

**本科毕业设计（论文）说明书**

**题目： 智能视频摘要软件**

**学 院 软件学院**

**专 业 软件工程**

**学生姓名 刘 远 一**

**学生学号 200730553455**

**指导教师 黄 翰**

**提交日期 2011 年 6月1 日**

**毕 业 设 计 （论文） 任 务 书**

兹发给 07级4 班学生 刘远一 毕业设计（论文）任务书，

内容如下：

1、毕业设计（论文）题目： 智能视频摘要软件

2、应完成的项目：

（1）软件的需求分析

（2）软件的代码开发

（3）软件的测试

（4）软件的相关文档写作

3、参考资料以及说明：

（1）Wilhelm Burger Mark J.Burge ;;数字图像处理——java语言算法描述［M］ 清华大学出版社 2010年2月

（2）田莘;;基于MeanShift算法的目标跟踪问题研究[D];西安科技大学;2010年

（3）汪冲;席志红;肖春丽;;基于背景差分的运动目标检测方法[J];应用科技;2009年第10期

（4）朱明旱;罗大庸;;基于帧间差分背景的运动物体检测与跟踪[J];计算机测量与控制;2006年08期

（5）卢璇;雷航;郝宗波;;联合多特征的自动CamShift跟踪算法[J];计算机应用;2010年03期

（6）肖思兴;;复杂场景下的运动目标检测与跟踪算法研究[D];厦门大学;2009年

（7）曹志刚;李宇成;;[运动目标的前景与背景实时分割](http://cpfd.cnki.com.cn/Article/CPFDTOTAL-YJZD201005001235.htm)[A];全国冶金自动化信息网2010年年会论文集[C];2010年

4、本毕业设计（论文）任务书于 年 月 日发出，应于 年 月 日前完成，然后提交毕业考试委员会进行答辩。

指导教师 签发， 年 月 日

教研组（系、研究所）负责人 审核， 年 月 日

摘 要

近年来，随着社会的迅速进步，银行、电力、交通、安检以及军事设施等领域对安全监控系统的需求与日俱增。在现在的视频监控领域中，一般来说很难有这么多人力资源来监控或者查看这么庞大的视频资源。

本文针对上述视频数据量大的问题，开发出智能视频摘要软件，将一个监控视频中的所有运动物体的运动提取出来，放到同一个视频中去播放，这样就大大降低了监控视频所需要的人力资源消耗。这对提高监控效率以及安全防护等起着很大的作用。

本文先对视频摘要软件的主要功能进行简单的介绍，对适用范围以及存在问题进行说明；而后对视频摘要软件的处理流程进行介绍；然后对整个视频摘要软件的模块构成和算法原理进行详细分析。

**关键词**：视频摘要；运动物体检测；智能监控

Abstract

In recent years, with the rapid social progress, the requirements of Safety Monitoring System Requirements in banking, electricity, transportation, security and military facilities. In the present field of video surveillance, it is difficult to have so many human resources to monitor such a large video.

In this article, to solve the problem of large amount of video data, I developed a software named Summary of Intelligent Video Software. It can extract all of the movement events in a surveillance video and play all of the movement events in a single video. This greatly reduces the need for surveillance video of the consumption of human resource. This will improve monitoring efficiency and is good for security.

This article first introduces the main function, scope and exiting problems of the software. Then introduces the processing of the software. At last, the article detailed analysises the modules of the software and the algorithm theory.

**Keyword**: Video summary, Moving object detection, Intelligent monitoring

**第一章 绪 论**

1.1 背景与意义

近年来，随着社会的迅速进步和国力的不断增强，社会生产中各行各业对安全防范和现场监控记录需求与日俱增，并且要求越来越高。虽然监控系统己经广泛地被使用，但实际的大部分监控工作仍然是由监控人员肉眼完成的，而且现有的视频监控系统通常只是录制视频图像，提供的视频信息都是监控场景原始视频片段，不管是人工实时监控或者是非实时查看监控视频都会耗费很多的人力资源。

此软件名为《智能视频摘要软件》，主要功能是分析视频，运动物体跟踪和检测，运动事件的提取，单个运动事件播放，全部运动事件播放等。此软件主要应用于监控视频的分析与提取，比如说一个24小时的监控视频，在这24小时内进入这个场景的人或者其他运动物体不频繁或者很少，就可以用这个软件将这些运动事件提取出来，然后用户查看监控视频的时候就可以有选择性的一个个运动事件去播放，或者将这些事件整合到同一个视频里面一起播放。这样做的好处就是不用把原始的24小时的监控视频从头到尾播放一次，方便使用者监控视频，节省很多时间。

这个软件比较适用于场景不复杂的监控视频，比如说室内监控，一些偏僻的街道监控等。不适用于人来人往的闹市，这些运动物体太多太复杂，如果一个监控视频每时每刻都有运动物体，那用这个软件提取出运动事件也就没意义了，还不如从头到尾看完监控录像。

1.2 国内外同类产品的技术情况

现有的智能视频监控技术主要有：特定物体识别，车牌识别，车流统计，人脸检测和打架等反常行为识别等。以上这些智能视频监控技术都带有确定性，监控目标具有一定的颜色、形状等特征，其未能解决的问题是，如果你并不知道某一物体的具体特征的时候，就要以人工的方式去将一个监控视频从头到尾查看一次，这样非常耗费人力资源和时间资源。

