

# 基于OpenCV的动态背景下的运动目标检测

黄天琪 陈冬琴 殷旭东(常熟理工学院计算机科学与工程学院, 江苏 常熟 215500)

**摘要:** 研究在动态背景下对视频图像序列进行运动目标的检测。主要包括三个步骤, 分别是运动估计, 运动补偿和目标检测。在运动估计中采用的主要是基于特征点匹配算法。这种算法与传统的块匹配算法最大的好处在于它的数据量少, 计算简单迅速而且图像的匹配可靠性更高。最后用计算机视觉库OpenCV进行实现。

**关键词:** 运动目标检测; 动态背景; 特征点匹配; OpenCV

## Moving object detection in dynamic background based on OpenCV

HUANG Tianqi CHEN Dongqin YIN Xudong

(School of Computer Science and Engineering, Changshu Institute of Technology, Changshu 215500, China)

**Abstract:** The moving object detection in dynamic background in the video image sequence is studied. The method includes three steps, they are motion estimation, motion compensation and object detection. At the motion estimation, an algorithm based on the feature points matching is taken. The advantages of this algorithm is that it needs fewer data and indicates faster calculating speed compared to the block matching algorithm and the matching of the video image sequence is more reliable. At last the algorithm is implemented using an computer vision library named OpenCV.

**Key words:** moving object detection; dynamic background; feature points matching; OpenCV

随着社会的发展, 在日常生活中摄像头已经随处可见, 我们经常需要对视频中的人、物进行捕捉、检测、跟踪等图片处理操作, 这就要求对视频中的运动目标进行检测及跟踪。作为视觉领域的一部分, 该技术在安保、交通、军事、医学等各种领域里都有着广泛的应用。

所谓运动目标的检测就是在一段序列图像中检测出变化区域, 并将运动目标从背景图像中提取出来<sup>[1]</sup>。能否正确地检测与分割出运动目标对于后续处理, 诸如跟踪分析等工作有着巨大的影响。常见的运动目标检测方法有: 背景差分法, 帧差法, 累积差分法, 光流法<sup>[1-2]</sup>。本文主要研究的是一种在动态背景下对运动目标进行检测的算法——改进的帧差法。

### 1 静态背景下的检测算法

目标检测算法有多种, 不同的算法有其各自的利与弊。

(1) 背景差分法是事先将背景图像存储下来, 再与观测图像进行差分运算, 实现对运动区域的检测。这种方法能得到较为完整的运动目标信息, 但背景图像必须随着外部条件比如光照等的变化而不断更新, 所以背景模型的获取和更新比较麻烦。

(2) 帧差法是直接比较相邻两帧图像对应像点的灰度值的不同, 然后通过阈值来提取序列图像中的运动区域。这种方法更新速度快, 算法简单易实现, 适应性强, 不需要获取背景图像。但是背景与运动目标间需要有一定程度的灰度差, 否则可能在目标内部产生空洞, 不能完整的提取出运动目标。为了改进相邻两帧间的差分效果, 人们提出了累积差分法。(3) 累积差分法是利用三帧图像计算两个差分图像, 再令其对应像素相乘的算

法。它通过分析整个图像序列的变化来检测小位移或缓慢运动的物体。(4) 光流法是在时间上连续的两幅图像中, 用向量来表示移动前后的对应点, 在适当平滑性约束的条件下, 根据图像序列的时空梯度估计运动场, 通过分析运动场的变化对运动目标和场景进行检测和分割。

### 2 动态背景下的检测算法

#### 2.1 概述

在很多情况下背景图像都不是静态的, 比如有时摄像机安装在一个运动的平台上, 或者它本身也会在一定范围内进行扫描运动, 这样一来就会出现图像背景的全局运动。这时就需要对帧间图像进行配准, 再经过全局背景的运动补偿, 然后才可以通过差分检测检测出目标。

这种通过全局运动估计进行运动目标检测的主要步骤如图1所示。

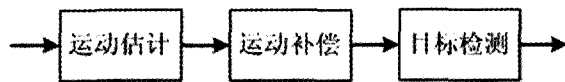


图1 动态背景下运动目标检测步骤图

#### 2.2 运动估计

运动矢量估计的算法有多种, 本文采用的是特征点匹配算法, 也是目前使用较多的一种算法。它的方法是在图像中找到基准图像的每一特征点的唯一匹配点, 通过两帧间点位置的变化求出其位移矢量, 再将所有运动信息带入运动模型求出背景的全局运动参数。以基准图像的特征点为中心对匹配图像中使用三步搜索算法<sup>[3]</sup>进行快速搜索。三步搜索算法的具体步骤为:

