

参赛密码 \_\_\_\_\_  
(由组委会填写)



## “华为杯”第十四届中国研究生 数学建模竞赛

题 目                      基于 Vibe 算法的视频前景目标提取

---

### 摘                      要：

前景目标提取是监控视频领域的关键性技术。本文主要通过改良和优化 Vibe 算法，解决相关问题。

**问题一：**当监控视频中背景为静态时，背景产生的扰动较小，提取前景目标相对容易。本文采用 Vibe 算法建立背景模型，其核心思想是在像素点附近邻域内进行随机采样并不断更新，背景模型中的每个像素都由  $N$  个背景样本构成。针对不同类型的视频，本文将该算法的参数进行了优化，并将原始的 8 邻域取样改为 24 邻域取样，在保持算法简单高效的同时，使其鲁棒性大大提高。应用经典的背景差分法得到前景目标，加入中值滤波算法对提取结果进行优化。为了定量对提取结果进行客观分析，引入背景正确率、前景正确率以及总体正确率作为评价指标。文中以 hall 视频和 pedestrian 视频为例进行模型求解，提取结果良好，整体正确率分别为 93.51% 和 97.20%。

**问题二：**当视频包含动态背景时，背景的轻微扰动就会对提取结果造成较大影响。为了去除动态背景的干扰，在第一问改良的 Vibe 算法基础上进一步优化，引入形态学的腐蚀和膨胀算法，使得提取的前景目标形态更为精确，并采用连通域改进，即丢弃小连通域、填补大连通域，解决动态背景的干扰问题。为了叙述

的方便，后文中将这种改良后的 Vibe 算法简称为 iVibe(improved-Vibe)算法。文中以 waterSurface 视频为例进行模型求解，总体正确率为 97.21%。

**问题三：**当摄像头发生偏移或晃动时，视频图像随之发生晃动，为前景目标提取带来一定困难。这种晃动在一定时间内可以近似视为旋转和平移。为了解决这一问题，首先提取图像中的特征角点并进行帧间特征角点的匹配，根据匹配结果求解仿射变换所需的参数。再以视频第一帧为基准，应用仿射变换对每一帧图像进行校正，调整至同一角度。此时校正后的图片会产生黑边，研究中采取逐张填补、实时更新的策略，对存在黑边的图像在迭代中不断完善，即选取第一帧当中的像素点填补第二帧的黑边，再选取填补后的第二帧图像的像素点填补第三帧，以此类推。经过校正和填补的视频即可应用 iVibe 算法进行前景目标的提取。文中以 car6 视频为例进行模型求解，总体正确率为 99.22%。

**问题四：**为了准确地检测出包含显著前景目标的视频帧标号，应有效地提取前景目标。经过前三问的研究与改进，iVibe 算法结合对于视频的校正，已经能较好地适用于各类场景，鲁棒性及实时性都比较理想。首先利用 iVibe 算法及背景差分法得到 8 个待处理视频的前景目标提取结果。为了对显著前景目标出现的视频帧标号进行筛选，提出一种简单高效的比例分析算法：计算第  $k$  帧前景图像像素  $S_k$  占图像像素总数的比例，并与阈值进行比较，判别当前帧是否存在显著前景目标。如若提取结果中存在噪点，将每帧的前景像素总数减去噪点个数，再计算前景占画面比例。进行修正后，即使前景提取结果中存在一定数量的噪点，也能够得到较为可靠的结果，鲁棒性较强。比例分析算法思路清晰、简单易行，不需要大量的计算就可以得到较优的结果。且通过前景目标占比进行判别，不依赖于背景特征或前景特征，可以适用与多种类型的视频前景目标检测。其中 office 视频总体预测正确率可达 99.54%。

**问题五：**对于给定的多角度同场景同时刻的一组视频，首先对视频进行预处理，得到几组图像序列。由于视频处于同场景同时刻，对应时刻的图像中很可能存在相同的前景目标，同时也会存在相同的背景。为了减少运算量并消除背景对于特征点提取的干扰，首先应用 iVibe 算法及背景差分法去除背景、并返回前景目标的原始灰度值，再应用 SIFT 算法提取并匹配这些前景目标上的特征点，根据匹配结果判断出哪些前景目标是同一目标并进行捕捉。该方法充分利用多角度监控视频所提供的具有相关性的有效信息，并丢弃无用的干扰信息，在不同角度的视频中识别出同一目标，可以为监控系统提供有价值的参考。

**问题六：**本文以楼房坍塌或爆炸及人群的异常行为为研究对象，设计异常事件的检测方案：首先利用 iVibe 算法及背景差分法得到前景目标，再从前景目标像素点数量以及前景目标特征点的变化情况入手，深入挖掘不同情况的变化特征。对于楼房坍塌或爆炸等异常事件，其变化主要在于像素点的突然增多，通过统计每一帧的前景目标像素点数量并进行相邻帧之间的比对，分析得出开始发生爆炸的视频帧标号。对于人群的异常行为，采用基于粒子熵值的检测算法：沿用 SIFT 算法进行特征点识别和匹配，得到相邻帧间特征点的速度以及粒子群熵值。人群分布集中时，粒子熵值较小；当人群四处逃散，粒子速度会明显加快，分布更为均匀，导致粒子熵值增大。将前  $l$  帧的熵值以及粒子平均速度与给定的阈值进行比较，得出异常行为发生的视频帧标号。

**关键词：**Vibe 算法；形态学；仿射变换；特征提取；粒子熵值

# 目录

1. 问题的提出 .....	6
1.1 研究背景 .....	6
1.2 研究问题 .....	6
2. 模型假设 .....	7
3. 关键性符号说明 .....	7
4. 包含静态背景视频的前景目标提取 .....	7
4.1 模型建立 .....	7
4.1.1 背景差分法 .....	7
4.1.2 Vibe 算法 .....	8
4.2 模型改进 .....	9
4.2.1 中值滤波 .....	9
4.2.2 参数和邻域改进 .....	9
4.3 实验结果及评价 .....	9
4.3.1 评价指标 .....	9
4.3.2 结果及评价 .....	10
5. 包含动态背景视频的前景目标提取 .....	11
5.1 模型建立及改进 .....	11
5.1.1 腐蚀和膨胀 .....	11
5.1.2 连通域改进 .....	12
5.1.3 iVibe 算法 .....	12
5.2 实验结果及评价 .....	12
6. 包含短暂干扰视频的前景目标提取 .....	13
6.1 模型建立 .....	13
6.1.1 角点的提取 .....	13
6.1.2 仿射变换 .....	14
6.1.3 图像填充 .....	15
6.2 实验结果及评价 .....	15
7. 包含显著前景目标的视频帧标号筛选 .....	16
7.1 模型建立 .....	16
7.2 实验结果 .....	17
7.3 结果检验及评价 .....	17
8. 多角度监控视频的信息获取 .....	19
8.1 模型建立 .....	19
8.1.1 SIFT 算法 .....	19
8.1.2 算法流程 .....	20
8.2 实验结果 .....	20
9. 异常事件的检测方案 .....	21
9.1 模型建立 .....	21
9.1.1 对于爆炸的检测 .....	21
9.1.2 对于人群异常行为的检测 .....	21
9.2 实验结果 .....	22
9.2.1 爆炸检测结果 .....	22

9.2.2 人群异常行为检测结果.....	23
10. 模型评价与改进.....	24
10.1 模型优点.....	24
10.2 模型不足.....	24
10.3 模型改进.....	24
10.4 模型推广.....	25
参考文献.....	25
附录.....	26
附录 1 评级指标算法代码.....	26
附录 2 iVibe 算法代码.....	27
附录 3 仿射变换及图像填补算法代码.....	29
附录 4 比例分析法算法代码.....	30
附录 5 SIFT 算法代码.....	31
附录 6 异常行为检测算法代码.....	39
附录 6-1 爆炸检测算法代码.....	39
附录 6-2 人群异常检测算法代码.....	40

## 1. 问题的提出

### 1.1 研究背景

随着“平安城市”、“金盾工程”、“天网工程”建设的不断完善以及“科技强警”理念的逐渐深入,视频监控已经成为中国安防产业当中相当重要的信息获取手段之一。全国各地广泛安装监控摄像,并利用监控视频的信息,解决安防领域存在的相应问题。

目前,视频前景目标的快速提取,已经成为智能视频监控、智能交通、机器人导航等领域的关键性技术。应用这一技术对监控视频进行处理,能够提供高效的辅助,快速筛选出有效信息。本文主要针对由固定位置摄像头拍摄的监控视频,建立并求解数学模型,进行前景目标的自动提取。

### 1.2 研究问题

**问题 1:** 对于一个不包含动态背景、且摄像头稳定拍摄的时长 5 秒的监控视频,构建数学模型,提取视频中的前景目标(如人、车、动物等)。针对该模型设计设计高效求解方案,使得提取结果能够达到类似图 1-1 的效果。



图 1-1 左图: 原视频帧; 右图: 分离出的前景目标

**问题 2:** 当监控视频中包含动态背景(如树叶晃动、喷泉变化、水波动、窗帘摇动)时,建立模型,设计有效的前景目标提取方法。

**问题 3:** 监控过程中,摄像头有时会发生偏移或晃动,导致视频发生短暂抖动(此类视频变换在短时间内可将其近似视为线性仿射变换)。本文进一步针对该种视频,尝试建立有效模型提取前景目标。

**问题 4:** 将前三问建立的模型应用到 8 组视频当中,筛选出每组视频包含显著前景目标的视频帧标号。

**问题 5:** 充分考虑并利用多角度视频的前景(或背景)之间的相关信息,尝试建立模型,从不同角度同时拍摄的近似同一地点的多个监控视频中(如图 1-2)高效地检测并提取视频前景目标。



图 1-2 在室内同一时间从不同角度拍摄同一地点获得的视频帧

**问题 6:** 充分思考并发掘视频给出的信息,判断监控视频中是否有人群短时聚集、人群惊慌逃散、群体规律性变化、物体爆炸、建筑倒塌等异常事件,并对更多类型的异常事件建立相应的检测方案。

## 2. 模型假设

- (1) 假设动态背景中背景的扰动幅度较小；
- (2) 假设当镜头晃动时发生的是线性仿射变换。

## 3. 关键性符号说明

符号	解释说明
$v(x)$	图像在 $x$ 处的像素值
$N$	样本数量
$v_i$	第 $i$ 个背景样本值( $i=1,2,\cdots,N$ )
$M$	背景模型
$R_k$	二值化后的第 $k$ 帧图像

## 4. 包含静态背景视频的前景目标提取

当监控视频不包含动态背景时背景中的扰动较少，前景目标的提取相对容易。在本章当中，针对视频不包含动态背景的情况，应用 Vibe 算法建立视频图像的背景模型，加入中值滤波算法并对相关参数和邻域选择进行调整，优化实验结果。最后采用背景差分法来获取前景目标。

### 4.1 模型建立

#### 4.1.1 背景差分法

背景差分法是前景检测中较为常用的一种算法。该算法根据灰度等特征的变化来判断，即将当前图像与背景进行差分，当差分所得值大于某一阈值时，则认为检测到目标<sup>[1-3]</sup>。设  $f_k(x,y)$  为第  $k$  帧图像， $B_{k-1}(x,y)$  为背景图像， $D_k(x,y)$  为当前帧图像与背景图像之差，即：

$$D_k(x,y) = f_k(x,y) - B(x,y) \quad (4-1)$$

选取  $Th$  为阈值，对差分图像进行二值化：

$$R_k = \begin{cases} 1, & D_k(x,y) \geq Th \\ 0, & D_k(x,y) < Th \end{cases} \quad (4-2)$$

