



**“华为杯”第十四届中国研究生
数学建模竞赛**

学 校

重庆理工大学

参赛队号

11660006

队员姓名

1.陈兵

2.鲁姣

3.黄铮

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 基于监控视频的前景目标提取

摘 要

在中国安防产业中视频监控作为最重要的信息获取手段之一，能对目标有效的提取是重要而基础的问题，因此本文在此背景下，围绕对监控视频的前景目标有效的提取问题，研究了关于 1) 静态背景、动态背景的前景目标提取，能在背景复杂化的条件下，将运动的目标；2) 带抖动视频；3) 静态背景下多摄像头对多目标提取；4) 出现异常事件视频的判断等问题。给出了在不同情况下的前景目标提取方案。

问题一是针对静态背景且摄像头稳定的情况下，如何对前景目标提取的问题。在题目要求的基础上，通过对附件 2 中几组视频的分析，我们发现所有前景目标的运动短暂且光线明暗变化不明显。由于传统的 Vibe 算法能抑制鬼影但是运行效果不理想，因此采用建立在帧差法上改进的 Vibe 算法模型求解问题。并和传统的 Vibe 算法做对比，结果显示改进的 Vibe 算法明显优于传统的算法。而且对我们的算法模型做了效果评价。详细数据参考正文与附录。

问题二是在背景为动态（如有水波的产生）的情况下，对前景目标的提取问题。在此问题中，由于动态背景存在使得提取出的图像帧具有大量的干扰噪声，对前景目标的识别和提取造成干扰，因此我们提出一种基于全局外观一致型的运动目标检测法。在用 Vibe 算法对场景预检测的基础上，建立混合高斯模型分别对前景和背景进行全局外观建模，将运动目标检测出来，再引入超像素去噪，进

一步优化结果。详细结果参考正文与附录。

问题三是在问题一、二基础上的进一步深化。问题一及问题二是建立在摄像机自身稳定的基础上，而问题三则是在摄像机抖动的前提下。由于摄像机抖动一般具有旋转和平移，因此我们建立了坐标变换模型，以仿射变换作为模型基础，结合改进的高精度鲁棒的 RANSAC 算法提取前景目标，并对比灰度投影法，比较两种模型效果。具体效果见正文与附录。

问题四是对前三个问题的综合应用。运用基于混合高斯模型背景建模 Vibe 算法，对前景目标进行提取；选出具有显著前景目标的参考帧，计算参考帧中显著前景目标所占的面积，并将此面积设定为阈值 T ，遍历所有的视频帧，计算其前景目标所占的面积，通过相减对比，判定显著前景目标。若判定为显著前景目标则输出其所在视频帧中的帧号，并将显著前景出现的总帧数增加 1。

问题五是针对多摄像头多目标的协同跟踪问题。在问题二的混合高斯模型基础上我们建立了动态背景提取法，对不断变化的背景进行实时更新。再利用单应性约束法对多目标发生重叠现象进行投影将重叠目标区分开来，对目标进行定位。由于目标的不断运动，我们采用粒子滤波法对前景目标进行实时跟踪，通过多摄像头的协同通信完成对多前景目标的检测。

问题六是针对监控视频中前景目标出现异常情况时判断是否有异常事件的问题。在基于稀疏表示的模型上，引入混合高斯模型用于学习不同类型的运动特征规律，然后通过各个单高斯模型中的均值建立一个相似矩阵作为字典。以测试阶段生成的核矢量为基础，用该局部特征的核矢量计算基于稀疏表示的重构误差，并将其与已设定的阈值进行比较，如果重构误差大于阈值，则判为异常。

关键词： 监控视频，前景目标提取，Vibe 算法，混合高斯模型，目标跟踪

目录

1 问题重述.....	1
1.1 问题背景.....	1
1.2 问题提出.....	1
2 模型假设.....	2
3 符号说明.....	2
4 问题一的模型建立与求解.....	3
4.1 问题描述与分析.....	3
4.2 改进的 Vibe 算法.....	3
4.2.1 Vibe 算法原理.....	3
4.2.2 改进的 Vibe 算法.....	4
4.3 问题的求解结果.....	5
5 问题二模型的建立与求解.....	7
5.1 问题描述及分析.....	7
5.2 模型的建立与求解.....	7
5.2.1 Vibe 算法检测.....	8
5.2.2 混合高斯模型建立.....	8
5.2.3 SLIC 超像素分割去噪.....	10
5.3 实验结果与分析.....	11
5.4 模型的评价.....	13
6 问题三模型建立与求解.....	13
6.1 问题描述及分析.....	13
6.2 模型的建立.....	13
6.2.1 仿射变换.....	13
6.2.2 基础矩阵和 RANSAC 算法模型.....	14
6.2.3 改进的 RANSAC 算法.....	15
6.2.4 M-estimators 法.....	15
6.3 结果与分析.....	16
7 问题四模型的建立与求解.....	17
7.1 问题描述及分析.....	17
7.2 问题的求解.....	17
7.3 实验结果与分析.....	17
8、 问题五模型建立与求解.....	18
8.1 问题描述及分析.....	18
8.2 模型建立.....	18
8.2.1 基于单应性约束的目标定位.....	19
8.2.2 粒子滤波目标跟踪.....	19
8.2.3 多摄像头协同跟踪.....	20
8.3 实验结果与分析.....	20
9 问题六.....	22
9.1 问题描述及分析.....	22

9.2 模型的建立与求解.....	22
9.2.1 混合高斯模型.....	23
9.2.2 生成字典.....	24
9.2.3 视频异常事件检测.....	24
9.3 实验结果与分析.....	25
10 模型的评价.....	27
10.1 模型的优点.....	27
10.2 模型的缺点.....	27
参考文献.....	28
附录.....	29

1 问题重述

1.1 问题背景

视频监控是中国安防产业中最为重要的信息获取手段。随着“平安城市”建设的顺利开展，各地普遍安装监控摄像头，利用大范围监控视频的信息，应对安防等领域存在的问题。近年来，中国各省市县乡的摄像头数目呈现井喷式增长，大量企业、部门甚至实现了监控视频的全方位覆盖。如北京、上海、杭州监控摄像头分布密度约分别为 71、158、130 个/平方公里，摄像头数量分别达到 115 万、100 万、40 万，为我们提供了丰富、海量的监控视频信息。