(1) 以当前特征点为中心, 以4 (可以更改) 为步幅, 将各单位正方形上的顶点 (共9个点) 与当前特征点进行匹配, 求出最佳匹配的特征点中心位置。

(2) 以第一步求出的最佳特征点为中心, 以2为步幅, 将各单位正方形上的顶点 (共9个点) 与当前特征点进行匹配, 求出最佳匹配的特征点中心位置。

**基金项目:** 2013年度常熟理工学院大学生实践创新训练计划项目 (No: 54)。

**作者简介:** 黄天琪 (1993-), 女, 江苏阜宁人, 2011级本科在读, 研究方向: 软件开发; 陈冬琴 (1991-), 女, 江苏高邮人, 2011级本科在读, 研究方向: 软件开发; 殷旭东 (1970-), 男, 江苏张家港人, 工程师, 硕士, 研究方向: 移动计算、软件测试。

## • 设计分析

(3)以第二步求出的最佳特征点为中心,以1为步幅,将各单位正方形上的顶点(共9个点)与当前特征点进行匹配,求出最佳匹配的特征点中心位置。它与当前特征点中心的位置偏移量即为估计的位移量。

### 2.3 运动补偿

这一步的主要目的是尽量将由于摄像机的运动而引起的背景移动消除。这里不需要估计所有像素点的运动矢量。在上面特征点匹配的基础上,采用仿射变换模型参数。设t时刻特征点的位置为(X,Y),t+1时刻的坐标为(X',Y')。若以二维仿射变换描述其运动,则关系可表示为:

$$\begin{pmatrix} X' \\ Y' \end{pmatrix} = \begin{pmatrix} d_0 & d_1 \\ d_3 & d_4 \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} + \begin{pmatrix} d_2 \\ d_5 \end{pmatrix} \quad (1)$$

由式(1)可知三个不共线的特征点就可以求得唯一解,显然这是不合理的。所以我们采用最小二乘法,用所有已知特征点来求解参数,这样可以减小误差。反向运动补偿后当前的帧图像和参考的帧图像重合这样就可以消除因摄像机扫描晃动而引起的背景移动。

### 2.4 目标检测

将背景图像的帧差序列进行帧差运算即可获得运动目标区域,完成目标检测。

## 3 OpenCV实现

### 3.1 概述

OpenCV是一个开源的计算机视觉处理库。它由一系列C函数和C++类构成,独立于操作系统、硬件和图形管理器,具有通用的图像与视频载入、保存和获取模块,具有底层和高层的应用开发包。功能涵盖图像处理,计算机视觉库和模式识别等领域<sup>[4]</sup>。

OpenCV常用的图像函数:

分配图像: `IplImage* cvCreateImage(CvSize size, int depth, int channels);`

其中: depth为像素深度, channels为每个像素值的通道数

释放图像: `void cvReleaseImage(IplImage**);`

读取图像: `img=cvLoadImage(filename);` //filename为文件的目录名

OpenCV常用的基础结构体包括:

(1) CvPoint: 定义整型坐标上的点:

```
typedef struct {
    int x;    //x坐标
    int y;    //y坐标
} CvPoint;
```

(2) CvSize: 定义矩形框大小:

```
typedef struct {
    int width;
    int height;
} CvSize;
```

还有其他基础结构体如: CvRect, CvScalar, CvMat, CvArr, IplImage等限于篇幅不再列出。

## 3.2 实现过程

### 3.2.1 灰度转换

灰度连接或灰度化就是将一张三通道RGB颜色的图像变成单通道灰度图,为接下来的图像处理做准备。在RGB模型中,如果R=G=B时,则彩色表示一种灰度颜色,其中R=G=B的值叫灰度值。

首先使用cvCvtColor函数转换原始图片至灰度图:

`cvCvtColor(img,pImg,CV_BGR2GRAY);`

其中img为加载的源图像,pImg是转换以后的灰度图。

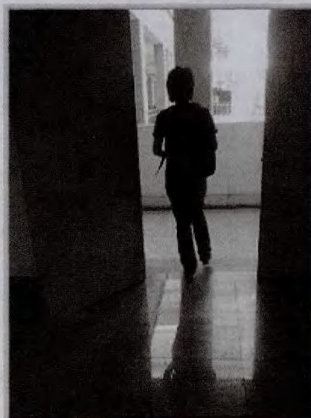


图2 灰度图

### 3.2.2 二值化处理

cvThreshold函数可把图像转换为自适应二值图。二值化是将图像上的像素点的灰度值设置为0或255,也就是使整个图像呈现出明显的黑白效果。二值化的处理强化了锐利的边缘,进一步去除图像中无用的信息,但在使用过程中要注意阈值的选取。