$R_k$  即为进行二值化后的图像。原理流程见图 4-1。

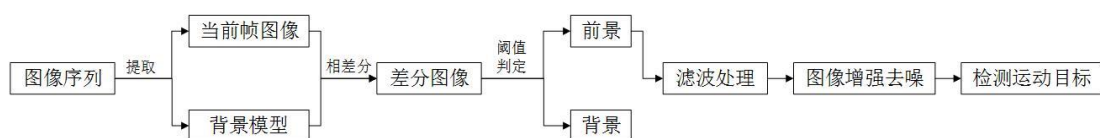


图 4-1 背景差分法原理流程图

由于背景差分法中，背景模型选取的精确与否将直接影响到最终前景目标检测的准确性。因此，选取合适的算法进行背景模型求解就尤为重要。本章主要参考当下较为前沿且高效的 Vibe 算法进行求解，具体见 4.1.2 节。

#### 4.1.2 Vibe 算法

Vibe (visual background extractor) 算法<sup>[4,5]</sup>, 即视觉背景提取子算法。该方法基于样本的随机聚类进行背景建模, 采用视频序列的第一帧进行背景模型的初始化, 实时性和鲁棒性都比较高, 且不需要类似混合高斯分布算法的大量计算, 实用且高效, 本章主要参考该种算法进行背景模型的建立。

在 Vibe 方法中背景模型中的每一个像素都由  $N$  个背景样本组成 (通常  $N$  取 20),  $v(x)$  表示图像在  $x$  处的像素值,  $v_i$  表示第  $i$  个背景样本值。背景模型  $M$  定义为式(4-3):

$$M(x) = \{v_1, v_2, \dots, v_{N-1}, v_N\} \quad (4-3)$$

利用视频序列第一帧创建背景的初始化模型, 图 4-2 表示  $N$  个背景样本中的某一帧, 图 4-3 为第一帧图像, 初始化思想是在图 4-3 中  $x$  位置的 8 邻域  $N_G(x)$  中, 随机选取一个像素值  $v_G(m)$  ( $m=1,2,3,\dots,8$ ) 来初始化  $v(x)$ , 如式(4-4)所示:

$$M^0(x) = \{v^0(y|y \in N_G(x))\} \quad (4-4)$$

$v(x-1)$	$v(x)$	$v(x+1)$

图 4-2 背景模型中某一帧

$v_G(1)$	$v_G(2)$	$v_G(3)$
$v_G(4)$	$v_G(x)$	$v_G(5)$
$v_G(6)$	$v_G(7)$	$v_G(8)$

图 4-3 视频序列第一帧

Vibe 算法采用 2D 空间中的欧氏距离对像素进行分类, 如图 4-4 所示:

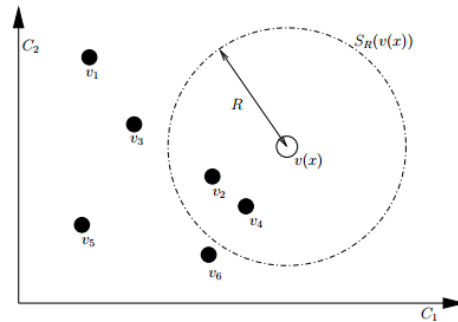


图 4-4 欧式空间中的像素分类

图中  $S_R(v(x))$  表示以  $v(x)$  为中心，半径为  $R$  的欧式空间。若  $S_R(v(x))$  与  $M(x)$  的交集个数  $\# \{S_R(v(x)) \cap \{v_1, v_2, \dots, v_N\}\}$  大于等于某一阈值  $\#_{\min}$ ，则认为  $v(x)$  为背景像素。

如果判断像素  $v(x)$  为背景像素，此时用该点的像素值替代  $M(x)$  中随机选取的某一个值。同时为了保证像素邻域的一致性，也用其替代  $x$  位置的 8 邻域  $N_G(x)$  中某一像素的样本，保持了信息的传播。

由于样本数量  $N$  以及阈值  $\#_{\min}$  对于样本选取和更新有决定性的影响，因此模型的灵敏度可以由  $\#_{\min}/N$  表示。

## 4.2 模型改进

### 4.2.1 中值滤波

SMF (standard median filter) 算法<sup>[6]</sup>，即标准中值滤波算法，是一种应用快速排序算法对噪声进行有效抑制的信号处理技术。当监控视频的背景发生轻微扰动时，未改进的前景提取算法很有可能将扰动的背景误判为前景，使提取的前景目标中产生噪声。因此采用中值滤波算法对改良 Vibe 算法进行完善，得到更理想的实验结果。

中值滤波的基本原理是把数字图像中一点的值，用该点邻域内各点值的中值替换。即在定义窗口范围内对样本取中值，如式(4-5)所示<sup>[7]</sup>：

$$Y(n) = \text{med} [v(n-N) \cdots v(n) \cdots v(n+N)] \quad (4-5)$$

其中  $\text{med}[]$  表示窗口内所有数按升序排列并取中值（窗口通常取  $3 \times 3$  像素大小），然后用邻域内的中值代替中心像素。该种算法可以使那些与周围像素灰度值相差较大的像素改取与周围像素更为接近的值，从而消除前景提取结果中孤立的噪声点。

### 4.2.2 参数和邻域改进

原始的 Vibe 算法在初始化时，在 8 邻域当中随机选取 20 个样本。该种样本选取策略过于集中，会导致像素的错误分类。针对这一现象可以采用扩大邻域的方式<sup>[9]</sup>，通过均匀分布的随机数生成算法，从  $v(x)$  的 24 邻域中选取 20 个样本。扩大邻域后可以有效避免重复选取像素的情况，使分类错误的概率大大降低。

通常情况下，在应用 Vibe 算法时，默认取阈值  $\#_{\min}$  为 2， $S_R(v(x))$  的半径  $R$  为 20，样本数量  $N$  为 20。然而在实验过程中我们发现，根据不同种类的视频对这三个参数在一定范围内进行适当的调整，往往可以得到更优的处理结果。在之后的实验结果中将有所体现。

## 4.3 实验结果及评价

### 4.3.1 评价指标

虽然将检测出的前景目标与蒙版相比可以对大体判别出提取效果的优劣，但通过图像对比，我们只能得到定性的分析以及比较主观的判断。为了衡量检测算法的性能，需要在统计意义上对算法进行评价，为此本文引入检测率作为评价指标，对前景目标检测结果进行定量分析。

前景检测其实是个二分类问题，即需要对图像中的每个像素进行判别是前景还是背景。对前景或背景检测效果进行评估的指标列举如下：

- (1) 前景点数量 AF；
- (2) 背景点数量 AB；
- (3) 实际为前景点，检测算法正确检测为前景点的数量 TP；



(4) 实际为背景点，检测算法正确检测为背景点的数量  $TN$ ；。

基于以上统计数据，可以得到下面一些衡量算法性能的更深层次指标：

(1) 背景正确率  $TNR=TN/AB$ ；

(2) 前景正确率  $TPR=TP/AF$ ；

而我们最关注的总体正确率  $TR$  定义如式 (4-6)，评价指标算法代码见附录 1。

$$TR=(TP+TN)/(TP+FN) \quad (4-6)$$

### 4.3.2 结果及评价

应用 Vibe 模型，加入中值滤波、参数改进和邻域改进的处理，选用两个典型静态背景的短视频进行模型检验，实验结果如下。

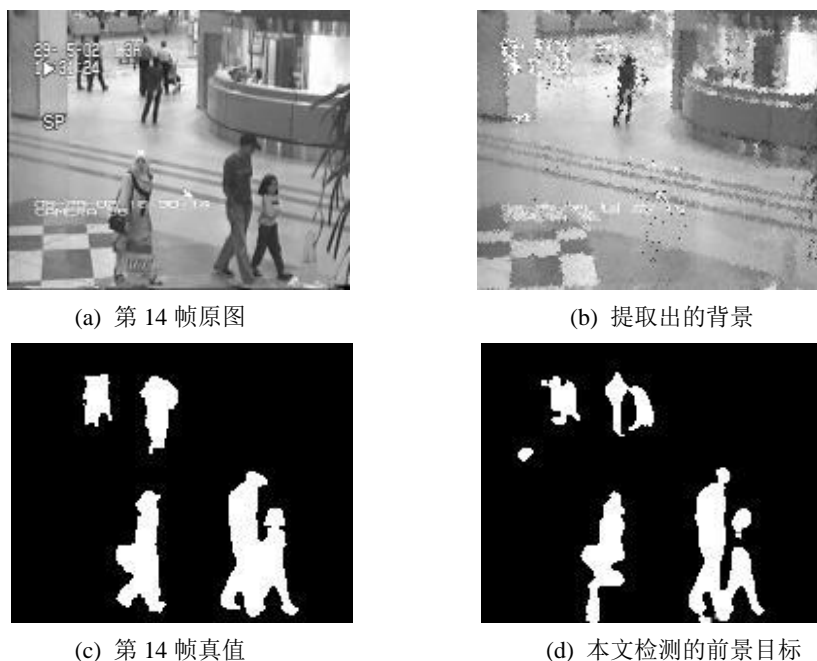


图 4-5 hall 视频的实验结果

由图 4-5 可以看出，本文改良参数并进行膨胀和腐蚀运算后的 Vibe 算法在背景模型的建立和前景目标的提取上相当高效。本文检测出的前景目标比给定的蒙版更为精细且清晰。通过不断地更新样本，最终提取的背景基本上去除了全部的前景目标。

图 4-6 给出了 pedestrian 视频的实验结果，背景处理比较彻底，行人的提取也较为清晰。当前景目标和背景的颜色较为接近时，可能会出现提取不完整的现象，但整体上与蒙版非常吻合，总体上可以达到很好的效果。



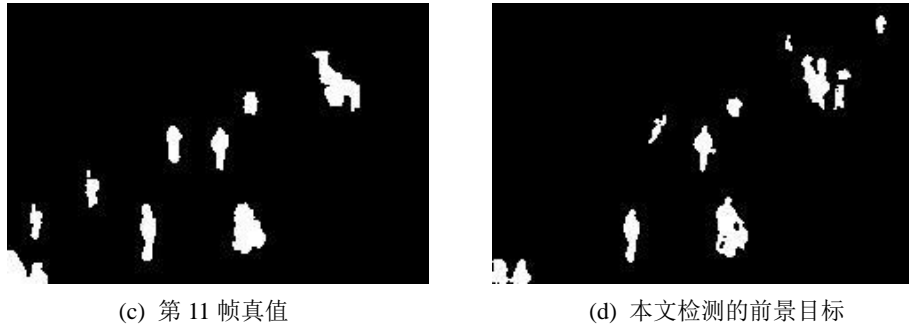


图 4-6 pedestrian 视频的实验结果

由于给定的蒙版与真值仍有一定差距，我们只能人工提取出前景目标的真值并与检测目标进行对比。时间原因，文中对于前景目标提取的结果分析，仅给出视频中某一帧的正确率。由 4.3.1 节定义的评价指标体系，可以得到以上两个视频的正确率指标，如表 4-1 所示：

表 4-1 hall/pedestrian 视频前景目标提取结果评价

评价指标 视频名称	背景正确率 TNR	前景正确率 TPR	总体正确率 TR
hall	0.9722	0.6779	0.9351
pedestrian	0.9897	0.6134	0.9720

