件不佳等原因，导致输出的图像会有噪声。而在监控视频中最明显的就是椒盐噪声，也就是画面上会出现黑白杂点，这可以在图xxx中明显的体现出来。为了能够提取更准确的运动物体区域，需要对图像进行去噪处理，虽然可以采用简单的线性滤波进行处理，但是多数线性滤波具有低通特性，在去除噪声的同时也使图像的细节和边缘变得模糊，所以这里我选择非线性的中值滤波进行去噪处理。

中值滤波是一种非线性平滑技术，它是基于排序统计理论的一种噪声去除技术，中值滤波能够在某些条件下既可以去除噪声而又可以保护图像的边缘和细节。中值滤波的基本原理是把数字图像中的一点的值用该点领域中各点值的中值代替，让周围的像素值接近真实值从而达到消除孤立噪声点的目的。中值滤波的实现方法为用一个滑动窗口沿着图像滑动，窗口中心像素点的灰度值用窗口内所有像素点的中值来代替：

其中是输出图像，是原图像，A是二维活动窗口，大小可以为3\*3、5\*5或7\*7等。假设一个图像的宽高分别为w和h，窗口A大小为n\*n，使用快排进行排序，则中值滤波的时间复杂度为：

1. **改进的中值滤波方法**

这里可以针对整个运动物体检测与跟踪的算法流程对中值滤波算法进行改进。由第三章的运动物体检测算法可以看出，改进的三帧差分法最后输出的结果是一个二值化的灰度图，也就是图像里面的像素值只有0和255两种，因此在中值滤波算法中可以对排序步骤进行改进，我们可以用计数的方法来替换掉排序算法，也就是统计窗口A中值为0的像素点个数和值为255的像素点个数就可以直接得到中值，如果0的像素个数大于255的像素个数，则中值为0，否则中值为255。这时，改进后的中值滤波时间复杂度为：

实验结果如下：

1. **形态学处理**

最基本的形态学操作是腐蚀和膨胀，进一步的还有开运算和闭运算。图像形态学变换一般可以用来消去噪声、分割独立的图像元素以及连接相邻的图像元素等。在4.2节已经使用改进后的中值滤波方法进行噪声消除，但是对于一个运动点团还是有可能产生一些“缝隙”而导致检测成为2个运动点团，所以这里我采用形态学处理将这些“缝隙”连接起来。

1. **腐蚀和膨胀**

腐蚀[1]和膨胀[1]操作其实类似于数学卷积操作。腐蚀是指将一个图像（或者是图像中的一部分，称之为A）与核（称之为B）进行卷积。这里的核可以是任意大小或形状，一般为带有一个参考点的正方形或者圆形，在这里我取核为3\*3的矩阵。腐蚀操作可以表示为：

*B*

这公式的意思是，核B与图像A卷积，即计算核B覆盖区域以内的最小值，并把这个最小值赋值给核B参考点指定的像素，腐蚀其实就是求局部最小值的操作，这样就可以消减图像一些连通区域的边缘。如图xxx所示就是腐蚀操作后的结果。

膨胀操作刚好和腐蚀操作相反，是一个求局部最大值的操作。可以表示为：

*B*

其含义就是计算核B覆盖区域以内的最大值，并把这个最大值赋值给核B参考点指定的像素，这样就可以使图像中高亮区域逐渐增长，达到填充图像微小空洞和缝隙的效果。如图xxx所示就是膨胀操作后的结果。

1. **开运算与闭运算**

开运算[1]是将图像先腐蚀后膨胀，开运算可以去除微小噪声，平滑物体边界或者分割物体等作用。例如开运算可以用来统计二值图中的区域数量，假设得到一个在显微镜上观察到的细胞图片，先将其进行二值化处理然后再进行开运算，可以将每个细胞清楚的分割开来，便于细胞数量的统计。

闭运算[1]刚好跟开运算相反，是将图像先膨胀后腐蚀。闭运算可以用来连接两个相邻近的连通区域，通常使用在连通区域分析中。

开运算和闭运算都是不改变物体大小的，开运算最显著特征就是消除孤立点，闭运算最显著的特征就是填充微小区域或者缝隙。而在4.2节已经讨论过采用改进的中值滤波来去除微小噪声，并且通过实验也发现对二值运动点团做闭运算的效果是最好的，能够填充一个运动对象中的空隙，使之在连通区域检测的时候不会被分割成两个物体。因此，我使用闭运算来做二值图像的缝隙填充处理，图xxx就是开运算后的结果。

1. **连通区域检测**

到目前为止得到的图像是经过多重处理的二值图，但是对于计算机来说还没办法区分图像中每一个运动物体，因此我们要对其进行连通区域检测[11,12]，使每个物体能够用计算机语言分别标记开来。