目前，监控视频信息的自动处理与预测在信息科学、计算机视觉、机器学习、模式识别等多个领域中受到极大的关注。而如何有效、快速抽取出监控视频中的前景目标信息，是其中非常重要而基础的问题。这一问题的难度在于，需要有效分离出移动前景目标的视频往往具有复杂、多变、动态的背景。这一技术往往能够对一般的视频处理任务提供有效的辅助。以筛选与跟踪夜晚时罪犯这一应用为例：若能够预先提取视频前景目标，判断出哪些视频并未包含移动前景目标，并事先从公安人员的辨识范围中排除；而对于剩下包含了移动目标的视频，只需辨识排除了背景干扰的纯粹前景，对比度显著，肉眼更易辨识。因此，这一技术已被广泛应用于视频目标追踪，城市交通检测，长时场景监测，视频动作捕捉，视频压缩等应用中。

1.2 问题提出

问题 1：针对不包含动态背景以及摄像头稳定拍摄的监控视频，构造合理的数学模型提取前景目标（如人，车等），并对所提出的模型设计有效的求解方法，从而可以实现类似图 1 的应用效果。



图 1 左图：原视频帧；右图：分离出的前景目标

问题 2：由于现实环境中拍摄的监控视频大都是包含动态背景信息（如流水，树叶的飘动等），构造合理的数学模型提取前景目标，并对所提出的模型设计有效的求解方法。应用效果如图 2 所示。



图 2 几种典型的动态视频背景：树叶摇动，水波动，喷泉变化，窗帘晃动

问题 3：现实生活中由于摄像机固定的不牢固，在监控视频中，当监控摄像头发生晃动或偏移时，视频也会发生短暂的抖动现象（该类视频变换在短时间内

可近似视为一种线性仿射变换，如旋转、平移、尺度变化等）。那么对这种类型的视频，我们如何有效地提取前景目标呢？

问题 4：对于附件 3 中提供的 8 组视频，分类考虑静、动态背景下摄像头抖动与否，利用问题一、二、三构建的数学模型方法，从每组视频中选出包含显著前景目标的视频帧号，注明前景目标是出现于哪一个视频（如 Campus 视频）的哪些帧（如 241-250，421-432 帧）。

问题 5：针对多摄像头从不同角度同时拍摄同一地点的多个监控视频中，利用各个视频中前景目标之间的相关性，对各个前景目标进行有效检测和提取，能够同时对多目标进行跟踪检测。

问题 6：通过所建立的模型获取有效的前景目标信息，自动判断监控视频中无人群众短时聚集、人群惊慌逃散、群体规律性变化（如跳舞、列队排练等）等异常事件，设计出相应的事件检测方案。

2 模型假设

- 1、假定摄像机的正常扫描运动在一定时间内具有方向和幅值的一致性，是平滑的低频运动，而随机振动则包含了快速的运动参数变化。
- 2、背景与提取的前景目标的色差比较大。
- 3、基于附件 2 中的不带晃动静态背景上，视频中没有明显的运动前景目标阴影的出现，忽视视频中运动目标的阴影。
- 4、动态背景中的动态场景是呈规律性的变化。
- 5、抖动的摄像机只有平移与旋转两种方式，不考虑其他因素。

3 符号说明

符号	意义
ω_0	前景像素点数目占整幅图像的比例
μ_0	前景像素点平均灰度值
ω_1	背景像素点数目占整幅图像的比例
μ_1	背景像素点平均灰度
μ	整幅图像的平均灰度
K	高斯模型的个数
μ_k	第 K 个高斯成分的均值
τ_k	第 K 个高斯成分的方差
σ_k	第 K 个高斯成分的标准差
w_k	第 K 个高斯成分的权值
$w_{k,loci}$	每个单高斯模型的权重值
x^{test}	输入测试数据
y^{test}	核矢量
Φ	合成字典
STER (Significant Target Extraction Rate)	显著目标提取率

4 问题一的模型建立与求解

4.1 问题描述与分析

本题是对一个不含动态背景而且摄像头稳定拍摄的视频, 对其进行前景目标的提取。本题探寻的是如何提取前景目标的问题, 因此我们需要去除检测前景目标会引入的鬼影, 以及消除视频中的噪声。解决本问题需要考虑的问题为: 如何进行运动目标的提取, 以及如何消除视频中的噪声和运动目标隐藏在背景模型中的鬼影。

4.2 改进的 Vibe 算法

4.2.1 Vibe 算法原理

Vibe 算法是一种高效的像素级背景建模算法^[2], 其核心部分是由模型初始化、前景检测和背景模型更新组成。初始背景模型是采用邻近像素来近似填充, 然后通过比对当前像素值和背景模型来检测前景, 从第二帧便开始了运动目标的检测及背景模型的更新工作, 流程图如图 4.1 所示。

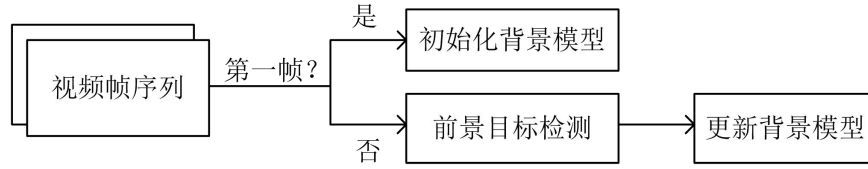


图 4.1 Vibe 前景检测算法

假定 x 表示当前帧图像的一个像素点, $p(x)$ 表示该点的像素值, 背景模型的建立就是为图像中每个像素建立各自地样本集, 这里像素点 x 样本集就能表示为:

$$S(x) = \{P_1, P_2, P_3, \dots, P_n\} \quad (4.1)$$

这里样本集的大小为 n , 通常这个 n 的值可以取 20, 也就是为每个像素点建立 20 个背景样本, 组成一个背景样本集。假如图像长为 M , 宽为 N , 那么总样本集大小就为 $M*N*n$, 背景模型建立好后开始对样本集进行初始化, 通过视频序列第一帧就可以完成, 背景样本集的填充方法为随机性的选择当前像素点的邻域像素, 用它邻域像素点的值来完成填充。

初始化完成之后, 就要开始对运动目标进行检测。在 Vibe 算法中, 前景检测过程可以看作是对图像中每个像素进行分类的过程, 也就是将像素根据一定规则分类为前景像素和背景像素两种。