## 5. 包含动态背景视频的前景目标提取

在包含动态背景的视频中，背景的扰动（树叶摇动、水波动、窗帘晃动等）对于前景目标的提取会产生一定干扰。因此本章将在第 4 章建立的改良 Vibe 算法的基础上，进一步对算法进行完善，采用腐蚀膨胀算法以及连通域改进的方法，提出最终的改进 Vibe 算法（improved-Vibe，简称为 iVibe），使得背景提取和前景目标检测更为精确可信。

### 5.1 模型建立及改进

第 4 章中建立的改良 Vibe 算法已经比较完整且高效，因此本章总体上继续沿用改良 Vibe 算法（具体算法参见 4.1 节和 4.2 节），并在已有算法中加入腐蚀膨胀算法和连通域改进，得到新的 iVibe 算法，使得动态背景得到较好的处理。

#### 5.1.1 腐蚀和膨胀

膨胀(Dilation)和腐蚀(Erosion)是形态学当中最基础的两种运算<sup>[8]</sup>。

##### (1) 图像的腐蚀

对于  $Z^2$  上元素的集合  $A$  和  $S$ ，使用  $S$  对  $A$  进行腐蚀，其表达式如下：

$$A \ominus S = \{z | (S)_z \subseteq A\} \quad (5-1)$$

算法具体可以总结为三个步骤：

**Step 1** 用结构元素  $S$ （通常是  $3 \times 3$  大小）扫描图像中的每个像素；

**Step 2** 用结构元素  $S$  与其覆盖的二值图像做“与”操作；

**Step 3** 如果都为 1，则结果图像的该像素为 1，反之为 0。

##### (2) 图像的膨胀

对于  $Z^2$  上元素的集合  $A$  和  $S$ ，使用  $S$  对  $A$  进行膨胀，其表达式如下：

$$A \oplus S = \left\{ z \left| \left( \hat{S} \right)_z \cap A \neq \emptyset \right. \right\} \quad (5-2)$$

类似的，算法具体可以总结为如下三个步骤：

**Step 1** 用结构元素  $S$ （通常是  $3 \times 3$  大小）扫描图像中的每个像素；

**Step 2** 用结构元素  $S$  与其覆盖的二值图像做“与”操作；

**Step 3** 如果都为 0，则结果图像的该像素为 0，反之为 1。

### 5.1.2 连通域改进

背景扰动以及光线变化等因素，会导致前景提取后出现一些不该存在的小目标，前景目标还有可能出现部分残缺的现象。为了对此进行改进，可以采用丢弃和修复策略，提升检测系统的精确性<sup>[9]</sup>。该方法的核心步骤如下：

**Step 1** 求取前景图像上的连通域集合  $B_1$

**Step 2** 如果连通域内的像素数小于  $L_{\min}$ ，则认为该区域是背景扰动，丢弃该连通域，得到新的连通域集合  $B_2$

**Step 3** 对新集合  $B_2$  中的连通域进行统计，若某连通域像素数大于  $L_{\min}$  且小于  $L_{\max}$ ，则认为该区域存在漏洞，对其进行填充；

上述步骤中， $L_{\min}$  和  $L_{\max}$  通常取 10 和 60，针对不同情况可以进行微调，使得检测结果最优化。

### 5.1.3 iVibe 算法

在原始 Vibe 算法的基础上对邻域和参数进行改良。应用背景差分法得到前景目标提取结果后，采用中值滤波去除噪音，并应用形态学当中的腐蚀和膨胀算法以及连通域改进对结果进行优化，最终得到经过一系列改良和优化的 iVibe 算法。iVibe 算法核心代码见附录 2。

## 5.2 实验结果及评价

为了验证膨胀腐蚀算法以及连通域改进的有效性，本节采用包含动态背景的 waterSurface 视频进行实验。首先我们不采用中值滤波和连通域改进，直接沿用第 4 章的方法进行前景目标提取，得到如图 5-1 所示的结果。



图 5-1 未优化的实验结果

可以看到由于水的波动，图上有较多应当舍弃的噪点和小连通域，提取效果并不可靠。将膨胀腐蚀算法和连通域改进应用到模型中，得到新的实验结果如图 5-2 所示。

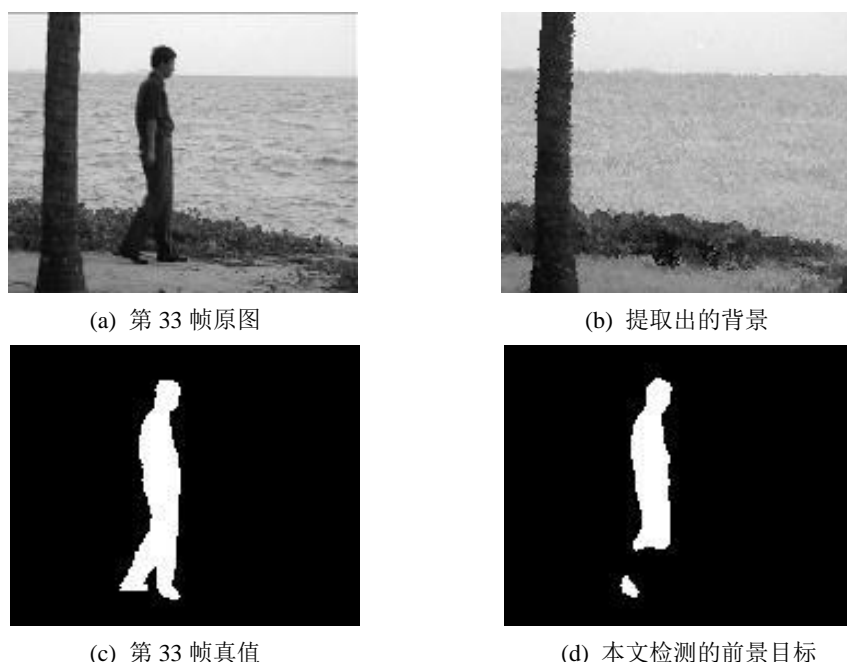


图 5-2 waterSurface 视频的实验结果

与图 5-1 进行比对，改良效果是相当显著的。由此可见本文提出的 iVibe 算法较之间的 Vibe 算法能更好的解决背景扰动问题，在保持实用性和可行性的同时，鲁棒性也更强。沿用 4.3.1 节中引入的评价体系，评价结果如表 5-1：

表 5-1 waterSurface 视频前景目标提取结果评价

评价指标 视频名称	背景正确率 TNR	前景正确率 TPR	总体正确率 TR
waterSurface	0.9944	0.6837	0.9721

## 6. 包含短暂干扰视频的前景目标提取

当摄像头发生偏移或晃动时，进行前景目标提取的最大难点在于背景的晃动，这种晃动在一定时间内可以近似视为旋转和平移。本章提取图像中的特征角点，应用仿射变换，以视频第一帧为基准，对其余的每一帧进行校正，调整至同一角度。此时校正后的图片会产生黑边，再对存在黑边的图像在迭代中不断填补，得到完整图像。经过校正和填补的视频即可应用 iVibe 算法进行前景目标的提取。

### 6.1 模型建立

#### 6.1.1 角点的提取

角点，即图像当中物体边缘上曲率较大的点<sup>[10-11]</sup>，丰富的角点信息可以提供大量局部特征和形状特征。要实现角点的跟踪，重点在于是角点提取以及角点匹配。角点提取的方法主要有两类，即根据灰度图像数据的检测和基于边缘信息的检测。本文主要采用基于灰度的角点提取。该方法无需边缘提取，且检出的角点可靠性很高。对相邻帧图像进行特征角点匹配，需要识别出每帧图像中的特征角点，再特征角点的定义找到与其对应的特征角点。图 6-1 为视频 people2 第 1 帧和第 24 帧的特征角点提取和匹配结果。

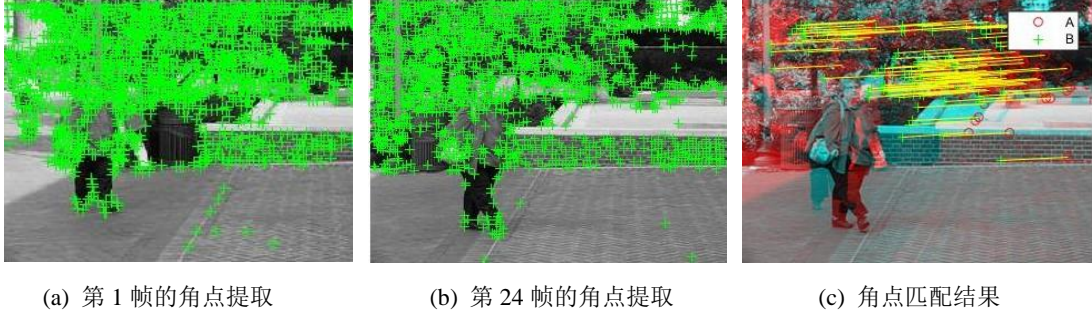


图 6-1 people2 视频的角点提取和匹配

### 6.1.2 仿射变换

仿射变换(affine transform)由一个线性变换接上一个平移组成。其定义为,若存在变换  $S$  如式(6-1)所示<sup>[12]</sup>:

$$S: R^n \rightarrow R^n, S(x) = T(x) + a \quad a \in R^n \quad (6-1)$$

其中  $T$  是非奇异线性变换,则称变换  $S$  为仿射变换。平移、缩放、旋转等是二维仿射变换中的特例。

(1) 平移: 对坐标点进行平移的仿射变换形式为:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (6-2)$$

(2) 旋转: 对坐标点进行旋转的仿射变换形式为:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (6-3)$$

(3) 缩放: 对坐标点进行平移的仿射变换形式为:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \beta \begin{bmatrix} x \\ y \end{bmatrix} \quad (6-4)$$

其中  $\beta$  是缩放的比例因子。

通过对监控视频的预处理,我们可以得到一帧帧的灰度图像,而任一二维灰度图像都可以用矩阵  $T$  来表示。一个图像或者一个图像函数  $f(x,y)$ ,  $x,y=1,2,\dots,N$  都可以看做  $R(N^2)$  空间内一点。 $f(x,y)$  可以认为该点在基组  $\{a_{ij}; i,j=1,2,\dots,N\}$  下对应于基  $a_{ij}$  的坐标。图像变换的结果就是图像函数在新基下的坐标<sup>[13]</sup>。

本文当中,我们以第一帧为基准,对每一帧图像进行匹配。具体的匹配参数根据 6.1.1 节中获得的特征角点匹配结果来进行求解。图 6-2 为仿射变换结果。



图 6-2 people2 视频第 24 帧仿射变换结果

可以看到,经过仿射变换后,背景视角调整到一致的状态,同时人与背景的



特征及其相对位置基本保持不变。然而背景边缘留下较大的黑边，这对于接下来的前景目标提取将产生较大影响。因此，需要将黑边进行填充。

### 6.1.3 图像填充

在进行图像填充时，采用逐张填充、实时更新的策略。首先应用仿射变换将所有帧图片以第一帧为基准进行变换。为了保证背景的实时性和有效性，填补过程逐张迭代，即以第一帧图像为标准填补第二帧图像，再以填补完整的第二帧图像为基准填补第三帧，以此类推。填充结果如图 6-3 所示。