本文采用线段编码算法来寻找连通区域。线段编码是一种通过逐行扫描，从而快速提取连通区域位置和大小的算法。这里我采用实例来说明此算法的过程。如图xxx所示是一个二值图像中的连通区域，将其每一行中连续的像素点看作一条线段，分别打上数字标签方便说明。首先扫描第一行，得到线段1和线段2，分别加入连通区域A和区域B中；然后继续扫描第二行，得到线段3和线段4，线段3和线段1是相接的，因此把线段3也加入区域A中，同理线段4加入区域B中；第三行的线段5加入区域A，线段6加入区域B；扫描到第四行的时候，线段7跟线段5和线段6都相接，因此就将区域A和区域B合并，都标记为区域A；最后将线段8和线段9也加入区域A中。

1

2

3

4

5

6

7

8

9

图xxx 线段编码算法演示图

根据这个例子，可以总结得到线段编码算法步骤描述如下：

1. 逐行对图像进行扫描，得到该行中的线段；
2. 如果当前行的线段跟上一行的一条线段相接，则将当前行的线段加入上一行线段所在的连通区域；如果当前行线段跟上一行有多条线段相接，则合并上一行相接线段所在的连通区域；否则将当前行线段标记为一个新的连通区域；

这种线段编码的算法简单高效，只需要扫描一次整个图像的像素点即可得到连通区域，假设图像宽和高分别为w和h，则算法时间复杂度是O(w\*h)。

1. **运动物体跟踪**

在获取了运动点团的连通区域大小和位置之后，就可以利用这些信息来进行运动物体跟踪了。进行运动物体跟踪是为了记录一个运动事件的发生时间、结束时间以及运动轨迹，将其完整的运动过程记录下来。由于应用场景特性和实时性的要求，这里我采用基于连通区域矩形框跟踪的方法，下面先对这种方法进行原理介绍，分析其不足，然后再对其进行改进。

1. **基于矩形框的运动跟踪**

4.4小节中得到了每个运动物体所对应的连通区域，用一个最小的矩形框将这个连通区域框起来，得到一个代表运动物体的矩形框，我们观察图xxx同一个运动物体的矩形框在连续图像序列中出现的位置，很容易发现在相邻帧之间同一物体的矩形框位置只是稍微移动了一下位置，相邻两帧之间的矩形框有大部分面积是重叠的。这是因为在录制监控视频的时候，只要视频采样率（帧率）够大，那么运动物体相邻两帧之间的位置变化就很小，通过实际统计验证，只要监控视频采样率在10帧/秒以上，相邻两帧的运动物体矩形框重叠面积大于它们各自面积的60%以上。根据这种运动特征，我们可以得到一个很简单的运动物体跟踪算法：

1. 对每一帧进行运动物体的最小外接矩形框检测；
2. 对当前帧检测到的每一个运动物体矩形框，分别与上一帧的矩形框链表进行匹配，如果在上一帧能找到重叠面积大于60%的矩形框，则判定为同一个物体并，如果没能找到匹配的矩形框，则将这个矩形框标记为一个新的运动时间的开始；
3. 对每一个正在跟踪的运动事件（一个矩形框序列），如果当前帧没有矩形框能与此运动事件的最后一个矩形框匹配，则标记此运动事件已经结束。

这个算法很容易理解，用于运动物体跟踪的时间复杂度也很低，不需要进行大量复杂的计算，较容易满足实时监控的需求。在第2.2.1小节所描述的应用场景中，多数情况下运动物体相对独立，没这么复杂，都能能用这种跟踪算法很好的满足运动物体跟踪的需求，但是场景总是多变的，比如当同一时间有两个人进入场景，他们面对面前进，当两人走到一起的时候就会产生遮挡，这时候算法检测出这两个人是一个整体，原来的运动跟踪算法就会将原来的两个运动事件跟丢，产生了错误的的跟踪情况。这种错误的情况有可能是一个人并入了另一个人的运动事件中，也有可能是生成一个全新的运动事件。

1. **改进的运动物体跟踪方法**

上一小节中的跟踪仅仅是基于运动物体最小矩形框序列进行跟踪，所能应用的场景十分有限，为了使算法能在多个运动物体相互干扰的情况下也能很好的进行跟踪，这里在跟踪过程中加入了运动方向。

首先列举基于矩形框跟踪可能出现的情况：正常运动，运动点团分裂，运动点团合并，运动点团先合并后分裂。下面结合图xxx详细说明这几种情况：

1. 正常运动：图xxx（a）中的情况是没有干扰的正常运动情况，虚线矩形框A表示前一帧运动物体的最小外接矩形框，实线矩形框B表示当前帧运动物体最小外接矩形框，这种运动情况没有运动点团的分裂或者合并，属于最理想的情况，使用上一小节所提的算法就可以很好的进行跟踪。属于这种情况的判定条件为矩形框A和矩形框B的重叠面积均大于矩形框A和矩形框B的面积的60%：
2. 其中是A和B的重叠面积，和分别代表矩形框A的面积和矩形框B的面积，类似的符号也应用在下文的描述中。