这里我们设 $P_t(x)$ 表示的是第 t 帧图像在像素点 x 处的像素值, 通过前面的初始化可得像素点 x 的背景样本集为: $S(x) = \{P_1, P_2, P_3, \dots, P_n\}$ 。在如图 4.3 像素具体分类过程可以形象化地在 2-D 欧式空间中表示, 其中 $SR(P_t(x))$ 表示的是以像素点 x 为中心, R 为半径的圆形区域, 图中点 P_1, P_2, \dots, P_6 表示的都是像素点 x 背景样本集中的像素值。如果 $SR(P_t(x))$ 表示的区域范围和背景样本集 $S(x)$ 的交集样本格式大于某个阈值, 那么像素点 x 就判定为背景像素点, 用公式表示如下所示:

$$\# \{ \{ S_R(p_t(x)) \} \cap \{ P_1, P_2, P_3, \dots, P_n \} \} \geq U_{\min} \quad (4.2)$$

计算当前像素点 x 的像素值 $P_t(x)$ 和它背景样本集中每个样本值的距离，当这个距离小于阈值即欧式距离 R ，那么近似的样本像素点个数加一，最后若求得的近似样本像素点的数目大于某个设定的阈值 U_{min} ，就认为像素点 x 为背景像素。

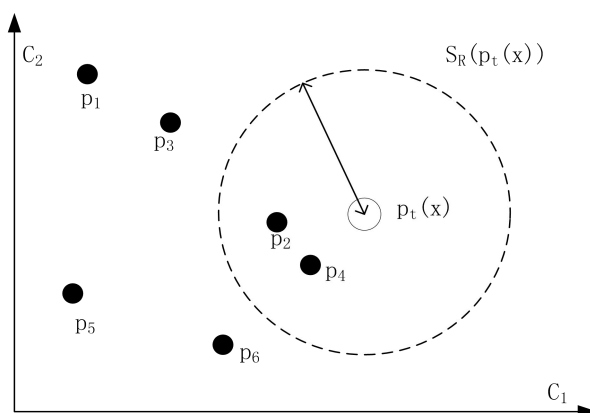


图 4.2 像素点 $p_t(x)$ 的分类过程

4.2.2 改进的 Vibe 算法

我们引入了基于 Otsu 阈值的鬼影抑制以及基于质心运动速度的背景自适应更新，对 Vibe 目标检测算法进行改进。如图 (4.3) 所示，鬼影抑制模块采用 Otsu 算法计算当前帧的最佳分割阈值，在传统 Vibe 判别的基础上，对前景像素点进行二次判别，抑制鬼影。背景更新率调整过程：对于检测出的前景目标，计算其质心位置，并计算帧间的质心差，根据质心差调整背景的更新率，能够很好解决传统 Vibe 算法由于采用固定更新率导致的残影孔洞等问题。

在鬼影抑制过程中^[3]，为了快速得到二次判别的最佳阈值，引入 Otsu 算法。Otsu 算法也称最大类间差法，其过程描述如下：

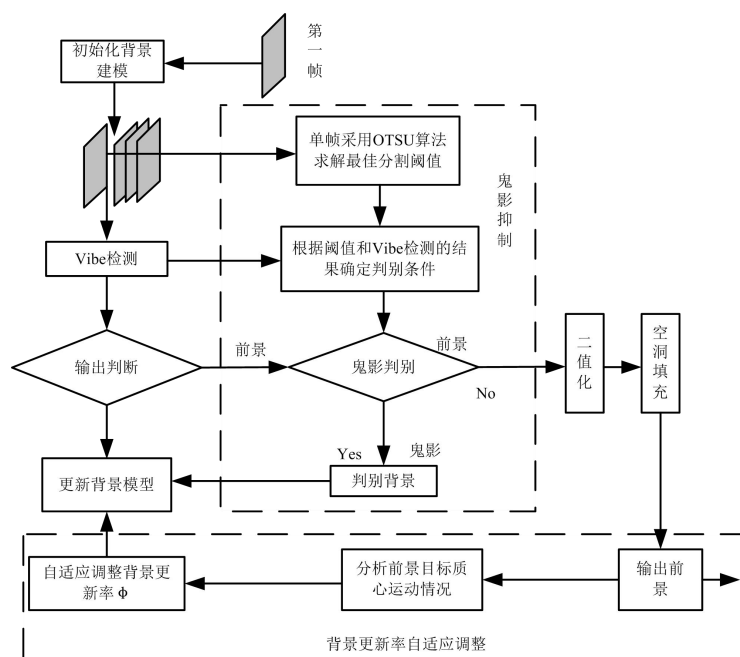


图 4.3 改进的 Vibe 算法流程图

(1) 计算当前帧的最佳分割阈值。假设当前图像帧的灰度级为 L ，灰度范围为 $[0, L-1]$ ，分割阈值为 t ，可以将图像分为灰度级为 $[0, t]$ 的区域 A 以及灰度级为 $[t+1, L-1]$ 的区域 B ，其中， A 、 B 分别代表前景以及背景。定义类间方差：

$$\sigma^2 = \omega_0(\mu_0 - \mu)^2 + \omega_1(\mu_1 - \mu)^2 = \omega_0\omega_1(\mu_0 - \mu_1)^2 \quad (4.3)$$

其中， ω_0 表示前景像素点数目占整幅图像的比例，平均灰度值为 μ_0 ； ω_1 表示背景像素点数目占整幅图像的比例，平均灰度为 μ_1 ，整幅图像的平均灰度为 μ 。类间方差越大，则两个区域差别越大，能更好地进行图像的分割。当 σ^2 取得最大值时的灰度值即为最佳阈值：

$$t^* = \text{Arg Max}_{0 \leq t \leq L-1} [\omega_0\omega_1(\mu_0 - \mu_1)^2] \quad (4.4)$$

(2) 确定判别规则。在经过 Vibe 算法检测得到的背景像素点中随机选择 M 个，计算这些像素点灰度的平均值，判别规则为：

$$\begin{aligned} \text{if } \overline{v_M} \leq t^* \quad f(x) &= \begin{cases} v(x) > t^*, \text{ 判别为前景} \\ v(x) \leq t^*, \text{ 判别为背景} \end{cases} \\ \text{else} \quad f(x) &= \begin{cases} v(x) < t^*, \text{ 判别为前景} \\ v(x) \geq t^*, \text{ 判别为背景} \end{cases} \end{aligned} \quad (4.5)$$