`cvSmooth(pMat,pMat,CV_GAUSSIAN,3,0,0);` // 高斯滤波,先平滑图像,pMat为当前帧。

`cvThreshold(pMat,pImg,60,255,0,CV_THRESH_BINARY);`



图3 二值化图

### 3.2.3 形态学处理

接下来就是进行一系列腐蚀膨胀处理,在阈值分割后,图像中可能还存在一些空洞噪声影响,形态学处理进行进一步去噪。膨胀腐蚀主要是通过cvDilate和cvErode这两个函数来实现效果。

(下转第159页)

(5)调测组作好监视工作及应急调度准备后,通知接续组开始进行旧光缆操作。

(6)接续组接到通知后,打开接头盒首先对接头盒中纤芯进行绕模,待机房监测人员确认纤芯正确时再进行下一步操作。

(7)接续组进行旧光缆开剥、新旧光缆接续;调测组做好监测工作,同时进行已接通光缆测试、业务尾纤连接,联系网管确认割接业务恢复正常。

(8)接续完成,接头盒封装后接续组通知调测组,调测组对空余纤芯进行测试,同时联系机房网管观察已恢复设备运行情况,经确认无问题后,通知割接小组。

(9)清理现场,后续工作。

### 3.3 应急方案

(1)由于采用带业务割接,具有一定的风险,为了避免在光缆开剥和接续过程中出现意外,造成电路中断,在割接前,割接相关部门必须共同制定详细的割接应急方案。

(2)在整个接续过程中,都要注意做好现场保护工作,现场的割接人员仔细认真。。

(3)在带业务割接过程中,如若没有其它资源可作调度,如果在割接过程中发生误操作,引起纤芯中断,必须快速的将断纤重新接通。因此在割接点应备有光纤接续子若干,一旦发生断纤情况,在机房指挥下,立即利用接续子将断纤接回。同时机房利用制定好的应急调度方案,迅速将系统抢通。

(4)为防止同时有多根纤芯发生意外中断时,能准确对应接回断纤,遇有在原有接头进行割接的,现场接头组必须做好对原光缆接头盒内纤芯的核对、标记工作。

## 4 经验总结

### 4.1 特别注意所割接业务经过多个中间跳接电路

割接前注意各业务情况一定要查清楚,特别在是一些中间经过许多跳接站的业务时应注意,要对完成割接后损耗增加情况计算清楚。假如完成割接后增加的中间接头过多、或又增加了跳接点,这样的话及其容易出现由于割接中间增加过多损耗超过设备接收灵敏度,使业务无法开通情况。

### 4.2 与设备厂家的协调工作

割接前一定要与割接涉及设备厂家取得联系,这样方便判断故障产生原因。因为在割接过程会出现各种状况、而割接的最终目的是为了业务恢复正常,在有些替换设备割接工作中,往往是应为设备数据问题或板件故障原因出现业务无法开启。这样在第一时间判断出的话会节省很多在线路端测试、查找故障时间。

### 4.3 接头处替换原光缆时注意光纤色谱对应关系

在原接头替换割接时,打开接头盒后一定要仔细观察原接头接续色谱情况。防止原有光缆未按色谱接续,与新光缆对接时出现错接现象。

## 5 结束语

最后、希望大家在通信光缆割接中不断总结经验,已最短的时间和代价完成割接工作,减少各运营商损失,降低割接对用户的影响,提升企业的服务品牌。

### 【参考文献】

- [1]冯璞.《通信光缆线路施工与维护》.人民邮电出版社.2008年4月.
- [2]卜爱琴.《光缆线路施工与维护》.北京师范大学出版社.2012年2月.

(上接第152页)

```
cvErode(pImg,pImg,0,1); //膨胀函数
cvDilate (pImg,pImg,0,1); //腐蚀函数
最后用OpenCV处理出的结果如图4所示。
```



图4 检测结果图

## 4 结论

本文采用了特征点匹配算法做运动估计,该方法能够迅速且高可靠性地对图像进行匹配。采用了基于特征点匹配基础上的仿射变换模型参数和最小二乘法,用所有已知特征点来求解参数。该方法有效地减少了误差,进行反向运动补偿后可以消除因运动而引起的背景移动。采用了帧差法获得运动目标区域,完成目标检测。在OpenCV开发平台下实现了算法,在运动平台上的摄像头拍摄的图像中可以有效地检测出运动的目标,且具有较好的检测性能。

### 【参考文献】

- [1]郭宝龙,孙伟.数字图像处理系统工程导论[M].西安:西安电子科技大学出版社,2012:121-126.
- [2]谷口庆治.数字图像处理(应用篇)[M].北京:科学出版社共立出版,2002:291-297.
- [3]朱秀昌,刘峰,胡栋.数字图像处理与图像通信(修订版)[M].北京:北京邮电大学出版社,2008:298-300.
- [4]刘瑞桢,于仕琪.OpenCV教程(基础篇)[M].北京:北京航空航天大学出版社,2007:1-90.