运动点团分裂：图xxx（b）中的是运动点团分裂的情况，运动点团由虚线矩形框A经过运动后分裂成两个实线矩形框B和C，这种情况的跟踪由3个部分组成，第一个部分记录合并情况下的矩形框A代表点团运动过程，第二部分和第三部分分别记录矩形框B和矩形框C所代表的点团运动过程，可以用一条分叉的链表来记录这样一个过程。这种运动点团分裂情况的判定条件为A和B的重叠面积大于B面积的60%但是小于A面积的60%，A和C的重叠面积大于C面积的60%但是小于A的60%：

1. 运动点团合并：图xxx（c）中的是运动点团合并的情况，两个运动点团的矩形框A和B经过运动后重叠在一起，形成一个新的运动点团，情况刚好和运动点团分裂相反，其跟踪过程也由3部分构成，第一部分和第二部分分别是矩形框A和矩形框B的运动过程，第三部分是矩形框C的运动过程，此种情况的判定条件为：
2. 运动点团先合并后分裂：这种情况其实是运动点团合并和运动点团分裂的组合，矩形框A和B进行相向运动，在矩形框C处合并，然后继续运动最终分裂，这种运动在行人监控中经常出现，在跟踪过程中加上运动方向就是针对这个情况的。这个运动过程保存就相对复杂一些，首先保存A和B各自的运动过程，然后保存重叠时的矩形C运动过程，在分裂的时候再根据其合并前速度矢量识别A和B进行运动跟踪和保存。合并过程和分裂过程的判定条件参照第c)和第b)种情况。

(a)

(b)

(c)

(c)

图xxx 运动跟踪情况示意图

A

B

A

B

C

A

B

C

A

B

C

只要不是在人山人海的步行街和交通堵塞的情况下，上述几种情况就已经能很好满足运动物体跟踪的要求，跟踪过程中以合并前或者分裂后的子矩形框为最小运动单位进行运动事件记录，记录运动事件的数据结构将在后文讲述。这种算法的步骤为：

1. 先用情况a)对每一帧检测到的运动物体矩形框进行跟踪，如果一个运动事件持续了0.5秒以上则称之为有效运动事件并进一步跟踪，如果持续时间小于0.5秒则删除；
2. 跟踪过程中如果出现了b)，c)或d)这3种情况之一，就按照上述情况分析中所说的去处理跟踪过程；
3. 当一个矩形框在视频边界处且像素面积大小小于100像素时或者消失则判定此运动物体已经不在视野范围内，结束此运动事件跟踪。

此算法在效率上可以说并没有增加太大的时间开销，因为这是在逻辑层面对运动事件可能出现的情况作出不同的跟踪处理，没有涉及到图像处理的东西。

1. **运动事件记录**
2. **EvenNode数据结构**

在跟踪过程中，由于跟踪的需要，最后得出来的跟踪链表可能是交叉型的，也可能是没有分叉的，但是跟踪完毕后实际记录运动事件是以分裂后或者合并前的矩形框进行一个事件保存的，一个运动事件用一个EventNode实例来保存，EventNode类的数据结构如表xxx所示，其中Rectangle类是保存一个矩形框位置和大小的数据结构，如表xxx所示。

表5-7 EventNode类数据结构

|  |  |  |
| --- | --- | --- |
| 变量类型 | 变量名 | 变量的含义 |
| int | startFrame | 运动事件的开始帧。 |
| int | endFrame | 运动事件的结束帧。 |
| List<Rectangle> | trackList | 一个Rectangle类型的List列表，保存了整个运动事件每一帧的矩形框以及矩形轮廓左上角的坐标。 |

表5-8 Rectangle类数据结构

|  |  |  |
| --- | --- | --- |
| 变量类型 | 变量名 | 变量的含义 |
| int | X | 矩形框左上角的x坐标。 |
| int | Y | 矩形框左上角的y坐标。 |
| int | Width | 矩形框的宽。 |
| int | Height | 矩形框的高。 |

1. **本地分析文件数据结构**

考虑到用户配置方便，本系统没有采用数据库进行数据存储，而是采用本地文件的形式保存分析完的视频信息，分析一个视频后会产生2个本地文件，一个是保存了所有运动事件的文本文件，一个是视频摘要文件（这个文件的生成在下一节讲述）。将运动事件信息保存到本地能有效避免二次分析，当用户要查看监控视频的运动事件或者进行视频检索时可以节省大量时间。