1.3 本论文的主要工作与内容框架

本文针对智能视频分析进行了一些研究，提出了一种提取视频摘要的方法，阐述了智能视频摘要软件的基本原理，以及对保存一个事件的数据结构进行描述，还有本地分析文件的数据格式等。然后介绍了怎么将各个技术环节以及系统模块逐个整合起来，最终得到一个完整的智能视频摘要软件。在论文最后，还对智能视频摘要软件进行了测试，并撰写测试报告。

第二章 核心算法：帧间差分法

2.1 帧间差分法的基本原理

帧间差分法是一种通过对视频图像序列中2个或者3个相邻帧采用基于像素的时间作差分运算来获得运动目标轮廓的方法，它可以很好地适用于存在多个运动目标和摄像机移动的情况。当监控场景中出现物体运动时，相邻的帧之间会出现较为明显的差异，相邻的两帧相减，得到两帧图像亮度差的绝对值，判断它是否大于一个设定的阈值来判断视频或图像序列是否有运动特性，确定图像序列中有无物体运动。将视频中的图像序列进行逐帧的做差操作，相当于对图像序列进行了时域上的高通滤波。

帧间差分法的优点是：算法实现相对简单，运行较快，复杂度低；能够适应各种动态环境，对光线等因素的变化不敏感。其缺点是：不能提取出对象的完整区域，只能提取出边界；同时依赖于选择的帧间时间间隔。对快速运动的物体，需要选择较小的时间间隔，如果选择太大的时间间隔，就会出现物体在相邻的两帧中没有重叠的情况，结果是检测为两个不同的物体：而对慢速运动的物体，应该选择较大的时间间隔，如果时间选择太小的时间间隔，就会出现物体在相邻两帧中几乎完全重叠的情况，结果是检测不到物体。

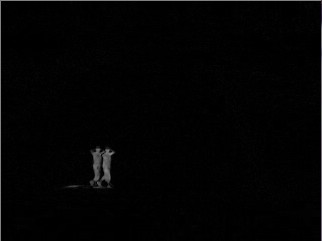
2.2 帧间差分法进行运动物体检测的原理以及流程

首先收集某一段时间内的原始视频序列图像，，…，用于运动物体的检测。设，，是其中的3幅连续图像（1<=a<=n-2，b=a+1，c=b+1），分别对应图2-1中的图（a）、（b）、（c）。如图2-2则是，，经过灰度处理的灰度图像，。然后用先作为基础背景，进行 — 运算，再将这2者的差值进行二值化处理后得到图2-3（a）所示的二值图像；同理再用作为基础背景，进行 — 运算，再将这2者的差值进行二值化处理后得到图2-3（b）所示的二值图像。图2-3（a）（b）两幅图中的白色像素块就是运动物体的大致轮廓，然后我们再用opencv中给出的轮廓提取函数将白色像素块的矩形轮廓大小和左上角的坐标提取出来，然后在原始图像上面用检测出来的矩形框圈出运动物体，就完成了运动物体检测的过程，如图2-4是被圈出的运动物体。帧差法的算法流程如图2-5所示。

 （a） （b） （c）

图2-1 视频原始序列图像

（a） （b） （c）

图2-2 视频灰度序列图像

（a） （b）

图2-3 视频相邻两帧图像灰度图之差

（a） （b）

图2-4 运动检测结果

图2-5

开始

图像序列采集

图像增强

滤波处理

帧差法处理

得出运动物体轮廓

视频是

否结束

结束

是

否

2.3 本章小结

本章主要描述了智能视频摘要软件核心算法，即帧差法的基本原理，并且给出了算法的主要流程，算法的截图结果等。

第三章 软件需求分析以及配置要求

智能视屏摘要软件旨在开发一个软件系统，该系统能将视频中的运动事件作为视频的摘要提取出来并可以供用户查看摘要事件。智能视频摘要软件主要分为2个大功能：视频摘要提取（视频分析）和视频摘要的播放。

3.1 视频摘要提取（视频分析）

3.1.1 功能概述

视频摘要提取功能，也可以说成视频分析功能，就是用运动物体检测与跟踪的方法对视频进行分析，将视频中所有运动事件提取出来并作为视频的摘要，然后将摘要信息保存。这里要求系统支持所有avi格式的视频，能从本地读入视频文件，然后能迅速地分析整个视频，并将视频摘要信息提取出来，然后把这个视频的所有摘要（运动事件）信息按照一定格式保存到本地文件，这样做的目的是进行摘要备份，避免二次分析视频导致浪费时间。分析完视频之后给用户提供一个事件列表，供用户点选。

3.1.2 功能规定

下面将会列出视频摘要提取功能的细化步骤的输入项与输出项。

选择本地视频

表3-1

|  |  |
| --- | --- |
| 输入 | 本地avi视频 |
| 操作 | 打开文件选择窗口选择一个avi视频文件 |
| 输出 | 在界面显示读入的avi视频路径 |

选择本地文件夹（批量处理）

表3-2

|  |  |
| --- | --- |
| 输入 | 存放有多个avi视频的本地文件夹 |
| 操作 | 打开文件夹选择窗口选择一个本地文件夹 |
| 输出 | 在界面显示读入的文件夹路径 |