(3) 利用 (2) 得到的判别规则对通过 Vibe 算法判断出的前景进行二次判别，其中 $v(x)$ 代表 Vibe 判别出的前景。

Vibe 算法的背景更新过程并不需要对新的一帧的背景模型中的所有像素值进行更新，当一个像素点判别为背景时，以 $1/4$ 的概率更新该像素点以及其邻域像素点的样本集，从而保证了算法的效率。 ϕ 影响着背景模型的更新速度， ϕ 越大背景样本集更新的速度越慢， ϕ 越小背景样本集更新的速度越快。对 Vibe 算法引入了帧差测量值 ΔD 来衡量目标的运动速度，使得 ϕ 能够自适应地调整。帧差测量值 ΔD 表示一段时间内运动目标质心的平均移动速度，即

$$\Delta D = \sum_{i=1}^f d_i / f, \quad \phi = \begin{cases} \phi_1 = 8, & \Delta D \geq \tau_2 \\ \phi_2 = 16, & \tau_1 \leq \Delta D < \tau_2 \\ \phi_3 = 32, & \Delta D < \tau_1 \end{cases} \quad (4.6)$$

其中， d_i 为当前帧与前一帧的运动目标的相对质心差， f 为要统计的帧数。首先计算质心的二维矢量差，然后将其转化为一维距离。为了与更新速度相对应， $f = \phi_1$ 。本文引入 ϕ_1 、 ϕ_2 、 ϕ_3 种更新率。对于 τ 的选取，一般 $\tau_1 = 2$ ， $\tau_2 = 8$ 。若 $\Delta D < \tau_1$ 表明当前目标运动速度偏慢，需要选取较大的 ϕ_3 ，降低背景模型更新速度，以延长缓慢移动的物体融入到背景样本集的时间，从而减少检测到的前景目标出现空洞以及不完整的情况，以保证检测目标的准确性；若 $\Delta D > \tau_2$ ，则表明当前目标运动速度较快，需要选取较小的 ϕ_3 ，提高背景模型更新速度，使得运动目标移动过的地方能够快速恢复为背景，得到更好的前景检测结果。

4.3 问题的求解结果

我们将 Vibe 算法^[4]与改进的 Vibe 算法模型做了对比。我们用附件 2 中静态背景视频 office 作为检测对象，从中提取出第 45、46、47、48 帧作为说明对象，如图 4.4 所示。并且将 office mask 视频中同样的帧数提取出来如图 4.5 所示，与

我们采用上述两种算法模型对 office 视频提取的前景目标做对比，其效果图分别如图 4.6 和图 4.7 所示。



图 4.4 office 第 45、46、47、48 帧原图



图 4.5 office 第 45、46、47、48 帧 mask



图 4.6 office 第 45、46、47、48 帧 vibe 算法



图 4.7 office 第 48 帧改进的 vibe 算法

我们可以从中看出，采用传统的 vibe 算法并没有能完全消除鬼影，而且由于图中还有灯的出现导致提取出的前景目标有噪点，该算法并不能较准确地提取目标。而我们采用的改进的 vibe 算法模型不仅能很好地提取前景目标，而且对消除鬼影和去噪方面都有很好的效果。而且相比较于 office mask 视频中的前景目标，我们的模型虽然仍有有空洞的出现，但是已经能较好地提取目标的轮廓，辨识度比较高。

我们采用 IEEE Change Detection Workshop(CDW)评价检测标准来定量地进

行客观评价分析。定义 TP 为检测到的正确的目标点数, FP 为错误目标点数, FN 为错误背景点数, TN 为正确背景点数。评价内容包括:

$Re=(Recall)=TP/(TP+FN);$
 $Sp=(Specificity):TN/(TN+FP);$
 $FPR(False\ Positive\ Rate):FP/(FP+TN);$
 $FNR(False\ Negative\ Rate): FN/(TP + FN);$
 $PWC(Percentage\ of\ Wrong\ Classifications):100*(FN+FP)/(TP+FN+FP+TN);$
 $Precision:TP/(TP+FP);$
 $F-Measure:(2*Precision*Recall)/(Precision+Recall)。$

表 4.1 评价参数比较

Review/static video	hall	office	smoke
Re (Recall)	0.6861	0.7118	0.5455
Sp (Specificity)	0.9973	0.9963	0.9933
FPR (False Positive Rate)	0.0424	0.0669	0.1046
FNR (False Negative Rate)	0.3139	0.2882	0.4545
PWC (Percentage of Wrong Classification)	5.2774	4.7409	5.6544
Precision	0.9801	0.9718	0.9113
F_Measure	0.9072	0.8217	0.6825

我们分别对三个视频做了评价分析如表 4.1 所示, 可知在错判率和误判率方面, 我们的模型的相关参数指标均较小, 正确检测率和精度也较高。这也从侧面反映出我们建立的模型的可靠性,

5 问题二模型的建立与求解

5.1 问题描述及分析

问题一中, 假设背景为静态的, 即拍摄设备处于静止状态, 但在现实生活中, 由于实际需要, 视频拍摄状况复杂很多, 因为不但目标在运动, 有时摄像机也要运动, 镜头也会发生切换等, 也就有了动态背景情况下的前景检测。

本文提出一种基于全局外观一致型的运动目标检测方法。首先, 通过 ViBe 算法对场景做出初步的检测, 将检测结果作为初步数据; 其次, 根据初始的检测结果, 使用 GMM 对前景和背景分别进行全局外观建模, 优化检测结果。同时, 引入超像素分割, 抑制噪声, 并进一步优化检测结果。

5.2 模型的建立与求解

背景减除法是在运动目标检测中最常用的方法, 背景建模作为背景减除法中最关键的一个环节, 复杂场景中动态背景模型的建立尤为困难, 由于背景的动态变化, 很难对场景做出一个静态的背景模型, 常用的背景模型根据一帧或者几帧来确定初始背景模型的方法在复杂场景中变得不够合理, 如果动态背景的变化较大, 而且在一序列的视频图像中一直存在着较大的变化, 如果首先根据初始一段序列的视频帧建立背景模型, 再进行背景模型的更新, 背景模型会始终跟不上

真实背景模型的变化。

本文提出一种基于全局外观一致型的运动目标检测方法。首先，通过 Vibe 算法对场景做出初步的检测；其次，根据初始的检测结果，使用 GMM 对前景和背景分别进行全局外观建模，优化检测结果。同时，引入超像素分割，抑制噪声，并进一步优化检测结果。算法流程图如图 5.1 所示。