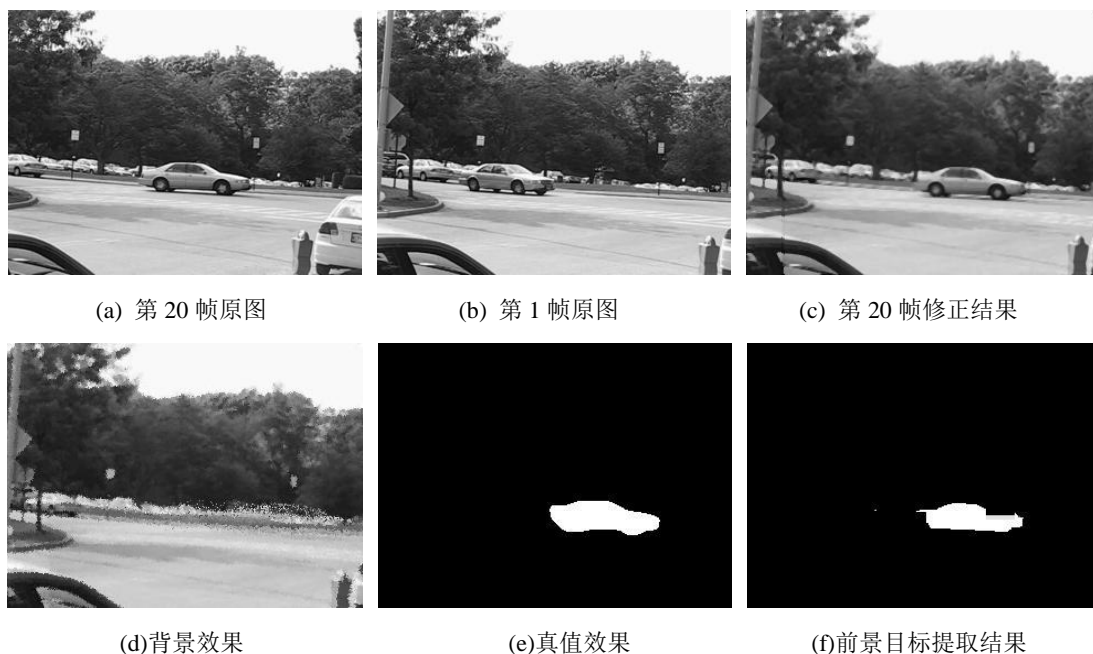
(a) 第 24 帧放射变换结果

(b) 第 24 帧图像填补结果

图 6-3 people2 视频第 24 帧图像填充结果

## 6.2 实验结果及评价

对所有帧图像进行修正之后，可以得到一组视角一致的视频。此时再应用 iViBe 算法提取该视频的前景目标可以得到比较可信的结果。图 6-4 是对视频 car6 视频进行修正和提取的结果。放射变换及图像填补算法代码见附录 3。



(a) 第 20 帧原图

(b) 第 1 帧原图

(c) 第 20 帧修正结果

(d)背景效果

(e)真值效果

(f)前景目标提取结果

图 6-4 car6 视频修正及目标提取结果

利用 4.3.1 节中给出的指标体系，对提取结果进行定量的评价，结果如表 6-1：

表 6-1 car6 视频前景目标提取结果评价

评价指标 视频名称	背景正确率 TNR	前景正确率 TPR	总体正确率 TR
car6	0.9998	0.7292	0.9922

## 7. 包含显著前景目标的视频帧标号筛选

经过第 4、5、6 章的研究与改进后，iVibe 算法已经能较好地适用于各类场景，鲁棒性及实时性都比较理想。在本章当中，首先利用 iVibe 算法得到 8 个视频的前景目标检测结果。为了对显著前景目标出现的视频帧帧号进行筛选，提出一种简单却高效的比例分析算法，通过统计每一帧的前景目标像素点数量并与阈值进行比较，判别当前帧是否存在前景目标。

### 7.1 模型建立

在应用背景差分法时，我们已经对差分图像进行了二值化处理即根据给定的阈值筛选出前景和背景（具体方法步骤见 4.1.1 节）。二值化之后的图像，前景点赋值为 1，背景点为 0。根据二值化图像我们可以得到前景目标在某一帧当中占据的像素总数。在此基础上，本文提出比例分析法，用来筛选包含显著前景目标的视频帧帧号。

比例分析法的基本思想是，根据检测到的前景目标占整个图片的比例来判断该帧当中是否出现了前景目标。算法流程如图 7-1：

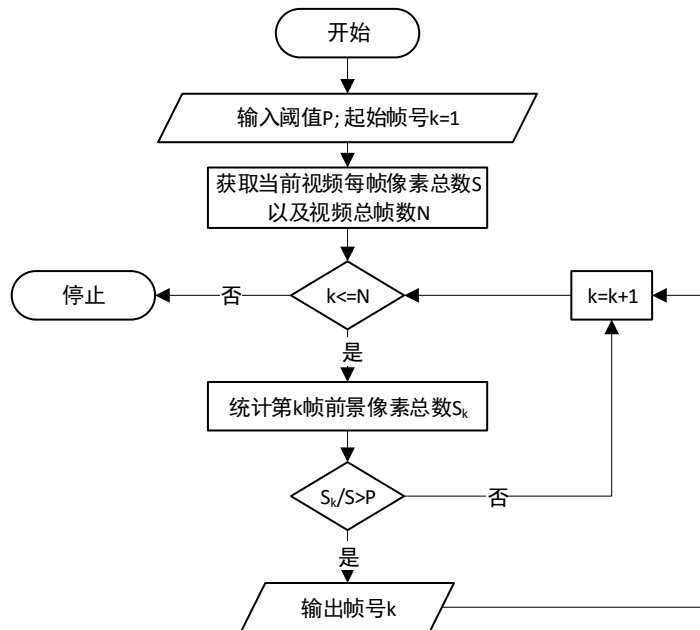


图 7-1 比例分析算法流程图

当前景提取结果中噪点较多时，可能影响比例分析算法的精确性。此时可以将该算法进行改进。如若噪点不可避免地存在于提取结果中，那么多数情况下每一帧的噪点数量  $n$  是比较接近的。此时统计出某一帧的噪点个数，并将每帧的前景像素总数  $S_k$  减去噪点个数，再计算前景占画面比例，即将判定条件修正为：

$$\frac{S_k - n}{S} > P \quad (7-1)$$

进行修正后，即使前景提取结果中存在一定数量的噪点，也能够得到较为可靠的结果，鲁棒性较强。比例分析算法思路清晰、简单易行，不需要大量的计算

就可以得到较优的结果。且通过前景目标占比进行判别，不依赖于背景特征或前景特征，可以适用与多种类型的视频前景目标检测。比例分析算法代码见附录4。

7.2 实验结果

首先应用第5章得到的改进 Vibe 算法对8组视频进行处理，得到每一帧的前景目标提取结果。部分提取效果图如图7-2所示：



(b)hall 视频检测图像 3485-3489 帧



(b)hall 视频实际图像 3485-3489 帧

图 7-2 hall 视频前景提取结果

应用7.1节提出的比例分析算法，对处理后的8组图片进行判别，表7-1列出8组视频的判别结果：

表 7-1 带有前景目标的视频帧标号检测结果

视频名称	检测结果
Campus	85, 204-217, 313-504, 600, 647-682, 694-711, 749-872, 1009-1033, 1193, 1264, 1334-1400
curtain	411, 967, 1561, 1762-1900, 2126, 2189-2298, 2642, 2781-2912
escalator	1-131, 233-2378, 2539, 2754, 2787-2953, 3026-3417
fountain	1-93, 141, 160-209, 259, 410-523
hall	1-66, 80-139, 151-328, 366-443, 578, 648-696, 795, 823-838, 890-1048, 1138, 1156-1198, 1246, 1305-3315, 3418-3534
lobby	161-191, 259, 346-391, 521, 625-665, 870, 969-1030, 1161, 1235-1277, 1341-1542
office	197, 372, 501, 590-2032, 2080
overpass	374, 570-641, 968, 1551, 1881, 2343-2795

由检测结果中可以看到，某些视频当中的某一帧（如 curtain 第411、967帧等）会突然出现前景目标，与前后并无过渡。当出现这样的图像时，就有必要重点关注该帧附近的视频段落，查看是否有异常情况发生。

7.3 结果检验及评价

通过将原视频分解为一帧帧的图像再进行人工检查，可以得到精确的前景目标出现的视频帧标号。与前景检测相类似的，对于包含显著前景目标的视频帧标号检测，也属于一个二分类问题，需要对图像中是否存在前景目标进行判别。因此可以参考并沿用4.3.1节中引入的评价指标，对本章的检测结果进行评价。真



实结果如表 7-2 所示。

表 7-2 带有显著前景目标的实际视频帧标号

视频名称	前景目标存在帧标号
Campus	85, 200-224, 307-522, 600, 644-682, 692-712, 737-875, 1006-1035, 1193, 1264, 1330-1374, 1377-1405
curtain	411, 967, 1561, 1758-1902, 2126, 2175-2314, 2642, 2769-2930
escalator	1-140, 228-2392, 2415, 2539, 2754, 2775-2970, 2991-3417
fountain	1-95, 141, 157-210, 259, 409-523
hall	1-68, 79-140, 160-334, 352-459, 578, 637-696, 795, 819-846, 874-1050, 1138, 1154-1198, 1246, 1279-1506, 1516-1704, 1739-2581, 2719-2782, 2815-2969, 3026-3316, 3415-3534
lobby	154-198, 259, 345-393, 521, 623-668, 870, 963-1037, 1161, 1239-1283, 1334-1538
office	197, 372, 501, 583-2040, 2080
overpass	374, 551-666, 968, 1551, 1881, 2098, 2335-2840

评价指标如表 7-3 所示：

表 7-3 检测结果评价指标

评价指标 视频名称	不存在前景正确率 TNR	存在前景正确率 TPR	总体正确率 TR
Campus	0.9978	0.8723	0.9500
curtain	1	0.8540	0.9777
escalator	1	0.9683	0.9728
fountain	1	0.9737	0.9866
hall	0.6979	0.9568	0.8896
lobby	0.9963	0.8977	0.9662
office	1	0.9927	0.9954
overpass	1	0.8437	0.9673

参考评价指标计算结果，可以认为该方法对于监控视频是否存在显著前景目标的检测十分精确。通过查看原视频的每一帧图像，可以发现的确有某些帧会出现异常，例如忽然出现一个人影又立即消失，或者忽然出现一个车身又马上消失等，如图 7-3:



(a) curtain 实际图像 965-969 帧



(b)curtain 检测图像 965-969 帧

图 7-3 异常帧图像

这可能是视频错误导致，也可能是车速过快等等。不论是那种情况都需要引起关注。这些出现异常的视频帧标号，与我们在 7.2 节中预测的异常帧基本上符合。可见比例分析算法的预测结果有效，可以为监控系统提供参考信息。

## 8. 多角度监控视频的信息获取

基于前文的 iVibe 算法以及对抖动视频的校正，我们已经可以在多数情况下较好地提取前景目标。本章应用基于 SIFT 算法的图像目标匹配和定位，充分利用多角度监控视频所提供的信息，在不同角度的视频中识别出同一目标，为监控系统提供极有价值的参考。

### 8.1 模型建立

#### 8.1.1 SIFT 算法

SIFT(scale invariant feature transform)特征算子<sup>[14]</sup>，是一种具有尺度、旋转、仿射、视角、光照不变性的比较稳定的特征算子。SFIT 算子在前景目标进行运动、被遮挡或者存在噪声等情况下也能保证较好的匹配。