读入分析文件

表3-3

|  |  |
| --- | --- |
| 输入 | 与读入视频相对于的视频分析文件 |
| 操作 | 在分析视频的时候选择读入已有视频分析文件 |
| 输出 | 根据读入的分析文件显示视频摘要事件列表 |

视频处理

表3-4

|  |  |
| --- | --- |
| 输入 | 读入的avi视频 |
| 操作 | 对视频进行运动物体检测与跟踪操作 |
| 输出 | 视频的摘要事件列表 |

过滤摘要事件

表3-5

|  |  |
| --- | --- |
| 输入 | 经过视频处理后得到的视频的摘要事件列表 |
| 操作 | 将视频处理后得到的视频摘要事件列表进行过滤，把过小的事件删除 |
| 输出 | 过滤后的视频摘要事件列表 |

写入分析文件

表3-6

|  |  |
| --- | --- |
| 输入 | 视频的所有摘要事件信息 |
| 操作 | 将视频的所有摘要事件信息通过一定的格式写入本地文件保存 |
| 输出 | 保存着视频所有摘要信息的本地txt文件 |

产生摘要事件列表

表3-7

|  |  |
| --- | --- |
| 输入 | 经过过滤后得到的视频的摘要事件列表 |
| 操作 | 将保存在内存中的视频摘要事件列表显示在主界面上 |
| 输出 | 将保存在内存中的视频摘要事件列表显示在主界面上 |

生成所有事件视频

表3-8

|  |  |
| --- | --- |
| 输入 | 视频的所有摘要事件信息 |
| 操作 | 根据视频所有摘要事件信息将每一个视频事件提取出来，整合到同一个视avi频文件里面保存 |
| 输出 | 保存有所有摘要事件的本地avi视频 |

3.2 视频摘要的播放

3.2.1 功能概述

视频摘要的播放，也就是将视频中的视频摘要重播再现。这里分为两种方式：一种是单个摘要事件的播放，用户可以在第一个功能中生成的事件列表选择某一个具体事件进行播放；另一种是所有事件的播放，这里将视频中的所有摘要事件集中在同一个视频中播放，并对立面的事件进行半透明处理，给每一个事件进行标号，以便识别。当用户从所有事件播放中找到自己感兴趣的事件时，可以根据事件标号在事件列表中找到对应事件，进行单独播放。

3.2.2 功能规定

下面将会列出视频摘要播放功能的细化步骤的输入项与输出项。

1）播放单个事件

表4-9

|  |  |
| --- | --- |
| 输入 | 视频摘要事件的编号 |
| 操作 | 根据用户选择的摘要事件编号将此事件播放出来 |
| 输出 | 在播放控件中重现的事件 |

2）播放所有事件

表3-10

|  |  |
| --- | --- |
| 输入 | 所有视频摘要事件信息 |
| 操作 | 调出本地所有视频摘要事件的视频并播放 |
| 输出 | 在播放控件中重现所有事件 |

3.3 系统用例图和流程图

如图3-1是智能视频摘要软件系统的用例图，对每一个用例的具体描述请参照表3-1到表3-10；图3-2是智能视频摘要软件系统的总体流程图。



图3-1 系统用例图

读入本地avi视频文件

生成事件列表

是否分析过

用帧间差分法分析视频

读入本地分析文件

否

是

生成本地分析文件

播放单个事件

播放所有事件

关闭软件

图3-2 系统流程图

3.4 配置需求

3.4.1 硬件环境需求

Intel双核CPU 2.0G以上

内存DDR2代，3G以上

256M以上的独立显卡

10G以上的硬盘空间，用以存放视频结果

**3.4.2 软件环境需求**

操作系统：windows xp或者windows 7

预安装xvid视频编码器

装有或者以上版本的Emgu.CV2.1，并配置好相应的环境路径

3.5 本章小结

本章先是介绍了智能视频摘要软件的功能需求，并对它们的输入输出等进行了细化说明，然后给出了整个软件系统的用例图以及整体流程图，最后简单的说明了运行该软件所需要的硬件环境和软件环境。

第四章 系统设计

4.1 系统用户界面设计



图4-1 智能视频摘要软件主界面

**系统主界面**如图4-1所示。

“选择视频文件”按钮：选择一个本地视频文件。

“选择文件夹”按钮：选择一个包含多个视频的本地文件夹。

显示所选择的文件路径

“分析视频”按钮：点击此按钮进行视频分析。

显示分析视频的进度条。

显示分析结果。

“播放选中事件”按钮：播放从列表中选中的事件。

“播放所有事件”按钮：播放所有事件。

显示视频摘要事件的列表框。



图4-2 打开视频界面

**打开视频**界面如图4-2所示。



图4-3 播放单个摘要事件界面

**播放单个视频**界面如上图4-3所示。

播放事件的标号。

被框选中的事件。



图4-4 播放所有事件界面

**播放所有事件**界面如图4-4所示。

图中对每一个运动事件都进行半透明处理，并且对每一个事件都进行标号，以便识别，标号跟事件列表的标号相对应。

4.2 系统模块设计

表4-1 系统模块设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **系统模块名称** | | **模块功能名称** | **在系统中的类结构** |
| **主**  **程**  **序** | | **UI模块** | 软件系统界面 | VideoMainForm.Designer.cs  VideoMainForm.cs  PlayForm.Designer.cs  PlayForm.cs  Program.cs |
| 界面事件响应 |
| **视频处理模块** | 分析视频 | VideoAnalyzeProcess.cs  EventNode.cs  Global.cs |
| 播放单个/所有事件 |
| 生成所有事件视频文件 |
| **本地文件操作模块** | 本地文件读入 | FileOperation.cs |
| 本地文件写入 |