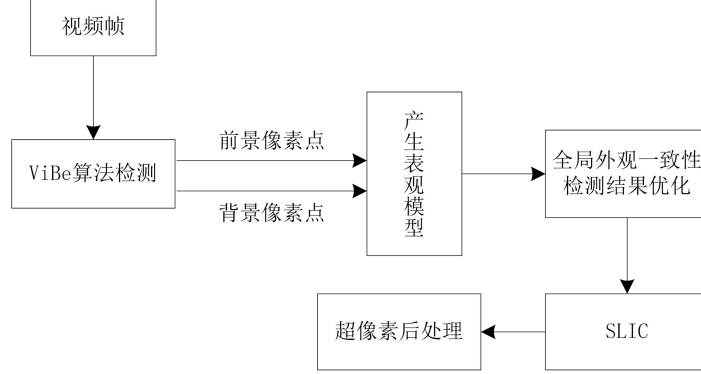


图 5.1 算法流程图

5.2.1 Vibe 算法检测

Vibe 是一种像素级别的背景建模，前景检测算法，该算法与其他算法的不同之处在于所建立背景模型的更新方式：随进选择需要替换的像素样本，随机选择邻域像素进行更新。在无法确定像素变化的模型时，随机的更新策略，在一定程度上可以模拟像素变化的不确定性。本章方法利用 Vibe 算法^[8]对场景做运动目标检测，把得到的检测结果作为初步数据，用来做进一步的处理。

5.2.2 混合高斯模型建立

本文基于全局外观一致性，对已经得到的前景像素点和背景像素点分别做出表观模型，通过判断像素点在区分性表观模型中的概率，使得前景与背景像素点更具区分性，并修正了部分被检测为前景的背景点，使检测结果得到进一步优化。本文在复杂场景中，把其中的前景像素点和背景像素点分别进行高斯建模，最终建立的模型即为前景或背景的区分性表观模型。高斯模型利用正态分布曲线精确的量化事物，将一个事物分解为若干的基于高斯概率密度函数形成的模型。混合高斯是单一高斯概率密度函数的延伸。本文中把 Vibe 算法检测结果中前、背景像素点分别作为前、背景混合高斯模型的样本集：每个像素的像素值作为随机变量的样本集，像素点集合 $\{x_1, x_2, x_3, \dots, x_n\}$ 即为模型的输入，单个采样点 x_t 服从的混合高斯分布的概率密度函数：

$$p(x_t) = \sum_{i=1}^k w_i \times \eta(x_t, \mu_k, \tau_k) \quad (5.1)$$

$$\eta(x_t, \mu_k, \tau_k) = \frac{1}{|2\pi\tau_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_t - \mu_k)^T \tau_k^{-1} (x_t - \mu_k)} \quad (5.2)$$

$$\tau_k = \sigma_k^2 \quad (5.3)$$

其中 K 为高斯模型的个数， μ_k 表示第 K 个高斯成分的均值， τ_k 、 σ_k 表示

第 K 个高斯成分的方差和标准差, w_k 表示第 K 个高斯成分的权值, 这些参数可以通过 EM 算法的获得。为得到混合高斯模型中的 K 个分布参数, 假设给定样本集 $\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \dots, \mathbf{x}^m\}$, 样例之间相互独立, 为找到每个样例中的类别 z , 使得 $p(\mathbf{x}, z)$ 取最大值, $p(\mathbf{x}, z)$ 的最大似然估计为

$L(\theta) = \sum_{i=1}^m \log p(\mathbf{x}; \theta) = \sum_{i=1}^m \log \sum_z p(\mathbf{x}, z; \theta)$ 。首先对极大似然取对数, 之后对每个样例的可能类别求出其联合分布概率之和。对每个样例 i , Q_i 表示样例隐含 z 的某分布, Q_i 满足 $\sum_z Q_i(z) = 1, Q_i(z) \geq 0$ 。那么

$$\sum_i \log p(\mathbf{x}^i; \theta) = \sum_i \log \sum_{z^i} p(\mathbf{x}^i, z^i; \theta) \quad (5.4)$$

$$= \sum_i \log \sum_{z^i} Q_i(z^i) \frac{p(\mathbf{x}^i, z^i; \theta)}{Q_i(z^i)} \quad (5.5)$$

$$\geq \sum_i \sum_z Q_i(z^i) \log \frac{p(\mathbf{x}^i, z^i; \theta)}{Q_i(z^i)} \quad (5.6)$$

式 (5.5) 到 (5.6) 为 Jensen 不等式, 为使 (3.4) 式最大化, 对每个样本 i 求 \mathbf{x} 的似然函数求解很困难, 所以用了 Jensen 不等式, 采用 EM 算法估计未知参数期望值, 给出当前参数估计, 调整参数后, 重新估计分布参数, 以使数据似然性最大, 给出了未知变量的期望估计。通过 EM 算法得到高斯模型参数为:

$$w^i = Q_i(z^i = j) = P(z^i = j | \mathbf{x}^i; \phi, \mu, \Sigma) \quad (5.7)$$

推导得高斯模型中均值 μ 的模型更新公式为

$$\mu_j = \frac{\sum_{i=1}^m w^i \mathbf{x}^i}{\sum_{i=1}^m w^i} \quad (5.8)$$

Φ_j 的更新公式为

$$\phi_j = \sum_{i=1}^m \sum_{j=1}^k w^i \log \frac{\frac{1}{2\pi^{\frac{n}{2}} |\Sigma_j|^{\frac{1}{2}}} \exp(-\frac{1}{2}(\mathbf{x}^i - \mu_j)^T \Sigma_j^{-1}(\mathbf{x}^i - \mu_j))}{w_j^i} \quad (5.9)$$

其中 ϕ 和 μ 确定的情况下, 式中分子部分为常数, 又 ϕ_j 满足约束条件为 $\sum_{j=1}^k \phi_j = 1$ 。

故对式 (5.9) 进行优化, 构造拉格朗日乘子, 得到

$$\phi_j = \frac{1}{m} \sum_{i=1}^m w_j^i \quad (5.10)$$

由此便可得到高斯模型的参数, 有混合高斯模型的参数即为所建立的区别性表观模型, 在得到产生式表观模型后, 通过比较像素点在前、背景模型中的匹配程度来判定像素点最终属于前景或背景, 若像素点在背景模型中的概率小于设定阈值 T_1 , 在前景模型中的概率大于设定阈值 T_2 , 则判断为前景像素点, 反之则

为背景像素点。即：

$$v(x) \in \begin{cases} 1, (p_1 < T_1) \wedge (p_2 > T_2) \\ 0, (p_1 > T_1) \vee (p_2 < T_2) \end{cases} \quad (5.11)$$