该方法的主要步骤为：

**Step 1** 将二维图像  $I(x, y)$  与高斯核  $G(x, y, \sigma)$  进行卷积，可以得到该二维图像在不同尺度下的尺度空间表示  $L(x, y, \sigma)$ 。在 DOG 尺度空间中的 26 个领域中检测极值， $D(x, y, \sigma)$  是两个相邻尺度图像之差，即：

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (8-1)$$

式中  $G(x, y, \sigma)$  为二维高斯函数， $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$  是尺度空间的定义。若一个点与 8 邻域以及上下 18 个邻域相比是最大值或最小值，则认为该点是图像的一个特征点。

**Step 2** 利用特征点邻域内的梯度分布特性，给出特征点指定方向参数，使算子具备旋转稳定性。

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (8-2)$$

$$\theta(x, y) = \arctan \{ [L(x, y+1) - L(x, y-1)] / [L(x+1, y) - L(x-1, y)] \} \quad (8-3)$$

其中  $m(x, y)$  是  $(x, y)$  处梯度的模值， $\theta(x, y)$  是该点处的梯度方向。

**Step 3** 生成 SIFT 特征向量，将坐标轴旋转到特征点方向，保证旋转稳定性。

**Step 4** 找出图像特征点后，根据特征点之间的对应关系进行匹配。SIFT 特征向量进行匹配时主要考量图像中特征向量的相似度，使用斜面距离作为相似性度量的标准。

### 8.1.2 算法流程

对于给定的多角度同场景同时刻的一组视频，首先对视频进行预处理，得到几组图像序列。由于视频处于同场景同时刻，因此对应时刻的图像中很可能存在相同的前景目标，但同时也会存在相同的背景。为了减少运算量并消除背景对于特征点提取的干扰，首先应用 iVibe 去除背景、并返回前景目标的原始灰度值，再应用 SIFT 算法提取并匹配这些前景目标上的特征点，根据匹配结果判断出那些前景目标是同一目标并进行捕捉。算法流程如图 8-1。SIFT 算法代码见附录 5。

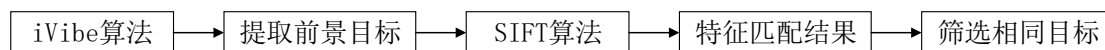


图 8-1 SIFT 捕捉相同前景目标

### 8.2 实验结果

在某研究室的官方网站上我们下载了一组名为 4p 的视频，原始视频有 4 个角度，本文以两个角度为例，很容易推广至多角度的目标识别（视频下载地址为 <http://cvlab.epfl.ch/research/surv/multi-people-tracking>）。

首先应用 iVibe 算法滤掉不需要识别的背景图像，留下前景目标，如图 8-2：

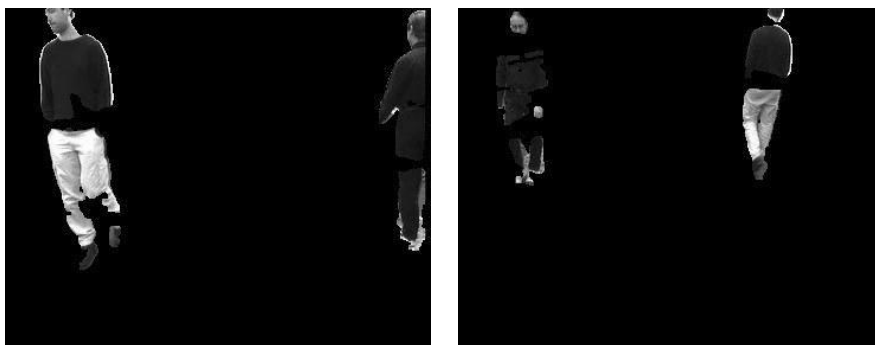


图 8-2 4p-c0 和 4p-c1 的第 521 帧前景目标提取结果

去除背景后，应用 SIFT 算法识别前景目标的特征算子，如图 8-3：

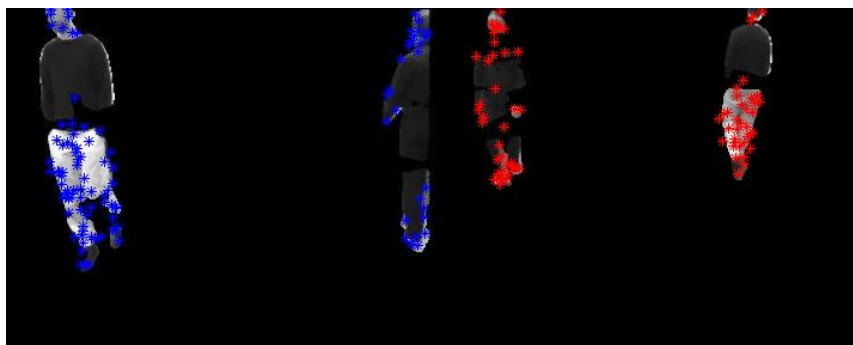


图 8-3 4p-c0 和 4p-c1 的第 521 帧前景目标特征点识别

将两个角度的特征点进行匹配，根据匹配结果判别出相同的前景目标。

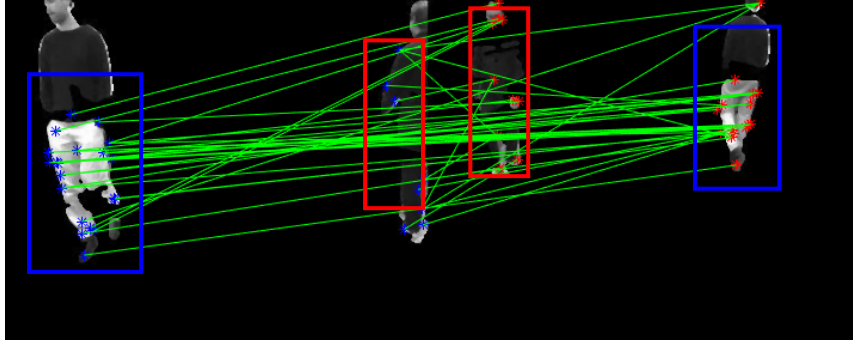


图 8-4 4p-c0 和 4p-c1 的第 521 帧前景目标特征点匹配

由图 8-4 可以看出，特征点匹配结果准确率较高。即使有少量点发生匹配错误，也不会影响整体上对前景目标的匹配。

## 9. 异常事件的检测方案

视频监控的理想情况是能对异常事件（如爆炸、人群聚集或逃散等）进行及时的反馈，甚至预警。本章基于第 4-8 章给出的 iVibe 前景提取算法、基于仿射变换的视频修正、前景目标视频帧标号检测以及 SIFT 算法等方法，充分利用可以从视频中提取出的信息对异常事件进行检测。

### 9.1 模型建立

#### 9.1.1 对于爆炸的检测

当爆炸、楼梯倒塌等异常事件发生时，视频图像中的主要变化在于前景像素点的忽然增多。在对这类事件进行预警时，我们首先应用 iVibe 去除背景影响，再将视频转换为一系列二值图像，使得计算量大大减少。统计每一帧的前景像素数量  $S_k$ ，并计算与前一帧像素总数  $S_{k-1}$  的差值  $C_k$ ，即：

$$C_k = S_k - S_{k-1}, \quad k=2,3,\dots,M \quad (9-1)$$

其中  $M$  表示视频帧总数。当差值  $C_k$  大于某一阈值时，认为此时有爆炸或楼体倒塌等情况的发生。爆炸检测代码见附录 6-1。

#### 9.1.2 对于人群异常行为的检测

人群异常行为的检测可以为公共安全提供可靠的信息支持。由于人群聚集或逃散时，主要是速度和分布密度的快速变化，因此采用基于粒子熵值的算法来进行检测<sup>[15]</sup>。

首先采用 iVibe 算法去除背景的干扰，再应用 8.1.1 节中的 SIFT 算法提取前景目标的特征点。检测异常行为的主要思想是，将图像进行网格划分，用  $h_j(t)$  来表示  $t$  时刻第  $j$  个网格中特征点的数量。当人群集中时，特征点主要集中于某个网格区间；当人群逃散时，特征点的速度会骤然加快，分布也更加均匀。这种分布特征，我们一般用熵值来描述。图像中粒子熵值定义为式(9-2)：

$$E_{Ht} = - \sum_{j=1}^m h_j(t) \log(h_j(t)) \quad (9-2)$$

其中  $m$  为网格个数，为了对群体行为进行准确的判断，避免由于某帧的异常而造成误判，该算法综合一段时间内的特征信息作为判别值，即前  $l$  帧图像的粒子熵值，如式(9-3)：

$$E_t = \sum_{j=t-1}^t E_{Ht} \quad (9-3)$$

计算前  $l$  帧特征点的平均速度，当粒子熵值  $E_t$  及平均速度均大于设定的阈值时，判定存在异常行为，否则判定为正常行为。基于粒子熵值的算法流程见图 9-1，核心代码见附录 6-2。

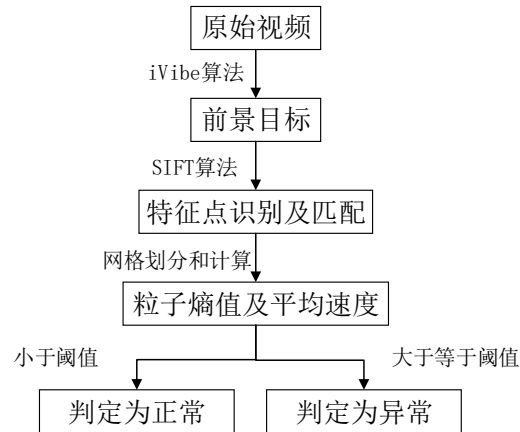
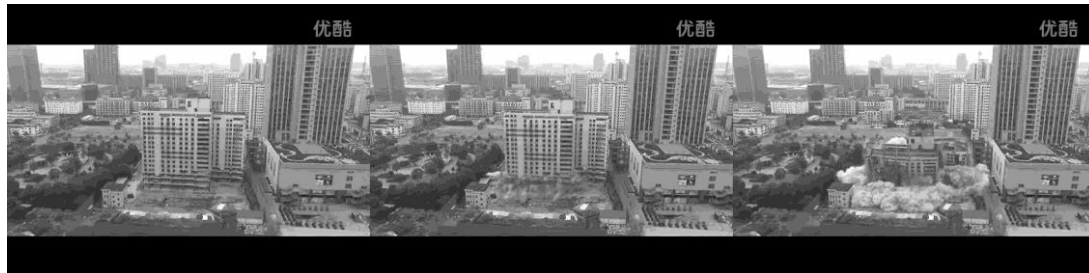