如表4-1阐述了只能摘要软件系统的3大主模块：UI模块、视频处理模块和本地文件操作模块，其中每个模块又被分为数个小的功能模块。

4.2.1 UI模块设计

该子系统是整个软件系统的表现层，是用户和软件系统的交互界面，通过人为点击软件界面上的按钮等触发响应事件对视频进行操作。

1）功能

表4-2 UI模块设计表

|  |  |  |  |
| --- | --- | --- | --- |
| 程序名称 | UI界面 | 标识符 | VideoMainForm |
| 相关数据库表 | 无 | | |
| 输入 | 鼠标点击事件，键盘输入字符 | | |
| 输出 | 显示事件列表，播放事件显示，分析结果显示等 | | |
| 程序处理说明 | 此模块主要是接收用户从鼠标或者键盘输入，并对相应的按钮事件进行响应，调用其他模块的接口函数进行操作，然后显示结果。 | | |

2）性能

本模块要求UI控件显示美观，响应快速。

3）输入项

鼠标点击事件，键盘输入字符。

4）输出项

显示事件列表，播放事件显示，分析结果显示等。

5）算法

无。

6）流程逻辑

鼠标点击，键盘输入事件响应

调用分析视频模块或者本地文件操作模块进行处理

显示处理结果

图4-5 UI模块流程逻辑图

7）接口

此模块没有接口。

4.2.2 视频处理模块

**4.2.2.1 视频分析**

1）功能

表4-3 视频分析模块设计表

|  |  |  |  |
| --- | --- | --- | --- |
| 程序名称 | 视频分析 | 标识符 | analyzeVideo |
| 相关数据库表 | 无 | | |
| 输入 | 本地avi视频路径，软件主界面的对象 | | |
| 输出 | 保存着事件详细信息的EventNode列表 | | |
| 程序处理说明 | 此模块主要是对视频进行分析，通过帧间差分法和轮廓匹配对视频进行运动物体的检测和跟踪，然后记录下运动事件的详细信息，包括起止帧，轨迹，轮廓等，保存在EventNode数据结构中，最后得出一个保存事件列表的EventNode列表。 | | |

2）性能

本模块要求分析视频的事件要小于源视频文件时间长度的五分之一。

3）输入项

本地avi视频路径。

4）输出项

保存着事件详细信息的EventNode列表。

5）算法

帧间差分法。

6）流程逻辑

读入本地avi视频文件

使用帧差法进行运动物体的检测

使用轮廓匹配进行运动物体跟踪

提取事件的详细信息

将信息保存到EventNode数据结构中

生成事件列表

图4-6 视频分析模块流程逻辑图

7）接口

此模块的接口函数为static public void analyzeVideo(string filePath, VideoMainForm form)，其中参数filePath为本地视频路径，参数form为软件主界面对象。

**4.2.2.2 播放单个事件**

1）功能

表4-4 播放单个事件模块设计表

|  |  |  |  |
| --- | --- | --- | --- |
| 程序名称 | 播放单个事件 | 标识符 | playSingleEvent |
| 相关数据库表 | 无 | | |
| 输入 | 本地avi视频路径，事件的下标 | | |
| 输出 | 显示框出轮廓的指定事件 | | |
| 程序处理说明 | 此模块主要是从EventNode数据结构中读取指定事件的详细信息，然后将此事件播放出来，这个事件必须用矩形框框出最小轮廓，以便识别。 | | |

2）性能

本模块无性能要求。

3）输入项

本地avi视频路径。

4）输出项

显示框出轮廓的指定事件。

5）算法

此模块无算法。

6）流程逻辑

读入本地avi视频文件

从EventNode数据结构中读取指定事件的详细信息

在视频中定位事件的时间轴

显示事件的运动

用矩形框框出事件的最小轮廓

图4-7 播放单个事件模块流程逻辑图

7）接口

此模块的接口函数为static public void playSingleEvent(int index)，其中参数index为选中事件的下标。

**4.2.2.3 播放所有事件**

在这个模块中，播放事件请参考4.2.2.2，这里主要介绍生成本地视频文件。

1）功能

表4-5 播放所有事件模块设计表

|  |  |  |  |
| --- | --- | --- | --- |
| 程序名称 | 生成所有事件的视频文件 | 标识符 | playAllEvent |
| 相关数据库表 | 无 | | |
| 输入 | 本地avi视频路径 | | |
| 输出 | 所有事件的本地视频文件 | | |
| 程序处理说明 | 此模块主要是从EventNode数据结构中读取所有事件的详细信息，将这些事件的最小轮廓提取出来，整合到同一个视频中去播放，并保持到本地avi视频文件中。 | | |