表5-9 本地视频分析文件结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 文件头 | 事件数 | | 处理时的帧间隔 | | 视频的帧率 | |
| 事件1 | 开始帧 | 结束帧 | | 事件1节点数 | | 事件1节点列表 |
| 事件2 | 开始帧 | 结束帧 | | 事件2节点数 | | 事件2节点列表 |
| .  .  .  . | .  .  .  . | .  .  .  . | | .  .  .  . | | .  .  .  . |
| 事件n | 开始帧 | 结束帧 | | 事件n节点数 | | 事件n节点列表 |

表5-10 事件的节点结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 节点1 | 矩形框x坐标 | 矩形框y坐标 | 矩形框宽 | 矩形框高 |
| 节点2 | 矩形框x坐标 | 矩形框y坐标 | 矩形框宽 | 矩形框高 |
| .  .  . | .  .  . | .  .  . | .  .  . | .  .  . |
| 节点n | 矩形框x坐标 | 矩形框y坐标 | 矩形框宽 | 矩形框高 |

如表5-9所示，一个本地视频分析文件包括2个大的方面：一个是文件头，包含了视频的运动事件数、视频处理时的帧间隔和视频的帧率；另一个是运动事件的列表。在运动事件列表中，每一个事件包含的数据有：事件开始帧，事件结束帧，事件总节点数，事件节点列表。

表5-10为本软件保存某一个运动事件信息的事件节点文件格式。一个事件的节点数据包括运动物体矩形框的左上角坐标（x坐标和y坐标），和此运动物体矩形框的宽和高。

1. **视频摘要生成**

视频摘要是指将监控视频中检测到不同时间段的运动事件都“拼接”到同一个时间播放，让用户在查看监控视频的时候可以快速浏览所有运动事件，找出自己感兴趣的事件并回溯。当然如果一个视频运动事件数量太多，如果都拼接到一个时间段去播放会看到很多“混乱”的运动物体，系统设置为可以让用户自定义同一时间最大运动事件数，这样生成的视频摘要文件长度就由视频总运动事件数和用户自定义的同一时间最大运动事件数来决定。视频摘要生成的主要步骤如下：

1. 在运动物体检测与跟踪的时候，截取一帧没有运动对象的图像作为背景图，新建一个视频写入对象；
2. 假设用户自定义同一时间最大运动数量为n，提取这n个运动事件的第一帧矩形框的图像，共同拼接到视频写入对象的第一帧，对接下来的每一帧重复这个操作，直到这n个运动事件都结束；
3. 重复步骤2)，直到整个视频的所有运动事件都拼接结束完毕。

需要说明的是，在拼接过程中不可避免的会出现运动物体遮挡现象，这里会对其做半透明处理，另外对每个运动事件都加上编号，以便识别。图xxx是对一个行人监控视频生成视频摘要的一帧。

1. **视频内容检索**

当用户不知道运动物体的具体特征时，可以通过先看一遍视频摘要，粗略浏览一遍所有运动事件，再根据感兴趣的运动事件编号跳转到此运动事件发生的时间，从源视频中查看整个运动过程。如果用户知道自己想要寻找什么样的对象，比如说从10点35分到10点50分期间经过此路段的黑色轿车，那就可以通过图像特征检索来实现了。本章描述了5种特征检索方式，根据第二章中描述，从时间复杂度最低的特征开始逐级筛选运动事件，这5种特征由复杂度从低到高排序为时间特征，入侵区域，运动方向，对象类型（人或车），颜色特征。

1. **根据发生时间检索**

根据运动事件发生的时间进行筛选是最简单的，但是前提要知道该视频开始拍摄的时间，这样就可以通过运动事件发生的帧号和视频的帧率来算出运动事件实际发生的时间，然后再判断这个时间有没有落入用户指定的时间区间即可。用步骤描述如下：

1. 读取源视频文件名，提取该视频开始拍摄的时间，记为t；
2. 读取源视频帧率f，根据运动事件的开始帧号s算出此事件相对于视频来说发生的时间为，该事件实际发生的时间为；
3. 获取用户输入的时间区间为，如果落入这个区间则符合筛选条件，否则过滤掉。

对于一个运动事件来说，这个筛选步骤的时间复杂度为O(1)。

1. **根据入侵区域检索**

根据入侵区域检是指用户指定一个矩形区域，然后检测视频中有哪些运动事件“入侵”了这块区域。因为每一个运动事件都保存了其每一帧的运动点团矩形框，所以遍历一个运动事件的矩形框序列，与用户所指定的矩形区域做匹配，即可得到此运动物体是否有入侵行为，具体步骤如下：

1. 用户在原视频背景界面上画一个矩形框，作为入侵区域的标识；
2. 按顺序读取一个运动事件的下一个矩形框，并将矩形框逐一与用户定义的入侵区域进行匹配，如果重叠面积大于入侵区域的20%（可以自定义），则直接返回此运动事件有入侵行为，否则进入第三步；
3. 重复步骤2)，直到此运动事件的矩形框序列被遍历完，如果都没有矩形框与入侵区域匹配，则返回此运动事件无入侵行为。