图 9-1 基于粒子熵值的人群异常检测算法

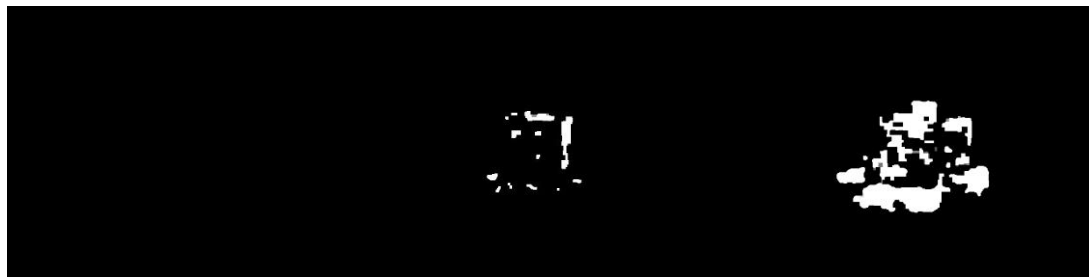
## 9.2 实验结果

### 9.2.1 爆炸检测结果

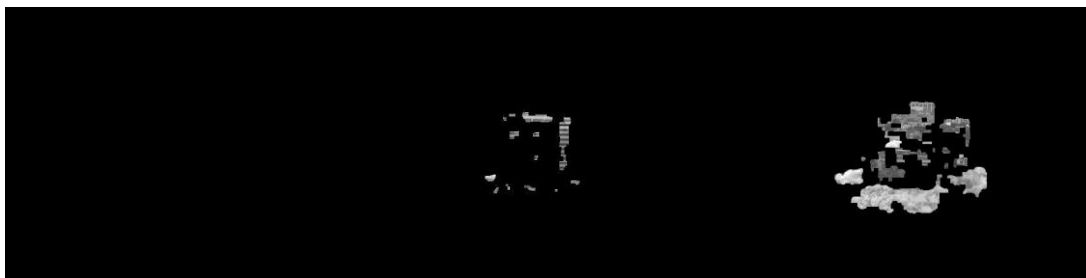
对于爆炸视频的前景提取结果如图 9-2 所示：



(a) 爆炸视频第 1/76/155 帧原图



(b) 爆炸视频第 1/76/155 帧蒙版



(c) 爆炸视频第 1/76/155 帧提取结果

图 9-2 爆炸视频第 1/76/155 帧前景提取结果

应用 9.1.1 节中的算法，我们可以估计出爆炸发生在第 76 帧，实际爆炸正是发生在第 76 帧，说明预测结果与实际情况相符。

### 9.2.2 人群异常行为检测结果

对于人群异常视频的前景提取结果如图 9-3 所示：



(a) 人群异常视频第 1/56/109 帧原图



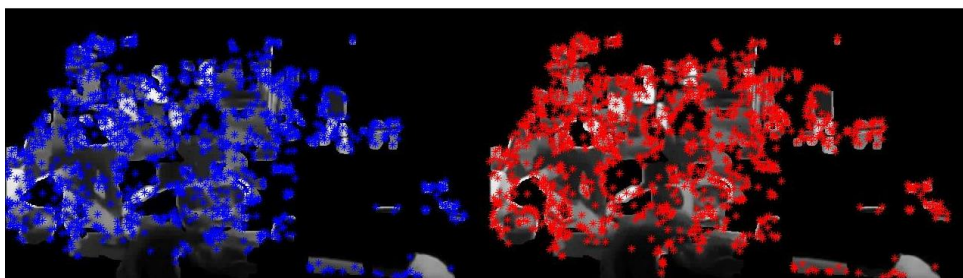
(b) 人群异常视频第 1/56/109 帧蒙版



(c) 人群异常视频第 1/56/109 帧提取结果

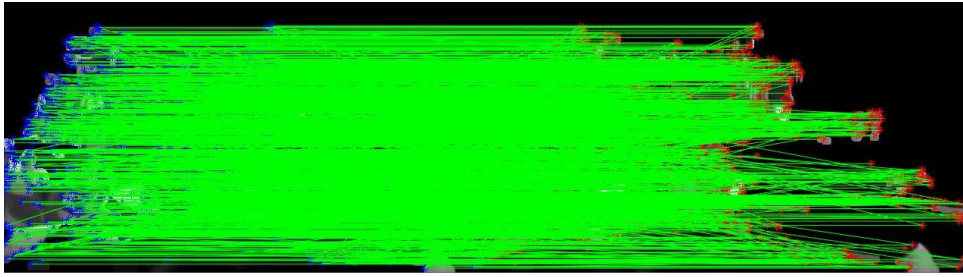
图 9-2 人群异常视频第 1/56/109 帧前景提取结果

首先应用 SIFT 算法识别特征点并进行匹配，得到图 9-3 的结果：



(a) 人群异常视频特征点识别





(b) 人群异常视频特征点匹配

图 9-3 人群异常视频特征点识别及匹配

根据匹配结果，应用 9.1.2 节当中基于粒子熵值的算法，可以得到特征点熵值变化图和速度变化图，如图 9-4：

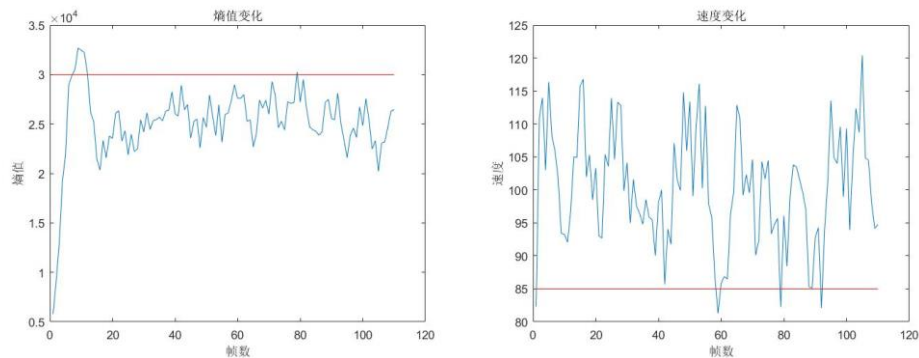


图 9-4 特征点熵值变化及速度变化曲线

对特征点熵值变化曲线和速度变化曲线进行分析。图中红色直线为设定的阈值，当速度超过阈值时，可根据熵值与阈值的对比确定异常情况的类型：熵值大于阈值时，说明特征点分布均匀，人群四处逃散；熵值小于阈值时，说明特征点分布集中，人群聚集。由图 9-4 可知，大部分帧数低于阈值，应判断为人群聚集。实际上，该视频中的人群始终向某一处聚集，与分析结果基本一致。

## 10. 模型评价与改进

### 10.1 模型优点

(1) 总体基于 Vibe 算法建模，引入中值滤波、形态学算法、仿射变换等多种方法不断对其进行优化。改进后得到的 iVibe 算法，在继承原有算法简洁高效的优势的同时，提高了鲁棒性，能适用于更多情况的前景提取；

(2) 引入评价指标，对前景提取结果进行定量分析；

(3) 提出简单可行的比例分析法，用来检测视频中是否存在显著前景目标，且结果准确率较高；

(4) 充分利用 SIFT 算法，将特征点的提取和匹配结果应用于多角度视频中相同前景目标的检测，以及异常事件的检测。

### 10.2 模型不足

(1) 对于前景目标与背景灰度比较接近的情况有待优化；

(2) 当前景目标静止或移动缓慢时，可能会误判为背景。

### 10.3 模型改进

分析本文得到的结果评价(见表 10-1)我们可以发现，背景预测的正确率普遍高于前景预测的正确率。这说明本文在进行前景提取结果的优化时，对于噪点和

背景扰动处理是比较好的。而前景提取结果稍差，一方面由于前景在图像中占比较小，且真值本身就不是完全精准的；另一方面，可能是由于腐蚀过度，有必要对进行腐蚀的结构元素  $S$  进行调整。

表 10-1 前景目标提取结果评价

评价指标 视频名称	背景正确率 TNR	前景正确率 TPR	总体正确率 TR
hall	0.9722	0.6779	0.9351
pedestrian	0.9897	0.6134	0.9720
waterSurface	0.9944	0.6837	0.9721
car6	0.9998	0.7292	0.9922

## 10.4 模型推广

在大数据时代的今天，智能监控相比于传统的只负责记录的监控更受到人们的重视与青睐。经过有效提取的前景目标，可以为视频监控提供有价值的参考信息，是智能化监控的重要基础。智能化监控不仅能够忽略监控视频中无用信息，还能节省大量的人力、财力。此外，还可以应用于多媒体领域对于视频的处理和分析，公安系统的犯罪检测、实时报警及预防，甚至是鱼雷、导弹等武器的研发和改进等。该技术应用前景广泛，能够带来巨大的经济效益和社会效益，具有很高的研究价值。

## 参考文献

- [1] 卢章平,孔德飞,李小蕾等.背景差分与三帧差分结合的运动目标检测算法[J].北京:计算机测量与控制,2013,21(12): 3315-3318
- [2] 汪国强,盖琪琳,于怀勇等.基于背景差分法的视频目标检测算法研究[J].哈尔滨:黑龙江大学工程学报,2014,5(4): 64-68
- [3] 侯宏录,李宁鸟,刘迪迪等.智能视频监控中运动目标检测的研究[J].西安:计算机技术与发展,2012,22(2): 49-52
- [4] O. Barnich, M. Van Droogenbroeck. ViBe: a powerful random technique to estimate the background in video sequences[C]. Taipei: IEEE International Conference on Acoustics, Speech and Signal Processing, 2009: 945 - 948,
- [5] O. Barnich, M. Van Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences[J]. IEEE Transactions on Image Processing, 2011, 20(6) : 1709-1724
- [6] M. Juhola, J. Katajainen,, T. Raita. Comparison of Algorithms for Standard Median[J]. IEEE Transactions on Signal Processing, 1991, 39(1): 204-208
- [7] 刘国宏,郭文明.改进的中值滤波去噪算法应用分析[J].北京:计算机工程与应用,2010,46(10): 187-189
- [8] 李晓飞,马大玮,粘永健等.图像腐蚀和膨胀的算法研究[J].天津:影像技术,2005(1): 37-39
- [9] 余烨,曹明伟,岳峰. EVibe: 一种改进的 Vibe 运动目标检测算法[J].北京:仪器仪表学报,2014,35(4): 924-931
- [10] 高晶,吴育峰,吴昆.基于角点检测的图像匹配算法[J].北京:仪器仪表学报,2013,34(8): 1717-1725
- [11] 王向军,王研,李智.基于特征角点的目标跟踪和快速识别算法研究[J].上海:光学学报,2007,27(2): 360-364
- [12] 曾文锋,李树山,王江安.基于仿射变换模型的图像配准中的平移旋转和缩放[J].天津:红外与激光工程,2001,30(1): 17-20
- [13] D.G. Lowe. Distinctive image features from scale invariant key points[J]. International Journal of Computer Vision, 2004, 60(2): 91-110
- [14] 傅卫平,秦川,刘佳等.基于 SIFT 算法的图像目标匹配与定位[J].北京:仪器仪表学报,2011,32(1): 1107-1112
- [15] 周洁,郭立君,张荣.基于粒子熵值的异常行为检测[J].石家庄:无线电通信技术,2015(3): 66-68



## 附录

注：本文代码运行环境均为 Matlab2014b

### 附录 1 评级指标算法代码

```
clc;clear;          %用于 1, 2, 3 问题的 评价指标
A=imread('4.jpg');  %真值图像
A= rgb2gray(A);
B=imread('8.jpg');  %待评价图像
[m,n]=size(A);
%二值化
for i=1:m
    for j=1:n
        if A(i,j)>150
            A(i,j)=255;
        else A(i,j)=0;
        end
        if B(i,j)>150
            B(i,j)=255;
        else B(i,j)=0;
        end
    end
end
%检测
Tp=0;Fp=0;Fn=0;Tn=0;
for i=1:m
    for j=1:n
        if A(i,j)==255 & B(i,j)==255
            Tp=Tp+1;
        else if A(i,j)==0 & B(i,j)==255
            Fp=Fp+1;
        else if A(i,j)==255 & B(i,j)==0
            Fn=Fn+1;
            Tn=Tn+1;
        end
    end
end
end
FPR=Fp/(Fp+Tn)
FNR=Fn/(Tp+Fn)
TRR=Tp/(Tp+Fn)
```