2）性能

要求能同时生成200个以上的事件视频（这受限于内存大小）。

3）输入项

本地avi视频路径。

4）输出项

所有事件的本地视频文件。

5）算法

此模块无算法。

6）流程逻辑

读入本地avi视频文件

从EventNode数据结构中读取所有事件的详细信息

提取每一个事件的最小轮廓大小

将所有事件每一帧整合到同一个视频中

生成本地视频文件

图4-8 播放所有事件模块流程逻辑图

7）接口

此模块的接口函数为static public void playAllEvent()。

4.2.3本地文件操作模块

该子系统是实现本地文件读入以及写入。

1）功能

表4-6 本地文件操作模块设计表

|  |  |  |  |
| --- | --- | --- | --- |
| 程序名称 | 本地文件操作 | 标识符 | FileOperation |
| 相关数据库表 | 无 | | |
| 输入 | 本地文件路径，保存着事件详细信息的EventNode列表 | | |
| 输出 | 本地分析文件，保存着事件详细信息的EventNode列表 | | |
| 程序处理说明 | 此模块主要按照一定的数据结构将视频事件的信息写入本地分析文件保存，或者按照一定的数据格式将本地文件的事件信息读入到计算机内存中。 | | |

2）性能

本模块无性能要求。

3）输入项

本地文件路径，保存着事件详细信息的EventNode列表。

4）输出项

本地分析文件（写入），保存着事件详细信息的EventNode列表（读入）。

5）算法

无。

6）流程逻辑

新建本地文件

按照一定格式将EventNode的信息写入本地文件

关闭本地文件

打开本地文件

按照一定格式将本地文件的事件的信息读入EventNode

关闭本地文件

图4-9 本地文件操作模块流程逻辑图

7）接口

此模块写入文件的接口为static public void writeToFile(string filePath)，读入文件的接口为static public void readFromFile(string filePath)。

4.3 系统数据结构设计

由于智能视频摘要软件没有使用到数据库，所以这里的数据结构不包括数据库设计说明，这里只介绍软件系统用到的2个主要数据结构：EventNode类和本地分析文件数据结构。一个EventNode对象保存一个视频摘要信息，一个本地分析文件保存一个视频所有视频摘要信息。

4.3.1 EventNode类结构设计

表4-7 EventNode类数据结构

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **变量的含义** |
| int | startFrame | 视频摘要事件的开始帧。 |
| int | endFrame | 视频摘要事件的结束帧。 |
| List<Rectangle> | trackList | 一个Rectangle类型的List列表，保存了整个视频摘要事件每一帧（或者间隔n帧）的矩形轮廓以及矩形轮廓左上角的坐标。 |
| Rectangle | rect | 此变量在运动物体检测与跟踪算法中，保存了上一帧中运动事件的矩形轮廓以及其左上角坐标。 |
| bool | mark | 此变量在运动物体检测与跟踪算法中，标记了该视频摘要事件是否已经完结。 |

表4-8 Rectangle类数据结构

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **变量的含义** |
| int | X | 矩形轮廓左上角的x坐标。 |
| int | Y | 矩形轮廓左上角的y坐标。 |
| int | Width | 矩形轮廓的长。 |
| int | Height | 矩形轮廓的高。 |

如表4-7写出了EventNode的数据结构，一个EventNode变量记录了一整个视频摘要事件的所有信息。主要变量是startFrame、endFrame、trackList3个，前两者分别表示事件的开始帧和结束帧，然后trackList主要是保存从开始帧到结束帧这个帧段中摘要事件在每一帧（或者间隔n帧）中出现的位置以及其矩形轮廓大小。这种数据结构的好处是节省空间，与把摘要事件作为连续的图片提取出来相比，只是保存了事件出现的位置以及轮廓大小的EventNode节省的空间要大好多，一个是保存图片的List，一个是保存矩形轮廓以及轮廓左上角位置。

如表4-8写出了Rectangle类的数据结构，Rectangle类是C#中自带的类，它表示了一个矩形的长宽以及矩形左上角的坐标。

4.3.2 本地分析文件数据结构

由于此智能视频摘要软件是一个单机版的视频分析软件，要是采用数据库的方式来存储视频摘要事件的信息就要求用户对数据库配置有一定的了解，考虑到数据库配置起来比较麻烦，为了让用户更方便地使用这个智能视频摘要软件，所以采用将视频摘要信息保存到本地的txt文件这种保存方式。众所周知，分析一个视频需要的时间成本很高，将分析后的视频摘要信息备份保存，可以避免对视频进行二次分析，快速从本地文件读入视频的摘要信息。

表4-9 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4-10 事件的节点结构

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

如表4-9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，包含了视频的总摘要事件数、视频处理时的帧间隔和视频的帧率；另一个是摘要事件的列表。在摘要事件列表中，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4-10为本发明保存某一个视频摘要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轮廓的左上角坐标（x坐标和y坐标），和此运动物体矩形轮廓的长，宽。

4.4 接口设计

4.4.1 用户接口

本软件系统采用图形用户接口，以键盘鼠标为用户的输入设备，方便用户对视频的操作。同时在有响应事件或者异常事件发生的时候，会用友好的方式对用户进行通知和提示等，包括视频分析结果显示，操作错误窗口提醒等。

4.4.2 外部接口

本软件系统可以把视频摘要信息在本地txt文件中保存或者读入，可以节省用户二次分析浪费不必要的时间。视频的输入可以是单个视频，也可以是包含有多个视频文件的文件夹，即批量视频处理，每一个视频的所有摘要事件会用avi格式的视频保存下来，方便用户查阅。