公式（5.11）中 1、0 分别表示前景与背景， p_1 、 p_2 分别表示像素点 $v(x)$ 在前景模型和背景模型中的概率。

由于上述检测结果中仍然存在一些干扰性的零散像素点和像素团块，本文引入基于超像素的后期处理，以排除这些干扰性像素的影响，使得检测结果更加准确和鲁棒。超像素在找寻图像中具有相近特征的像素点区域时不会破坏物体完整性，在图像分割中可以保留物体边界信息。在运动目标检测方法往往基于像素点进行判断，在复杂场景中难免出现一些干扰因素，超像素分割根据像素点之间的特征相似性把相似像素点组合成超像素分割子区域，通过超像素的分割，在检测时对点的处理基础上增加了对超像素区域的处理，在视觉上保证了处理合理性。在运动目标检测方法如帧间差分、背景减除法等，是基于像素点进行判断，对于像素点敏感，在动态背景中检测时，由于背景动态的特点往往由于干扰会对运动目标检测结果带来影响。在超像素子区域中，计算超像素子区域中出现前景点个数在子区域中的比例，通过设定阈值的方法来判断超像素子区域中是否存在真实前景。

5.2.3 SLIC 超像素分割去噪

本文采用 SLIC 超像素分割方法，由 LAB 空间以及 x 、 y 像素坐标共 5 维空间来计算，算法流程为：

（1）聚类中心的初始化：首先设定超像素数量，在图像中均匀的分配超像素，假设图像总像素点个数为 N ，超像素个数为 K ，则每个超像素大小为 N/K ，相邻聚类中心的距离（步长）近似为 $S = \sqrt{N/K}$ 。

（2）在 $n \times n$ 邻域内重新选择聚类中心（ n 取 3），计算该邻域内所有像素点的梯度值，将聚类中心移到邻域内梯度最小的地方，这样是为了防止聚类中心落在梯度较大的轮廓边界上，影响后续聚类效果。

（3）在聚类中心的 $2S \times 2S$ 的邻域内为每个像素点分配类标签，与标准的 K-means 在整张图中的搜索不同，SLIC 搜索范围限制为 $2S \times 2S$ ，可以加速算法收敛。

（4）对颜色距离和空间距离的度量，对于每个搜索到的像素点，分别计算它和该聚类中心的距离。距离计算方法如下：

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \quad (5.12)$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (5.13)$$

$$D' = \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2} \quad (5.14)$$

其中， d_c 代表颜色距离， d_s 代表空间距离， N_s 是类内最大空间距离，定义为 $N_s = S = \sqrt{N/K}$ ，适用于每个聚类，最大的颜色距离 N_c 既随图片不同而不同，又随聚类不同而不同，所以我们取一个固定阐述 m （取值范围[1,40]，一般取 10）代替。最终的距离度量 D' 如下：

$$D' = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{s}\right)^2} \quad (5.15)$$

每个像素点都会被多个聚类中心搜索到，所以每个像素点都会有一个与周围聚类中心点的距离，取最小值对应的聚类中心作为像素点的聚类中心。

(5) 通过 10 次迭代变化，大部分图像可以得到理想效果（即每个像素点聚类中心不再发生变化）。

(6) 连续性增强。经过迭代优化往往出现一些小问题：例如多联通情况、超像素尺寸较小、单个超像素被切割成多个不连续超像素等。可以通过增强连通性解决：为超像素建立标记表，顺序遍历标记表，将不连续的超像素、尺寸过小超像素重新分配给临近的超像素，遍历过的像素点分配给相应的标签，直到所有点遍历完毕为止。

设超像素子区域中像素点的个数为 n ，此 n 个像素点的超像素标签一致，设这 n 个像素点中由检测算法检测出的前景的像素点个数为 n_v ，对前景像素点在此超像素中所占比例判断是否小于设定阈值

$$\frac{n_v}{n} < t \quad (5.16)$$

如果比例小于设定阈值 (5.16)，那么可以判定此超像素属于背景超像素子区域，然后将超像素子区域中误检为前景的像素点进行处理。

5.3 实验结果与分析

我们选取动态背景视频中第 48 帧图像为例，图 5.2 给出了原始视频中第 48 帧的原图图像，图 5.3 给出了动态背景下前景目标提取后的示例视频中第 48 帧的图像。

首先，我们采用问题一中所建立的模型进行测试，实验结果如图 5.4 所示，发现该模型无法解决动态背景下的前景目标提取问题，可看出 Vibe 算法对出现变化的像素点较为敏感，对动态幅度较大的水面区域，算法无法分辨出大量波动的水纹与画面中出现的运动目标，导致其水面动态区域几乎完全被判断为前景，无法保证检测结果的准确性，导致实验效果图中出现非常多的噪点。

对于这种结果，我们改进背景模型的建立方法，采用混合高斯模型进行背景模型的建立，之后再进行前景目标的提取，实验结果如图 5.5 所示，可以看出背景噪声有了一定程度的减少，但是对于前景目标的提取并不理想，出现前景目标不能完全提取，只有轮廓信息的现象。

接着，本题提出一种在使用混合高斯模型进行背景建模的基础上，使用改进的 Vibe 算法进行前景目标提取的方法，本题算法改进了 Vibe 算法，继承了 Vibe 算法查全率较高的特点，保证了检测结果的查全率，并且利用建立区分性表观模型的方法进一步抑制了动态幅度较大的水面区域，并在超像素分割的后处理中消除了整个场景中出现的噪声，实验结果如图 5.6 所示，可以看出，本文方法对动态背景干扰因素的抑制优于其他方法。与图 5.3 对比发现，本题所给出的算法能够更有效的提取前景目标的形状。



图 5.2 第 48 帧原图



图 5.3 示例前景目标提取效果图



图 5.4 问题一所建模型提取效果图



图 5.5 混合高斯模型提取效果图



图 5.6 本题算法提取第 48 帧效果

实验过程中提取第 45、46、47、48 帧图像，如图 5.7-5.9 所示，可以看出本题所给出的算法能够有效的提取前景目标。



图 5.7 原始视频中第 45-48 帧图像



图 5.8 原始示例所提取的前景目标第 45-48 帧图像



图 5.9 本题算法所提取的前景目标第 45-48 帧图像

5.4 模型的评价

本题模型的评价检测标准与问题一相同。

表 5.1 评价参数

Review/Active video	Water Surface
Re (Recall)	0.9481
Sp (Specificity)	0.9878
FPR (False Positive Rate)	0.7671
FNR (False Negative Rate)	0.0519
PWC (Percentage of Wrong Classification)	1.4863
Precision	0.8473
F_Measure	0.8949