## 附录 2 iVibe 算法代码

```
%clear,clc;    %（本论文核心算法，基于 matlab2014b 环境下完美运行）
video = VideoReader('4p0.avi'); %输入视频的名字（同一文件夹下）
height = video.Height;          % 视频信息
width = video.Width;
% 参数
NumFrames = video.NumberOfFrames;
cardinality = 2;%基数
r = 20;%给定半径
n = 16;
%初始化
imageRGB = read(video,1);
image = rgb2gray(imageRGB);
bgpadd = padarray(image,[2 2],'replicate');
for i=3:height+2
    for j=3:width+2
        mm=unidrnd(5,[1,n])-3;
        nn=unidrnd(5,[1,n])-3;
        for k=1:n
            bgpad(i-2,j-2,k)=bgpadd(i+mm(k),j+nn(k));
        end
    end
end
%随机更新背景
for f = 2 : NumFrames
    imageRGB = read(video,f);
    image = rgb2gray(imageRGB);
    for i = 1:height
        for j = 1:width
            div = abs(bgpad(i,j,:) - double(image(i,j)));
            logic = div < r;
            bignum = sum(logic);
            if bignum > cardinality
                randz = randi(n);
                bgpad(i+1,j+1,randz) = image(i,j);
                %            随机改变(i,j)邻域
                randy = randi([-2 2]);
                randx = randi([-2 2]);
                bgpad(i+2+randy,j+2+randx,randz) = image(i,j);
            end
        end
    end
    end
    randbg = randi(n);
```

```

out = bgpad(2:height+1,2:width+1,randbg);
% figure(1); %展示提取出的背景
% imshow(out,[]);
% drawnow;
end
finalbg = mean(bgpad(2:height+1,2:width+1),3);
% figure(2); %最终提取的背景
% imshow(finalbg,[]);
% drawnow;
for f = 1 : NumFrames
    imageRGB = read(video,f);
    image1 = rgb2gray(imageRGB);
    for i = 1:height
        for j = 1:width
            divv = abs(bgpad(i,j,:) - double(image1(i,j)));
            logicc = divv < r;
            bignumm = sum(logicc);
            if bignumm > cardinality
                image1(i,j)=0;
            else
                image1(i,j)=255;
            end
        end
    end
end
%形态学的腐蚀膨胀处理
image1=imerode(image1,ones(3,3)); %腐蚀
image1=imdilate(image1,ones(3,3)); %膨胀
%连通域处理
[L,num] = bwlabel(image1); %二值图像连通区域标号，L 是标号图，num 是区域个数
for i=1:num
    if numel(find(L==i)) < 30 %去掉小联通区域
        image1(L==i) = 0;
    end
end
end
[LL,numm] = bwlabel(image1); %二值图像连通区域标号，L 是标号图，num 是区域个数
for i=1:numm
    if numel(find(L==i)) < 100 && numel(find(L==i)) > 40 %填充大联通区域
        image1(L==i) = 255;
    end
end
end
image1=medfilt2(image1); %中值滤波处理
imwrite(image1, strcat('E:\ljf\', num2str(f), '.jpg'), 'jpg'); %存储算法计算后所得的图像
end

```

### 附录 3 仿射变换及图像填补算法代码

```

clear all; clc          % 用于第三问 仿射变换+图像修补
filename1 = 'E:\MatlabCode\View2mat\Q3\1.jpg';
s=30; % s 为帧数
for n=2:s
    filename2 = strcat('E:\MatlabCode\View2mat\Q3\',num2str(n),'.jpg');
    hVideoSrcA = vision.VideoFileReader(filename1, 'ImageColorSpace', 'Intensity');
    hVideoSrcB = vision.VideoFileReader(filename2, 'ImageColorSpace', 'Intensity');
    imgA = step(hVideoSrcA); imgB = step(hVideoSrcB); ptThresh = 0.1;
    pointsA = detectFASTFeatures(imgA, 'MinContrast',ptThresh);
    pointsB = detectFASTFeatures(imgB, 'MinContrast',ptThresh);
    [featuresA, pointsA] = extractFeatures(imgA, pointsA);
    [featuresB, pointsB] = extractFeatures(imgB, pointsB);
    indexPairs = matchFeatures(featuresA, featuresB);
    pointsA = pointsA(indexPairs(:,1), :); pointsB = pointsB(indexPairs(:,2), :);
    % figure; showMatchedFeatures(imgA, imgB, pointsA, pointsB);
    % legend('A','B');
    [tform, pointsBm, pointsAm] = estimateGeometricTransform(pointsB, pointsA, 'affine');
    imgBp = imwarp(imgB, tform, 'OutputView', imref2d(size(imgB)));
    pointsBmp = transformPointsForward(tform, pointsBm.Location);
    % figure;
    showMatchedFeatures(imgA, imgBp, pointsAm, pointsBmp);
    % legend('A', 'B');
    H = tform.T; R = H(1:2, 1:2);
    theta = mean([atan2(R(2),R(1)) atan2(-R(3),R(4))]);
    scale = mean(R([1 4])/cos(theta));
    translation = H(3, 1:2);
    HsRt = [[scale*[cos(theta) -sin(theta);sin(theta) cos(theta)];translation],[0 0 1]];
    tformsRT = affine2d(HsRt);
    imgBold = imwarp(imgB, tform, 'OutputView', imref2d(size(imgB)));
    imgBsRt = imwarp(imgB, tformsRT, 'OutputView', imref2d(size(imgB)));
    [height,width]=size(imgA);
    for i=1:height
        for j=1:width
            if imgBsRt(i,j)==0
                imgBsRt(i,j)=imgA(i,j);
            end
        end
    end
    imgBsRt=medfilt2(imgBsRt);
    imwrite(imgBsRt,strcat('E:\ljf\',num2str(n),'.jpg'),'.jpg');
    % figure
    % imshow(imgBsRt);
end

```

#### 附录 4 比例分析法算法代码

```
clc;clear;          %比例分析法判断是否有显著前景
tic
DIR='E:\jf\喷泉可以\';
file=dir(strcat(DIR,'*.jpg'));          %读取所有 jpg 文件
s=size(file,1);          %图片总数
YZ=1/200;          %给定阈值
js=zeros(1,s);fai=zeros(1,s);          %初始化
for n=1:s
    fname = strcat(DIR, num2str(n), '.jpg');
    A = imread(fname);
    [height,width]=size(A);
    T=height*width;
    for i=1:height          %二值化
        for j=1:width
            if A(i,j)<10
                A(i,j)=0;
            else A(i,j)=1;
            end
        end
    end
    fai(n)=sum(sum(A));
end
ZD=min(fai);
%分析比例，判断是否有显著前景
for n=1:s
    if (fai(n)-ZD)/T>YZ
        js(n)=1;
    end
end
toc
B=find(js==1);
disp('已计算完毕，矩阵 B 中数据即为显著帧数')
```

## 附录 5 SIFT 算法代码

main

```
close all;clear all;clc
pic1=imread('E:\UC 下载\gmcm2017\2017 年试题\2017 最终赛题\D\2017 年全国研究生数学建模竞赛 D 题附件\第五问\1.jpg');
pic2=imread('E:\UC 下载\gmcm2017\2017 年试题\2017 最终赛题\D\2017 年全国研究生数学建模竞赛 D 题附件\第五问\2.jpg');
if length(size(pic1))==3
    pic1=rgb2gray(pic1); %对 1.jpg 进行灰度处理
end
if length(size(pic2))==3
    pic2=rgb2gray(pic2); %对 2.jpg 进行灰度处理
end
pic1=medfilt2(pic1); %对 1.jpg 进行滤波处理
pic2=medfilt2(pic2); %对 2.jpg 进行滤波处理
[des1,loc1]=SIFT(pic1);
[des2,loc2]=SIFT(pic2);
XY1=loc1(:,[2 1]);
XY2=loc2(:,[2 1]);
isSave=0;
figure(1)
isNumbered=0;
drawPoint(pic1,XY1,pic2,XY2,isNumbered,isSave); %显示两张图片的特征点
Num=3;
Thresh=0.85;
match=BidirectionalMatch(des1,des2,Num,Thresh); %获得两张图片特征匹配点的匹配信息
clear des1 des2
XY1=XY1(match(:,1),:); %1.jpg 的特征点
XY2=XY2(match(:,2),:); %2.jpg 的特征点
figure(2)
isNumbered=0;
drawLine(pic1,XY1,pic2,XY2,isNumbered,isSave); %用直线将两个图片的匹配点连起来
agl=getRotatedAngle(loc1,loc2,match); %两张图片的旋转角度
HistNum=180;
```

**BidirectionalMatch**

```
function match=BidirectionalMatch(des1,des2,Num,Thresh)
match12=UnidirectionalMatch(des1,des2,Num,Thresh);
match21=UnidirectionalMatch(des2,des1,Num,Thresh);
match21(:,[1,2])=match21(:,[2,1]);
match=[];
a1=match12(:,1);
b1=match21(:,1);
a2=match12(:,2);
```

```

b2=match21(:,2);
for k=1:size(match12,1)
    t=abs(a1(k)-b1)+abs(a2(k)-b2);
    IX=find(t==0);
    len=length(IX);
    if len
        match=[match;match21(IX,:)];
    end
end

function match=UnidirectionalMatch(des1,des2,Num,Thresh)
len1=size(des1,2);
C=(des1'*des2)./sqrt(((diag(des1'*des1))*(diag(des2'*des2)))));
[C,IX]=sort(C,2,'descend');
match=[];
for k=1:len1
    c=C(k,:);
    ix=IX(k,:);
    c=c(1:Num);
    ix=ix(c>Thresh);
    len=length(ix);
    if len
        mch=zeros(len,1);
        mch(:)=k;
        match=[match;mch,ix'];
    end
end
end

```

## drawLine

```

function drawLine(pic1,XY1,pic2,XY2,isNumbered,isSave
%用直线将两个特征点连起来
dif=size(pic1,2);
imshowpair(pic1,pic2,'montage');
hold on
drawPoint(pic1,XY1,pic2,XY2,isNumbered,0);
for k=1:size(XY1,1)
    line([XY1(k,1),XY2(k,1)+dif],[XY1(k,2),XY2(k,2)],'Color','g');
end
hold off
if isSave
    savePicture(gca);
end
end

```

### drawMatchedPicture

```
function drawMatchedPicture(pic,isSave)
%画出匹配图像
set(gcf,'Color','w');
imshow(pic);
if isSave
    savePicture(gca);
end\
```

### drawPoint

```
function drawPoint(pic1,XY1,pic2,XY2,isNumbered,isSave)
%画出特征点
dif=size(pic1,2);
imshowpair(pic1,pic2,'montage');
hold on
set(gcf,'Color','w');
switch isNumbered
    case 0
        plot(XY1(:,1),XY1(:,2),'b*');
        plot(XY2(:,1)+dif,XY2(:,2),'r*');
    case 1
        for k=1:size(XY1,1)
            plot(XY1(k,1),XY1(k,2),'b*');
            text(XY1(k,1)-4,XY1(k,2)+4,num2str(k),'FontSize',12,'Color','y');
        end
        for k=1:size(XY2,1)
            plot(XY2(k,1)+dif,XY2(k,2),'r*');
            text(XY2(k,1)+4+dif,XY2(k,2)-4,num2str(k),'FontSize',12,'Color','y');
        end
    end
end
hold off
if isSave
    savePicture(gca);
end
```