4.4.3 内部接口

本系统的内部各模块的接口主要放在视频处理模块以及本地文件处理模块中，供UI模块调用。

**4.4.3.1 视频处理模块**

在视频处理模块中，接口函数用类VideoAnalyzeProcess类进行封装，主要对视频进行各种操作，主要的接口函数有：

static public void analyzeVideo(string filePath, VideoMainForm form)

分析视频主函数，提供给UI模块调用，参数filePath为视频文件路径，参数form为主窗口对象，通过此参数可以在窗口实时更新视频的分析进度以及分析结果。此函数将指定路径的视频分析，然后得出分析结果，生成事件列表，并将所有事件信息写入本地文件进行保存，写入文件操作需要调用本地文件操作模块。

static public void playSingleEvent(int index)

播放单个事件主函数，提供给UI模块调用，参数index为在事件列表中选中事件的下标。

static public void playAllEvents()

播放所有事件主函数，提供给UI模块调用，这里播放一个包含所有事件的avi格式视频，视频长度取决于最长事件的时间。

**4.4.3.2 本地文件处理模块**

用类FileOperation进行封装，实现将视频摘要事件信息按一定格式写入本地文件或者按一定格式从本地文件读入事件信息的功能，主要接口函数有：

static public void writeToFile(string filePath)

将分析完的视频摘要事件信息按照一定格式写入本地文件进行保存，参数filePath为本地文件路径。这里的文件数据格式请参照5.3节的数据结构设计。

static public void readFromFile(string filePath)

按照一定的数据格式读入本地视频分析文件，此文件包含视频中所有摘要事件的信息，参数filePath为本地视频分析文件的路径。这里的文件数据格式请参照4.3节的数据结构设计。

4.5 代码清单

下表4-11列出了本项目所包含的所有源代码文件，以及对应的说明。

表5-11 代码清单

|  |  |
| --- | --- |
| **文件名** | **程序功能用途** |
| Program.cs | 这个类为整个C#窗口应用程序的入口 |
| Globals.cs | 保存了一些全局变量 |
| VideoMainForm.Designer.cs | 软件主界面显示的类 |
| VideoMainForm.cs | 软件主界面事件响应处理 |
| VideoMainForm.resx | 此文件保存了软件主界面的资源文件 |
| PlayForm.Designer.cs | 播放界面显示的类 |
| PlayForm.cs | 播放界面事件响应处理 |
| PlayForm.resx | 此文件保存了播放界面的资源文件 |
| EventNode.cs | 一个摘要事件的数据结构以及相应操作 |
| VideoMainForm.cs | 负责视频处理的所有操作 |
| FileOperation.cs | 负责本地文件读写操作 |

4.6 本章小结

本章主要是介绍了整个软件系统的设计，包括软件界面设计，系统内部模块设计，并对系统的各个模块进行深入分析，然后还介绍了本软件系统所用到的数据结构，最后是列举了整个系统的代码清单文件。

第五章 软件运行与测试结果分析

5.1 系统运行实例

系统运行实例这部分主要介绍智能视频摘要软件的运行步骤，以及这些步骤的预期输入与输出。

5.1.1 分析单个视频

步骤1：在软件的主界面点击“选择视频文件”按钮，然后在跳出的文件选择框中选择一个avi格式的视频（opencv只支持avi格式视频的处理），选择完视频之后在“选择视频文件”按钮下方的文本标签会显示所选择视频文件的路径。

步骤2：确认选择的视频文件无误后，点击“分析视频”按钮，系统就会自动根据刚才选择的视频文件路径找到对应视频进行分析（如果有本地分析文件则读入分析文件），在“分析视频”按钮下方的进度条同步显示了现在的分析进度。当分析完视频后，在进度条下方的文本标签会显示视频的分析结果，分析结果包括视频的总摘要事件数以及视频的最大事件持续时间。另外在软件界面的右边会显示视频所有摘要事件的列表，列表信息包含事件的编号，以及每个事件的开始时间和结束时间。分析完一个视频后如图5-1所示：

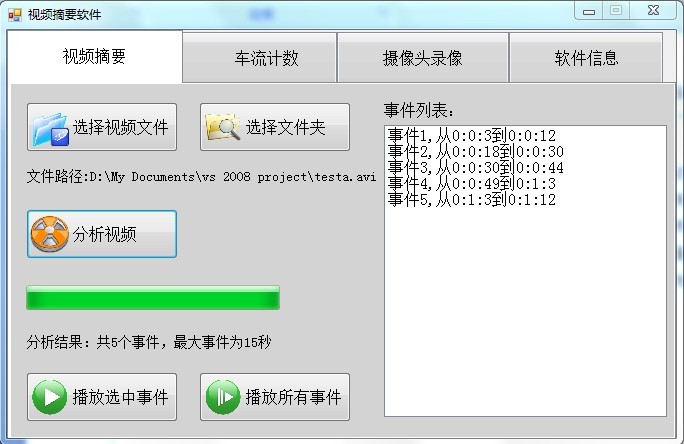


图5-1 分析单个视频结果显示

5.1.2 批处理视频

步骤1：点击“选择文件夹”按钮，然后在跳出的文件夹选择框中选择一个包含有多个avi视频的文件夹，注意这里如果文件夹中夹杂有其他格式的文件，系统会自动过滤掉，只读取avi格式的视频文件，然后在“选择文件夹”按钮下方的文本标签会显示出这个文件夹的路径。