对于一个运动事件来说，假设矩形框序列长度为n，则该筛选步骤的时间复杂度为O(n)。

1. **根据运动方向检索**

根据运动方向检索是指用户事先定义一个检测方向，然后查找视频中按照检测方向运动的事件。我们已经有了每一个运动事件的每一帧运动点团矩形框，可以将矩形框中心作为其质心，这样就可以得到运动轨迹。但是由于运动物体并不都总是沿着一个方向运动，有可能运动一段距离后突然转弯或者折返，因此不能简单的将质心点拟合成一条直线，本文采用的方法是逐个检测相邻质心点的运动方向，如果连续1秒内的质心点运动方向都跟检测方向一致，则将此事件返回，否则过滤掉。算法步骤为：

1. 用户在源视频背景界面上画一根带箭头的直线，作为检测方向；
2. 读取一个运动事件，将运动事件的矩形框中心作为运动对象的质心，得到运动轨迹；
3. 依次检测相邻两个质心点的运动方向是否与检测方向匹配（角度误差为10度），如果连续5次匹配成功，则筛选出此运动事件，如果遍历完所有质心点都没有连续5次匹配成功，则过滤掉此事件。

对于一个运动事件来说，如果矩形框序列长度为n，则根据运动方向检索的时间复杂度为O(n)。

1. **基于Haar分类器的对象类型检索**

由于本文的监控视频主要是车辆视频和行人视频，所以检测对象分为行人、车辆和其它物体3种。这里我采用基于Haar特征的级联分类器[13,15]来进行运动物体的分类，关于此分类器的实现原理不属于本文的研究范围。Haar分类器最成功的就是应用在人脸检测中，但是并不仅限于人脸检测，它适用于其它外表有区别的（接近刚性的）物体检测，例如人脸，鼻子，眼睛，行人，车的正面，车的后部等都可以用Haar分类器进行检测。Opencv已经实现并提供了Haar分类器的训练接口和检测接口，本文所做的工作主要是选取合适的样本为行人和车辆分别训练出可靠的分类器，运动物体对象类型识别的步骤如下：

1. 从监控视频中截取300个行人的全身图片作为训练样本，样本形式如图5-1所示，通过训练后得到一个行人分类器；
2. 从监控视频中截取300个车辆正面和背面的图片作为训练样本，样本形式如图5-2所示，通过训练后得到一个车辆分类器；
3. 读取运动事件的矩形框序列，选取中间的5个矩形框并定位到相应的源视频图像中，根据矩形框位置大小截取相应的运动物体彩色图；
4. 将上一步中截取的5个彩色图像放到行人分类器中进行识别，如果有4张或者4张以上的图像被判定为行人，则返回此运动对象为行人，否则进行下一步；
5. 将5个彩色图像放到车辆分类器中进行识别，如果有4张或者4张以上的图像被判定为车辆，则返回此运动对象为车辆，否则进行下一步；
6. 如果该运动对象即不是行人也不是车辆，则判定为其它对象。

上述步骤3)和步骤4)中如果将一个运动事件所有帧都识别的话会耗费很多时间，只选取5帧图像作识别；选取中间的图像是因为运动开始或者运动结束附近的运动对象有可能已经有部分在整个图像的边缘外面，运动时间轴中间的图像可以得到一个较完整的运动对象。

1. **基于HSV颜色直方图的颜色特征检索**

颜色直方图被广泛应用于许多图像检索系统中的颜色特征提取。它描述了不同的颜色在整个图像中所占的比例，但是没有描述每种色彩所处的空间位置。Swain和Ballard最先提出了应用颜色直方图进行图像特征提取的方法[17]，颜色直方图也可以是基于不同颜色空间和坐标系的，常用的颜色空间有RGB空间，HSV空间，Luv空间和Lab空间等。这里我选择了更接近于人们对颜色主观认识的HSV颜色空间，颜色直方图的计算要先将颜色空间量化为72维（H分量12维，S分量3维，V分量2维），然后通过计算颜色落在每个小区间的数量即可得到颜色直方图，最后通过对这72维直方图进行排序得到图像的主颜色。具体的步骤如下：

1. 用户选取一个检索颜色；
2. 读取一个运动事件的矩形框序列，选取中间的5个矩形框并定位到相应的源视频图像中，根据矩形框位置大小截取相应的运动物体彩色图；
3. 分别将5个彩色图由RGB颜色空间转换到HSV颜色空间；
4. 计算5个彩色图的颜色直方图，并通过排序选出每个直方图频率排前3的颜色作为主颜色；
5. 如果有4个直方图主颜色中都颜色能 能跟检索颜色匹配在同一个直方图维度，就筛选出此运动物体，反之就过滤掉此物体。
6. **系统实现与运行测试**
7. **系统主要功能**