### drawRotatedAngleHist

```
function drawRotatedAngleHist(agl,HistNum,isSave)
hist(agl,HistNum);
hold on
set(gcf,'Color','w');
xlabel('Rotated Angle(°)');
ylabel('Number of Corners');
box off
```



```

hold off
if isSave
    savePicture(gcf);
end

```

### getMatchedPicture

```

function im=getMatchedPicture(pic1,pic2,T)
% 获得匹配图像
[row1,col1]=size(pic1);
[row2,col2]=size(pic2);
SZ=3*max(length(pic1),length(pic2));
cntrX=SZ/3;
cntrY=SZ/3;
im=zeros(SZ,SZ);
j=1:col2;
ti=ones(2,col2);
for i=1:row2
    ti(1,:)=i;
    tmp=round(T*cat(1,j,ti));
    nx=tmp(1,:)+cntrX;
    ny=tmp(2,:)+cntrY;
    for k=1:col2
        if nx(k)>=1 && nx(k)<=SZ && ny(k)>=1 && ny(k)<=SZ
            im(ny(k),nx(k))=pic2(i,k);
        end
    end
end
end
im=imresize(medfilt2(im),1,'bicubic');
Tpic1=zeros(SZ,SZ);
Tpic1(1+cntrY:row1+cntrY,1+cntrX:col1+cntrX,:)=pic1;
re=uint8(im)-uint8(Tpic1);
im(re==0)=Tpic1(re==0);
clear Tpic1 re
im=getCroppedPicture(im);
im=imresize(medfilt2(im),1,'bicubic');
im=uint8(im);

function im=getCroppedPicture(pic)
SZ=length(pic);
k=1;
while k<SZ
    if any(any(pic(k,:)))

```

```

        break
    end
    k=k+1;
end
ceil=k;

k=SZ;
while k>0
    if any(any(pic(k,:)))
        break
    end
    k=k-1;
end
bottom=k;

k=1;
while k<SZ
    if any(any(pic(:,k)))
        break
    end
    k=k+1;
end
left=k;

k=SZ;
while k>0
    if any(any(pic(:,k)))
        break
    end
    k=k-1;
end
right=k;

im=pic(ceil:bottom,left:right);

```

### getOptimizedMatchedIndex

```

function ind=getOptimizedMatchedIndex(agl,HistNum)
%获得匹配点的指标
[n,xout]=hist(agl,HistNum);
alpha=0.6;
[~,IX]=find(n>alpha*max(n));
n=n(IX);
xout=xout(IX);

```

```

theta=sum(xout.*n)/sum(n);
rg=[theta-1,theta+1];
ind=[];
for k=1:length(agl)
    if agl(k)>=rg(1) && agl(k)<=rg(2)
        ind=[ind,k];
    end
    if length(ind)>20
        break
    end
end
end

```

### getRotatedAngle

```

function agl=getRotatedAngle(loc1,loc2,match)
%获取旋转角度
ori1=loc1(match(:,1),4);
ori2=loc2(match(:,2),4);
agl=ori2-ori1;
agl=agl*180/pi;

```

### getTransformMatrix

```

function T=getTransformMatrix(pic1,XY1,pic2,XY2)
%获取变换矩阵
pic1=double(pic1);
pic2=double(pic2);
len=size(XY1,1);
tt=len*(len-1)*(len-2)/6;
rr=zeros(tt,1);
SEL=zeros(tt,3);
cnt=1;
for i=1:len-2
    for j=2:len-1
        for k=3:len
            if i<j && j<k
                SEL(cnt,:)=[i j k];
                cnt=cnt+1;
            end
        end
    end
end
end
for k=1:tt
    selection=SEL(k,:);
    x=XY2(selection,1)';

```

```

y=XY2(selection,2)';
X=XY1(selection,1)';
Y=XY1(selection,2)';
U=[x;y;ones(1,3)];
V=[X;Y;ones(1,3)];
T=V/U;
grayset=getGrayset(pic1,pic2,T);
if isempty(grayset)
    continue
end
d1=grayset(1,:);
d2=grayset(2,:);
rr(k)=(d1*d2')/sqrt((d1*d1')*(d2*d2'));
end
[~,indx]=max(rr);
selection=SEL(indx,:);
x=XY2(selection,1)';
y=XY2(selection,2)';
X=XY1(selection,1)';
Y=XY1(selection,2)';
U=[x;y;ones(1,3)];
V=[X;Y;ones(1,3)];
T=V/U;

function grayset=getGrayset(pic1,pic2,T)
% 获得灰度集
[row1,col1]=size(pic1);
[row2,col2]=size(pic2);
grayset=[];
j=1:col2;
ti=ones(2,col2);
for i=1:row2
    ti(1,:)=i;
    tmp=round(T*cat(1,j,ti));
    nx=tmp(1,:);
    ny=tmp(2,:);
    for k=1:col2
        if nx(k)>=1 && nx(k)<=col1 && ny(k)>=1 && ny(k)<=row1
            grayset=[grayset,[pic1(ny(k),nx(k));pic2(i,k)]];
        end
    end
end
end
end

```

## savePicture

```
function savePicture(h)
%保存图片
f=getframe(h);
str=[getTime '.jpg'];
imwrite(f.cdata,str);

function time=getTime
t=clock;
time='P';
for k=1:length(t)-1
    s=num2str(t(k));
    if length(s)==1
        s=strcat('0',s);
    end
    time=strcat(time,s);
    if k==3
        time=strcat(time,'_');
    end
end
s=t(6);
integer=floor(s);
decimal=num2str(s-integer);
decimal=decimal(2:end);
len=length(decimal);
if len==2
    decimal=strcat(decimal,'00');
elseif len==3
    decimal=strcat(decimal,'0');
end
integer=num2str(integer);
if length(integer)==1
    integer=strcat('0',integer);
end
time=strcat(time,integer,decimal);
```

## 附录 6 异常行为检测算法代码

### 附录 6-1 爆炸检测算法代码

```
close all
clear all
clc
%% 判断前景出现
DIR1='E:\UC 下载\gmcm2017\2017 年试题\2017 最终赛题\D\2017 年全国研究生数学建模竞赛 D 题附件\爆破
（蒙版）\';
for k=1:255
    filename1=strcat(DIR1,num2str(k),'.jpg');
    filename2=strcat(DIR1,num2str(k+1),'.jpg');
    imgM1=imread(filename1);imgM2=imread(filename2);
    M1=length(find(imgM1>0&imgM1<256));M2=length(find(imgM2>0&imgM2<256));
    if abs(M1-M2)>1000
        break;
    end
end
k+1
%%
pic1=imread('image1.jpg');
pic2=imread('image2.jpg');
if length(size(pic1))==3
    pic1=rgb2gray(pic1);
end
if length(size(pic2))==3
    pic2=rgb2gray(pic2);
end
pic1=medfilt2(pic1);
pic2=medfilt2(pic2);
[des1,loc1]=SIFT(pic1);
[des2,loc2]=SIFT(pic2);
XY1=loc1(:,[2 1]);
XY2=loc2(:,[2 1]);

isSave=0;

figure(1)
isNumbered=0;
drawPoint(pic1,XY1,pic2,XY2,isNumbered,isSave);

Num=3;
Thresh=0.85;
match=BidirectionalMatch(des1,des2,Num,Thresh);
```

```

clear des1 des2
XY1=XY1(match(:,1),:);
XY2=XY2(match(:,2),:);

figure(2)
isNumbered=0;
drawLine(pic1,XY1,pic2,XY2,isNumbered,isSave);

agl=getRotatedAngle(loc1,loc2,match);
HistNum=180;

```

## 附录 6-2 人群异常检测算法代码

```

close all
clear all
clc
tic
%% 平均速度以及粒子熵值计算
xMin=0;binSizeX=60;xMax=540;
yMin=0;binSizeY=60;yMax=960;
xGrid = xMin:binSizeX:xMax;
yGrid = yMin:binSizeY:yMax;
DIR1='E:\UC 下载\gmcm2017\2017 年试题\2017 最终赛题\D\2017 年全国研究生数学建模竞赛 D 题附件\人群
（还原）\';
for k=1:110
    filename1=strcat(DIR1,num2str(k),'.jpg');
    filename2=strcat(DIR1,num2str(k+1),'.jpg');
    pic1=imread(filename1);
    pic2=imread(filename2);
    if length(size(pic1))==3
        pic1=rgb2gray(pic1); %对 1.jpg 进行灰度处理
    end
    if length(size(pic2))==3
        pic2=rgb2gray(pic2); %对 2.jpg 进行灰度处理
    end
    pic1=medfilt2(pic1); %对 1.jpg 进行滤波处理
    pic2=medfilt2(pic2); %对 2.jpg 进行滤波处理
    [des1,loc1]=SIFT(pic1);
    [des2,loc2]=SIFT(pic2);
    XY1=loc1(:,[2 1]);
    XY2=loc2(:,[2 1]);

    isSave=0;

    figure(1)

```

```

isNumbered=0;
drawPoint(pic1,XY1,pic2,XY2,isNumbered,isSave); %显示两张图片的特征点

Num=3;
Thresh=0.85;
match=BidirectionalMatch(des1,des2,Num,Thresh); %获得两张图片特征匹配点的匹配信息
clear des1 des2
XY1=XY1(match(:,1),:); %1.jpg 的特征点
XY2=XY2(match(:,2),:); %2.jpg 的特征点

figure(2)
isNumbered=0;
drawLine(pic1,XY1,pic2,XY2,isNumbered,isSave); %用直线将两个图片的匹配点连起来

agl=getRotatedAngle(loc1,loc2,match); %两张图片的旋转角度
HistNum=180;
%% 计算平均速度
aveVt(k)= mean(sqrt((XY1(:,1)-XY2(:,1)).^2+(XY1(:,2)-XY2(:,2)).^2)); %各帧平均速度
if k<6
    aveV(k)=mean(aveVt); %前五帧各帧的速度
else
    aveV(k)=mean(aveVt(k-5:k)); %第五帧以后各帧的速度
end

%% 计算粒子熵值
h = zeros(length(xGrid)-1,length(yGrid)-1);
EHT(k)=0;
for i=1:length(xGrid)-1
    for j=1:length(yGrid)-1
        X=find(xGrid(1,i)<XY1(:,1)&xGrid(1,i+1)>XY1(:,1));
        Y=find(yGrid(1,j)<XY1(:,2)&yGrid(1,j+1)>XY1(:,2));
        h(i,j)=length(intersect(X,Y)); %每个网格的特征点数目
        EHT(k)=(EHT(k)+h(i,j)*log(h(i,j)+1)); %各帧熵值
    end
end
if k<6
    ET(k)=sum(EHT); %前五帧各帧的判别值
else
    ET(k)=sum(EHT(k-5:k)); %第五帧以后各帧的判别值
end
k
end
%% 判断人群异常帧数

```



```

for i=1:110
    if ET(i)>2e4 & aveV(i)>80
        YC(i)=1;
    end
end
figure(1);plot(YC);title('人群异常判断');xlabel('帧数');ylabel('判断值');
%% 画出熵值和速度的图像
figure(2);plot(ET);title('熵值变化');xlabel('帧数');ylabel('熵值');line([1,110],[2e4,2e4],'color','r');
figure(3);plot(aveV);title('速度变化');xlabel('帧数');ylabel('速度');line([1,110],[80,80],'color','r');

toc

```