步骤2：点击“分析视频”按钮，系统就会自动根据刚才选择的文件夹路径找到对应视频逐个进行分析，在“分析视频”按钮下方的进度条同步显示了当前正在被分析视频的进度，同时在进度条下方的文本标签会显示当前正在分析第几个视频，以及总视频的个数。当分析完所有视频后，在进度条下方的文本标签会显示“所有视频分析完毕”。因为这里的界面显示跟分析单个视频的界面大同小异，只是在事件列表中没有事件显示而已，所有就不做截图了。

备注：不管是分析单个视频还是分析批处理视频，都会在视频当前目录新建一个名字叫“analyze”的文件夹，用以存放分析文件以及所有摘要事件的avi视频文件。本地视频分析文件的名字格式为：视频文件名字.txt，所有摘要事件的avi视频文件名字就用源视频文件的名字。假如你分析的视频名字为“test1.avi”，那生成的本地分析文件名字就是“test1.avi.txt”，所有摘要事件的avi视频文件名字为test1.avi。这里值得一提的是，由于所有摘要事件的视频文件时保存在源视频文件所在目录的子目录（analyze）下面的，所有不用担心出现同名的文件，如下图5-2所示：



图5-2 生成的本地文件目录结构

5.1.3 播放单个事件

步骤1：在完成5.1.1介绍的分析单个视频之后，在事件列表会显示出一系列事件，用户在事件列表中选择自己想观看的事件。

步骤2：点击“播放选中事件”按钮，系统会跳出一个视频播放界面，并且自动将视频的播放进度跳到该事件的开始事件点，用户就可以方便的观看该事件了，在播放过程中用户可以随意进行拖动进度条，暂停等操作，如图5-3所示



图5-3 播放单个事件

5.1.4 播放所有事件

步骤1：点击“播放所有事件”按钮，系统会跳出一个类似于播放单个事件的播放界面，然后读取所有摘要事件的视频文件将其播放。不同的是，所有事件的视频中每一个事件左上角都会有一个事件编号，并且将重叠的事件进行半透明处理，方便用户识别，一旦你对某个编号的事件感兴趣，那你就可以在事件列表中选中那个事件，进行单独播放。所有事件的运行效果如下图6-4所示：



图5-4 播放所有事件

5.2 测试环境

CPU：AMD双核，2.30GHz

内存：DDR2代，3.25GB

显卡：ATI Radeon HD 3850，512MB

硬盘空间：10G以上的硬盘空间，用以存放视频结果

5.3 测试结果分析

软件的测试主要是性能测试，其中又可以分为压力测试和速度测试。

5.3.1 压力测试

这里所谓的压力测试就是测试这个智能视频摘要软件的瓶颈所在，比如说当分析一个时间很长的视频文件，系统会不会崩溃。鉴于软件本身的实现方式以及其算法，在进行视频处理的时候只用了单线程，而且对视频或者图像进行操作的函数都是opencv中一些基本的函数，也没有涉及复杂的算法，因此从这点上来看软件所占用的cpu资源并不会很多。反倒是视频处理过程中对内存占用率相当高，因为随着视频的分析，摘要事件个数越来越多，所需要用到的内存来保存这些摘要事件或者生成所有摘要事件的视频的时候占用的内存越大，下表5-1的数据用来显示当视频长度不一样或者事件数不一样的时候软件的运行情况：

表5-1 压力测试表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **视频编号** | **视频长度/时：分：秒** | **分析时间/时：分：秒** | **事件个数/个** | **内存最大占用量/MB** | **CPU平均占用率** | **软件运行情况** |
| 1 | 00:05:09 | 00:00:30 | 25 | 148.4 | 48% | 正常运行 |
| 2 | 00:30:00 | 00:03:50 | 18 | 110.5 | 47% | 正常运行 |
| 3 | 01:01:00 | 00:7:30 | 100 | 820.6 | 51% | 正常运行 |
| 4 | 01:00:00 | 00:10:17 | 199 | 1557.9 | 52% | 内存溢出 |
| 5 | 04:47:00 | 00:15:44 | 20 | 147.1 | 48% | 正常运行 |
| 6 | 09:36:00 | 00:25:44 | 29 | 185.7 | 51% | 正常运行 |

从表5-1中可以看出（1）不管视频长度是多少，视频的摘要个数是多少，CPU的占用率都基本维持在50%左右，这说明此软件在运行过程中并不会占用系统太多CPU资源，也就是说CPU不是智能视频摘要软件的瓶颈；（2）同时从表中“视频长度”那列和软件“运行情况”那列的结果来看，软件可以处理很长的视频这是无可非议的，不管你视频有多长，只要事件数比较少，那就能正常运行，由于我手上也没有比6号视频更长的视频了，所有也不能做更深入的测试；（3）从“事件个数”、“内存最大占用量”和“软件运行情况”这3列可以明显看出事件个数和内存最大占用量的关系，视频的摘要事件数量越多，所需要的内存就越大，而这部分内存主要用在生成所有摘要事件的avi视频文件中，4号视频有199个事件，内存占用了1557.9MB，直接导致内存溢出。