基于摘要的监控视频内容检索系统旨在开发一个软件系统，该系统能对监控视频进行实时或离线的摘要分析，然后根据特征进行视频信息检索。系统主要分为两大模块：视频摘要和视频信息检索。表6-1描述了系统的主要功能：

表6-1 系统功能列表

|  |  |  |
| --- | --- | --- |
| **模块名称** | **子功能** | **功能描述** |
| 本地/实时视频摘要 | 本地视频文件读取 | 支持主流视频格式，包括avi，mp4，wmv，flv，rmvb等 |
| 摄像头视频流读取 | 支持usb摄像头和rtsp视频流读取 |
| 事件跟踪 | 对视频中的运动对象进行检测与跟踪 |
| 事件统计 | 实时统计监控视频中出现的事件数量 |
| 生成视频摘要 | 将不同时间发生的事件整合到同时间的场景中 |
| 生成分析文件 | 生成分析文件，记录监控视频中所有事件的信息 |
| 摘要事件查看 | 视频摘要查看，可以根据事件编号回溯原始视频 |
| 视频信息检索 | 根据事件发生时间检索 | 用户指定一个时间区间，检索该时间区间的事件 |
| 根据事件的运动方向检索 | 根据事件中目标的运动方向进行检索 |
| 根据事件的入侵区域检索 | 根据事件中目标的入侵区域进行检索 |
| 根据事件颜色特征检索 | 根据事件中目标的主要颜色进行检索 |
| 根据事件的目标类型检索 | 根据目标的类别（分为人、车、物体）进行检索 |

1. **开发环境**
2. **软件环境**

操作系统：windows7

IDE：Qt Creator4.8

开源库支持：opencv2.4.0和ffmpeg2.0

1. **硬件环境**

cpu：intel® Core™ i3-3220（双核3.30GHz）

显卡：i3核心显卡

内存：4G

硬盘：320G

摄像头：usb摄像头若干（分辨率：1280\*720，640\*480）

1. **系统用户界面与操作**

因为此项目属于应用研究型项目，主要是做算法相关的研究与实现，所以没有太复杂漂亮的用户界面。系统支持本地视频的分析和实时视频流的分析，同时支持网络rtsp监控视频流的处理（需要联网）。为了部署方便，本系统没有使用数据库进行数据存储，视频分析结果是存储在本地文件中的，同时本系统无需安装，直接运行可执行文件即可。

本系统存储视频运动事件的文件和生成的视频摘要文件统一放在名为“analyze”的文件夹中，该文件夹与源视频在同一个目录下。比如源视频的文件路径是“D:\video\test.avi”，则“analyze”文件夹的路径是“D:\video\analyze”，这样可以方便查找。

1. **实时视频摘要分析**

实时视频摘要是在实施监控中进行运动物体检测与跟踪，并在监控结束（生成一段监控视频的时候）后形成视频摘要。实时视频摘要主界面如图6-1所示。



图6-1 实时视频摘要主界面

视频显示

开始监控视频显示

参数设置视频显示

摄像头选择视频显示

事件数量视频显示

由于硬件条件限制，这里只实现了2路摄像头的实时监控，当然如果在处理核心更多的服务器上还可以实现多路摄像头的实时监控。实时视频摘要监控的操作步骤如下：

1. 用户先选择需要监控的摄像头，预览视频并调整摄像头角度；
2. 然后点击“参数设置”按钮进行参数设置，这里可以设置的参数有视频缩放百分比、分析跳帧、保存的文件路径和运动事件的面积限制；
3. 点击“开始监控”按钮进行监控并实时进行分析，此时“开始监控”按钮会变换一个图标，变成“结束监控”按钮；
4. 结束监控时点击“结束监控”按钮，系统自动生成视频摘要。
5. **本地视频摘要分析**

在实时视频摘要界面点击左下角的“本地模式”可以进入本地视频摘要界面，如图6-2所示。本地视频摘要是对本地视频文件进行运动物体检测与跟踪然后生成视频摘要，其核心算法是与实时视频摘要一样的，另外本系统支持批量本地视频文件处理功能。



图6-2 本地视频摘要主界面

本地视频摘要分析的步骤如下：

1. 点击“打开文件”按钮，在本地硬盘中选择一个或多个视频文件，如果选择多个则会进行批量视频处理，选择完视频后在“打开文件”按钮上方的文本框会显示视频文件的路径；
2. 点击“参数设置”按钮，设置一些视频分析相关的参数，这里可以设置的参数有视频缩放百分比、分析跳帧和运动事件的面积限制；
3. 确认选择的视频文件以及参数设置无误后，点击“分析视频”按钮，如果该视频已经分析过并记录在本地文件中，则会提示用户读入本地分析文件还是重新分析，如果该视频没有分析过或者用户选择重新分析则开始进行本地视频分析。分析过程中会有进度条显示进度，运动事件数量显示以及耗时显示。
4. 视频分析完毕后会在节目右边的事件列表中显示，每一个事件都有其各自的编号，开始时间和结束时间。
5. **视频摘要播放和事件回溯**