6 问题三模型建立与求解

6.1 问题描述及分析

本题的目的是在视频发生抖动时，我们能准确提取到前景目标。首先如何去视频的抖动（包括平移、晃动等）是重点，再将稳像后的视频与前面的模型相结合从而提取前景目标。因此，本题的解题过程分为两步，第一步，现将抖动的视频通过设计的模型减轻因抖动带来的视觉影响。第二步，按照问题二采用的算法提取前景目标。

6.2 模型的建立

坐标的变换是一种基于三维或二维的一种参考系转换计算，在建立本模型时，我们采用仿射变换作为基础模型。我们采用改进的 RANSAC 算法，提出了一种高精度鲁棒的基础矩阵估计算法，首先在随机抽样时采用分块技术，并定义匹配点离散度约束机制，保证构成子集的匹配点分布均匀，避免出现集中现象；在判别内外点时，根据对极线距离的统计均值，自适应设定阈值；最后使用 M 估计法，为所有内点构造权重，进一步提高基础矩阵的估计精度。

同时我们也利用灰度投影算法模型与改进的 RANSAC 算法-M 估计法模型做比较，发现该方法优于灰度投影法模型^[1]。

6.2.1 仿射变换

相似变换是一个等距变换与一个均匀缩放的复合。当欧式变换（没有反射）与均匀缩放复合时，相似变换的矩阵表示为：

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (6.1)$$

或分块形式

$$x' = H_s x = \begin{bmatrix} A & t \\ D^T & 1 \end{bmatrix} x \quad (6.2)$$

定义 A 是一个思维的非奇异矩阵。它具有六个自由度，矩阵 A 可以分解为：

$$A = R(\theta)R(-\phi)DR(\theta) \quad (6.3)$$

其中 $R(\theta)$ 和 $R(-\Phi)$ 分别表示转角 θ 和 Φ 的旋转，而 D 为对角矩阵：

$$D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad (6.4)$$

分解公式（6.3）由 SVD 为直接得到：

$$A = UDV^T = (UV^T)(VDV^T) = R(\theta)R(-\phi)DR(\theta) \quad (6.5)$$

其中第一个等式来自 A 的 SVD，其他等式则因为 U 和 V 是正交矩阵。

6.2.2 基础矩阵和 RANSAC 算法模型

设同一场景在不同视角下的两幅图像为 I, I' ，空间点 M 在 I, I' 中的投影匹配点对为 m, m' ，则 m' 必定位于 m 在图像 I' 中的对极线 $l'(l' = Fm)$ 上，即 $m'^T Fm = 0$ ，其中， F 就是秒杀对极集合关系的基础矩阵。

使用齐次坐标表示匹配点分别为 $(x, y, 1)$ ， $(x', y', 1)$ ，则上述等式可以线性表示为 $Uf=0, U=(x'x, x'y, xy', yy', y', x, y, 1) f=(F_{11}, F_{12}, F_{13}, F_{21}, F_{23}, F_{31}, F_{32}, F_{33})^T$

(1) 计算最小采用次数。已知局外点概率为 ε ，每个样本由 S 对匹配点构成， M 个样本中至少有一个样本包含的匹配点对全部都是正确的概率，即置信率为 $P=1-(1-(1-\varepsilon)^S)^M$ ，则采样次数为：

$$M = \frac{\log(1-P)}{\log(1-(1-\varepsilon)^S)} \quad (6.1)$$

(2) 对所有匹配点对做归一化处理，然后随机采样抽取 S 对匹配点构成样本，一般 $S=8$ ，采用改进的八点法计算出基础矩阵 F 。

(3) 利用基础矩阵 F 求出所有匹配点对的对极线距离 d ，与预先设置的阈值 T 比较，当 $d < T$ 时，该匹配点为内点，否则为外点，获得内点集。

$$d = \sqrt{\frac{(m'^T Fm)^2}{(Fm)_1^2 + (Fm)_2^2} + \frac{(m'^T F^T m')^2}{(F^T m')_1^2 + (F^T m')_2^2}} \quad (6.2)$$

其中 $(Fm)_i^2$ 表示矢量 Fm 的第 i 个元素的平方。

(4) 重复步骤 (2) 和 (3) M 次, 获得 M 组基础矩阵和相应的内点集, 取匹配点个数最多的内点集为最优内点集。

(5) 利用改进的八点法对最优内点集进行基础矩阵估计, 并进行规范化处理的逆变换, 得到最终的基础矩阵。

6.2.3 改进的 RANSAC 算法

为了构成样本集即的匹配点分布均匀, 首先将匹配点所在的区域均匀分割为 $b*b$ 块, 去除其中不包含任何点的块, 假设共有 1 块, 然后采用蒙特卡洛法随机选择其中的 S 块, 每块选取一点, 构成样本子集。本文提出一种新的采样策略, 分两步进行, 首先将匹配点所在区域均匀分为 $2*2$ 块, 可知每块中至少一个匹配点, 不存在无匹配点的块。每块中采样个数根据下式求得

$$N_i = \left\lceil S \frac{\sum_{n=1}^i B_n}{\sum_{n=1}^4 B_n} \right\rceil - \left\lceil S \frac{\sum_{n=1}^{i-1} B_n}{\sum_{n=1}^4 B_n} \right\rceil \quad (6.3)$$

式中 N_i 表示第 i 块中应采样的点数, $i=1,2,3,4$, B_n 表示第 n 块中含有匹配点的个数, $n=1, 2, 3, 4$, s 表示一个样本子集中含有的匹配点个数, 符号 $\lceil \cdot \rceil$ 表示四舍五入运算。这样采样所得的样本子集基本保证了分布均匀性, 但是仍存在不符合要求的情况, 为此, 我们引入离散度来对子集进行筛选, 定义离散度为:

$$SD = \sum_{i=1}^m (x_i - x)^2 + (y_i - y)^2 \quad (6.4)$$

式中 (x_i, y_i) 表示第 i 个点的坐标, (x, y) 表示所有匹配点的质心坐标。设所有匹配点的离散度为 β , 采样子集的离散度为 α , 当满足条件: $\alpha > k\beta$ 时, 该子集分布均匀, 所求基础矩阵较稳定, k 为调谐常数。