所以我得出的结论就是：当视频中的摘要事件数太多的时候，将会在生成所有摘要事件的avi视频时候占用很大内存，甚至导致内存溢出。那么从这看来，智能视频摘要软件的瓶颈就是摘要事件数以及计算机的物理内存大小。后来我控制事件个数进行测试，发现在150个事件左右的时候内存会溢出，就是说此软件目前最多能处理150个事件以内的监控视频。

关于瓶颈的解决方案：（1）改进生成所有摘要事件avi视频的算法，降低其内存消耗，目前正在往这方面去努力，但是还没有得出更好的算法；（2）将所有摘要事件的视频拆分成几个，比如说一个500摘要事件的所有事件视频文件可以分成5个100摘要事件的视频，这个方法相对于第一个方法更容易实现。

5.3.2 速度测试

速度是指软件分析视频的速度，这里主要研究智能视频摘要软件的分析速度跟什么因素有关。

速度测试一：从上面的表5-1可以知道软件分析一个视频所需要的时间随着视频长度增加而增加的，并且不难得出，当视频的其他参数条件（分辨率，单位时间内的事件数，视频帧率）都一样时，视频分析时间跟视频的总长度是成正比的。这个关系当然不是从直接测试得出来的，因为很难找到分辨率，单位时间内的事件数，视频帧率都相等的测试视频，只是根据之前表5-1推测出来的一个视频总长度跟视频分析速度之间的关系。

速度测试二：

秒



个

图5-5 事件个数与分析时间关系图

如图5-5所示，横坐标表示事件个数，纵坐标表示分析所需时间。这个测试我用了23个视频，每个视频除了事件数的其他参数都相同：帧率=15fps，长度=30min，分辨率=320\*480。由图5-5可以看出，软件分析视频的时间是随视频摘要事件数的增加而增加的，因为测试环境是电脑，有很多不稳定因素，他们的关系曲线大致可以拟合成一条直线。

5.4 本章小结

本章主要介绍了软件的运行流程以及对应的运行效果，并对软件进行了性能测试报告，说出了软件的瓶颈所在和分析了软件的运行速度跟其他因素之间的关系等。

结 论

为了方便视屏监控人员更快速更方便地浏览监控视频，本文提出一种提取视频摘要事件的方法，即将视频中所有的摘要事件都整合到同一个视频文件中播放，让使用者能快速地将这个视频浏览一遍，再选择自己感兴趣的摘要事件进行监控，并可以将视频的摘要信息进行保存，供日后使用。

随着社会的不断发展，视频监控肯定会进入每一个领域，遍布世界的每一个角落，这就意味着监控视频的数量将会很快膨胀，这为我们的安全提供了不可或缺的保障，但是不可能找到这么多人力资源和时间资源对每一个监控视频进行人工分析。智能视频摘要软件刚好能解决这个矛盾，通过分析视频，提取视频感兴趣事件并集中播放来缩短人工检测视频的时间，相信这将会让智能监控领域更进一步。

另外，从对软件的测试结果分析上来看，在软件用到的算法方面可以进行一些改进，提高内存利用率以及提高软件的运行速度，还可以添加一些额外的视频分析功能，用以满足多样化的监控视频需求。

参考文献

[1]王建平,刘伟,王金玲.一种视频运动目标的检测与识别方法[J].计算机技术与自动化,2007,26(3):78-81

[2]石时需,郑启伦,黄翰.基于自适应混合差分的快速视频目标检测法[J].计算机科学.2008(7):224-226.

[3]Tao Xiang, Shaogang Gong. Video Behavior Profiling for Anomoly Detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on. Volume 30, Issue 5, May 2008 Page(s):893-908.

[4]田莘.基于MeanShift算法的目标跟踪问题研究[D]，西安科技大学，2010年.

[5]王建宇,陈熙霖,高文,赵德斌.背景变化鲁棒的自适应视觉跟踪目标模型.软件学报.2006,17(5):1001-1008.

[6]吴思,林守勋,张勇东.基于动态背景构造的视频运动对象自动分割.计算机学报.2005,28(8):1386-1392.

[7]李海波.视频监控中基于人的检测与研究.北京邮电大学,2010.

[8 ]肖思兴.复杂场景下的运动目标检测与跟踪算法研究[D].厦门大学，2009.

[9]王长军.[基于视频的目标检测与跟踪技术研究](http://cdmd.cnki.com.cn/Article/CDMD-10335-2006073629.htm)[D].浙江大学,2006.

[10]曹志刚,李宇成.[运动目标的前景与背景实时分割](http://cpfd.cnki.com.cn/Article/CPFDTOTAL-YJZD201005001235.htm)[A].全国冶金自动化信息网2010年年会论文集[C],2010.

致 谢

经过几个月的不懈努力，我完成了毕业设计论文，同时也学到很多在课堂上学不到的知识。在这里，我首先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，多次给我解答难题，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。

感谢培养我的母校，给我提供了一个良好的学习环境；以及大学四年中给我上过课的老师，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。

感谢大学生活中一起生活，一起学习，一起玩乐的同学们，谢谢你们平时对我的帮助和照顾。

最后，感谢自我出生以来一直养育我的父母，他们不忘昼夜的工作，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。