视频摘要播放是直接播放analyze文件夹下生成的所有运动事件视频文件，在该视频中所有运动事件都会打上标号，用户可以根据标号在主界面右边的运动事件列表中回溯到源视频观看原始运动过程如图6-3是播放视频摘要的截图，图6-4是回溯运动事件的截图。具体步骤如下：

1. 在分析完一个视频或者读入一个已经分析完的视频后，点击“播放视频摘要”按钮，系统会跳出一个新的播放界面进行播放，重叠的运动对象会做半透明处理，在每个对象左上角有一个编号用作识别；
2. 用户在视频摘要中看到感兴趣的运动对象，可以根据其左上角的编号，在事件列表中选择相应的事件点“播放”进行回溯，这里的播放界面跟视频摘要播放界面相同，用户可以随意进行拖动进度条，暂停和循环播放等操作。



图6-3 播放视频摘要

图6-4 回溯运动事件

1. **视频内容检索**

视频内容检索的步骤如下：

1. 点击主界面中的“视频检索”按钮，进入视频检索界面。如图6-5所示，该界面可以定义5种检索特征：在视频背景上画带箭头的直线代表运动方向；在视频背景上画矩形框代表入侵区域；时间控件可以定义事件发生的时间区间；调色板可以定义检索颜色；类别选择按钮组合框可以选择检索的目标类型；
2. 用户选择一个或者多个检索特征；
3. 点击确定按钮开始检索，检索结果会显示在主界面右边的事件列表中，点击“播放”按钮就可以回溯到原始视频中观看运动过程。
4. **系统测试**

基于摘要的监控视频内容检索系统测试分为两个部分：运行速度测试与准确率测试。

1. **运行速度测试**

运行速度测试是指测试在对视频进行运动物体检测与跟踪过程中的处理速度，该测试主要是验证此系统所使用的算法能否满足实时视频分析的要求。测试环境与6.2节所述的开发环境相同，测试资源采用本地视频和实时摄像头视频流，对不同分辨率和帧率的视频进行分析测试，得到系统的处理速度。

1. **准确率测试**

准确率是指视频摘要所得到结果的准确度。视频摘要准确率是指运动事件的检出率，也就是在运动物体检测与跟踪过程中漏掉的运动事件越少越好，检出率的具体计算方法参考系统指标中所述。

1. **总结与展望**

通过前面的算法设计和实验结果分析，最后本文得到如下结论：

第一，本文实现几种常用的了基于背景差分法的运动物体检测和基于帧差法的运动物体检测方法，为了满足实时性的要求，都是先把图像从RGB彩色图转化为灰度图然后再进行处理。在通过实际实验对比后，基于高斯分布的背景差分法能够最准确的提取运动物体点团，但是对环境适应力不够强，当出现环境突变的情况需要一段时间进行背景重建，并且计算量大；基于改进后的三帧差分法速度快，提取的运动点团虽然没有背景差分法精确，但是能基本满足跟踪的需求。

第二，本文对一些图像处理算法做了适当的优化。在灰度图转化步骤，将浮点数乘法转化为整数乘法与位移的运算，在不影响结果的情况下一定程度的缩短了处理时间。在中值滤波步骤，针对二值图的特性对算法作出优化，也在不影响结果的情况下缩短了处理时间。然后再使用形态学闭操作连接相近的连通区域，使一个运动物体不至于分开。最后通过线段编码算法得到所有连通区域。

第三，运动物体跟踪采用了改进后的矩形框跟踪 ，对运动物体相对较少的监控场景能够得到很好的效果，而且视频摘要本来就适用于不复杂的运动场景。

第四，在视频摘要的基础上，本文实现了通过5种特征进行运动对象检索，分别是：根据发生时间的检索，根据入侵区域检索，根据运动方向检索，根据运动对象的类型检索和根据运动对象的主要颜色检索。

第五，最后本文采用上述所述算法实现了一个基于视频摘要的视频内容检索系统，实现了视频摘要和基于基于摘要的检索等功能，可以用于实时视频分析和本地视频分析。整个系统有很好的健壮性和实用性。

系统也有一些可以改进的地方，对于视频摘要部分，核心就是运动物体跟踪与检测，虽然能够保证运动事件的检出率，但是在复杂的运动场景效果不是很理想，可以在这方面加以研究和改进，使算法适用于更多的监控场景。在视频内容检索模块，目前只是实现了5种简单的特征检索，还可以再继续扩充检索特征。本系统是基于单台PC终端开发的，现在的云技术越来越发达，可以将其在云端实现，将视频摘要和视频内容检索应用到城市的高清监控中。

# 参考文献

[1] Gary Bradski, Adrian Kaebler. 学习OpenCV（影印版）[M]. 东南大学出版社, 2009.

[2] 于仕琪, 刘瑞祯. OpenCV教程：基础篇（附光盘）[M]. 北京航空航天大学出版社, 2007.

[3] XuHua Fang, Jianhong Fang. Human Motion Tracking Based on Adaptive Template Matching and GM(1,1)[J]. Intelligent Systems and Applications, 2009, Page(s):1-4.

[4], [Guiran Chang](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=p_Authors:.QT.Guiran,%20Chang.QT.&searchWithin=p_Author_Ids:37276340000&newsearch=true), [Wei Cheng](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=p_Authors:.QT.Wei,%20Cheng.QT.&searchWithin=p_Author_Ids:37931500300&newsearch=true), [Huiyan Jiang. Background Extraction and Update Method Based on Histogram in YCbCr Color Space[C].](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=p_Authors:.QT.Huiyan,%20Jiang.QT.&searchWithin=p_Author_Ids:38015119700&newsearch=true) [2011 International Conference on](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5779818) E -Business and E -Government (ICEE), 2011, Page(s):1-4.

[5] Yang Shu-Ying, Zhang Cheng, Zhang We-Yu, He Pi-Lian. Unkonwn Moving Target Detecting and Tracking Base on Computer Vision[C]. Fourth InterNational Conference on Image and Graphics, 2007, Page(s):490-495.

[6] Kesrarat D., Patanavijit V.. Experimental study efficiency of robust models of Lucas-Kanade optical flow algorithms in the present of Non-Gaussian Noise[C]. 2012 4th International Conference on Knowledege and Smart Technology (KST), 2012, Page(s): 43-48.

[7] Bouttefroy P.L.M., Bouzerdoum A., Phung S.L., Beghdadi A.. On the analyses of background subtraction techniques using Gaussian Mixture Models[C]. 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), 2010, Page(s):4042-4045.

[8] Omer O.A.. Region-based Horn-Schunck optical flow estimation[C]. 2012 Japan-Egypt Conference on Electronics, Communications and Computers (JEC-ECC), 2012, Page(s):73-78.

[9] Benedicte Bascle, Rachid Deriche. Region tracking through image sequences[C]. Fifth International Conference on Computer Vision, 1995, Page(s):302-307.

[10] 甘明刚, 陈杰, 刘劲, 王亚楠. 一种基于三帧差分法和边缘信息的运动目标检测方法[J]. 电子与信息学报第32卷第4期, 2010年4月, Page(s):394-397.

[11] Yang Yang, David Zhang. A novel line scan clustering algorithm for identifying connected componets in digital images[J]. Elscvier Science: Image and Vision Computing, 2003, Page(s): 459-472.

[12] Luigi Di Stefano, Andrea Bulgarelli. A Simple and Efficient Connected Components Labeling Algorithm[C]. International Confference on Image Analysis and Processing, 1999, Page(s):322-327.

[13] P. Viola and M. J. Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features[C]. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, Page(s):I-511-I-518 vol. 1.

[14] Michael Boyle. The Effects of Capture Coditions on the CAMSHIFT Face Tracker[R]. University of Calgary, Alberta, Canada, 2001.

[15] R. Lienhart and J. Maydt. An Extended Set of Haar-like Features for Rapid Object Detection[C]. 2002 international Conference on Image Processing, 2002, Page(s): I-900 - I-903 vol.1.

[17] Swain M J, Ballard D H. Color indexing[J]. International journal of computer vision 7.1, 1991, Page(s):11-32.

# 致谢

经过一年多的实验研究，我终于将这篇毕业论文写完，在论文的写作和实验过程中不可避免地遇到了很多困难，但在老师和同学们的帮助之下我跨越了重重障碍。在这里，我尤其要感谢我的导师黄翰老师，黄老师从始至终都给予了我很多教导，在实验过程、文写作和修改都多次给我解答难题，指导我完成相关工作，给了我很大的帮助。黄老师为人师表，工作认真，教人不悔的工作态度给我留下很深的印象。

感谢培养我的华南理工大学，给我提供了一个良好的学习环境，在校园里总能感觉到一股浓厚的学术氛围；以及研一一年给我上过课的老师，谢谢你们对我的倾囊教育，我一定会好好运用你们传授给我的知识为社会做贡献，为母校增光。

感谢这篇论文所涉及的各位学者。本文引用参考了多位学者的学术文献，他们的研究成果给了我很多帮助和启发，让我受益匪浅。

最后，感谢自我出生以来一直养育我的父母，他们从小就对我严格要求，教育问题上从不马虎，为的就是让我能上更好的学校，学习更多的知识，将来能出人头地。养育之恩，永生难报，祝愿父母一生平安快乐。

由于我的研究有限，所写论文难免有不足之处，恳请各位老师和学友批评指正！