针对传统 RANSAC 算法中阈值设定的问题为我们借鉴 LMeds 算法的思想, 根据对极线距离的统计均值来自适应设定阈值。具体操作为: 由一组样本子集计算出基础矩阵 F 后, 将 F 作用于所有匹配点, 求出所有点对的对极线距离 d_i 及其均值 $Md = \text{mean}(d_i)$, 进而计算出标准差:

$$\sigma = 1.4826 \left(1 + \frac{5}{n-s}\right) \sqrt{Md} \quad (6.5)$$

其中 n 为所有匹配点的个数, s 为样本中所含匹配点的个数。利用标准偏差来检验匹配点是否为内点, 即:

$$z = \begin{cases} \text{inliers} & d \leq T = 2.5\sigma \\ \text{outlier} & \text{otherwise} \end{cases} \quad (6.6)$$

通过 M 此抽样并检测内点, 得到 M 组内点集, 取匹配点数量最多的内点集作为最优内点集。

6.2.4 M-estimators 法

我们使用 M 估计法, 利用点到对极线距离 d 来构造内点的权重因子, 减少定位误差点的影响, 使用 Huber 给出的一个通用的权重函数, 其定义如下:

$$w_i = \begin{cases} 1 & d \leq \sigma \\ \sigma/d & \sigma < d \leq 3\sigma \\ 0 & 3\sigma < d \end{cases} \quad (6.7)$$

其中 σ 通过公式(6.5)得到。为了提高算法的估计精度，通常采用迭代的方式，逐步减小定位误差对基础矩阵估计精度的影响，具体求取步骤为：

- (1) 利用公式(6.7)计算各点的权值因子 w_i （初始时权值都记为1）；
- (2) 将 $Uf=0$ 乘以 w_i 再用奇异值（SVD）分解求基础矩阵 F ；
- (3) 将求得的基础矩阵 F 加上秩为 2 的约束，返回步骤（1）求出新的 w_i 及 F ，直到相邻两次求得的基础矩阵 F 的误差在设定的阈值范围内。

6.3 结果与分析

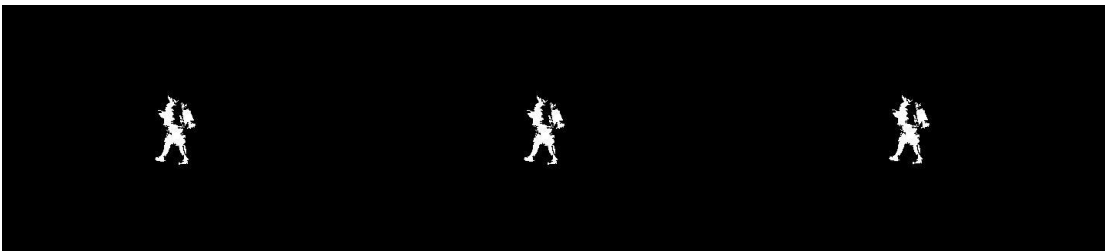
我们将附件 2 中的 people1 作为检测对象，图 6.1 为从原视频中截取的第 6、7、8 帧图像，从视频中能够看出摄像头在轻微抖动拍摄后向右平移了一段距离。因此当我们采用传统的 vibe 算法提取前景目标时，如图 6.2 所示，未能很明显地提取出前景目标，同时噪声较多，说明传统 Vibe 算法并不能去抖动稳像。当我们采用灰度投影法去抖动后再用 Vibe 算法提取前景目标，图 6.3 所示，相较于单独使用 Vibe 算法有很大的改进，但是提取目标的辨识度不是特别高。



图 6.1 有晃动 people1 第 6、7、8 帧原图



6.2 单独地 vibe 算法提取的有抖动前景目标



6.3 采用灰度投影算法去抖动后提取前景目标

7 问题四模型的建立与求解

7.1 问题描述及分析

利用问题一、二、三中所构造的建模方法，考虑静态背景、动态背景下相机晃动与否，从给出的 8 组视频中提取出包含显著前景目标的视频帧，并记录其标号。

本文中对显著前景目标进行空间上的界定，认为该目标只要在视频中出现运动上的差别，我们就将其判定为显著前景目标。

7.2 问题的求解

给定的 8 组视频中并没有发生相机抖动的视频，因此运用问题一问题二所建立的模型进行求解：

Step1: 运用基于混合高斯模型背景建模的 Vibe 算法，对前景目标进行提取；

Step2: 选出具有显著前景目标的参考帧；

Step3: 将参考帧图像转为黑白图像，统计参考帧中显著前景目标所占的面积，并将其设定为阈值常数 T ；

Step4: 遍历所有的视频帧，统计其中前景目标所占的面积 S ，通过对比，若该面积 $S \geq T$ ，则将其判定为显著前景目标，将其所在视频中的帧号记录在矩阵 $frame_num$ 中，并将显著前景出现的总帧数 N_1 增加 1；




Step5: 输出显著目标总帧数 N_1 ；






Step6: 人工统计视频中显著前景目标出现的帧数 N_2 并记录，作为参考；

Step7: 计算 STER (Significant Target Extraction Rate)，即显著目标提取率)， $STER = N_1/N_2$ 。

7.3 实验结果与分析

表 7.1 实验结果记录表

Scene	预测帧数 (N_1)	实际帧数 (N_2)	STER	提取出显著目标所在帧号
 Office	1438	1459	98.56%	197, 372, 501, 592~2025, 2080
 Fountain	172	201	85.57%	141, 158~210, 259, 335, 409~523
 Lobby	511	537	95.16%	79,156~194, 259, 347~391,624~666, 870, 964~1036,1241~1282,

 Campus	461	494	93.32%	205~218,311~517, 660~669,695~707, 746~899,1013~1028, 1335~1349,1353~1369,1383~ 1397
 Curtain	418	454	92.07%	967, 1764~1897, 2126, 2178~2308, 2772~2922
 Escalator	3016	3069	98.27%	1~166, 173~2383, 2415, 2780~3417
 Hall	2700	2779	97.16%	1~508,578,604~733, 795,858~1048,1138, 1161~1208,1246, 1282~1505,1559~2580,2725~ 2772,2817~2867,2921~2960,3 033~3348,3417~3534
 Overpass	222	253	87.75%	374,968,1551, 2398~2616

从以上实验结果可以看出，本文针对问题一问题二提出的前景目标提取模型具有较高的显著目标提取率，说明了本文所给出的模型能够较为有效的提取前景目标及其所在帧数，实验取得了预期效果。

8、问题五模型建立与求解

8.1 问题描述及分析

本题提出在不同角度多摄像头的条件下如何有效检测和提取前景目标。本题的关键在于如何从单摄像头追踪检测应用到多摄像头多目标检测提取，不仅要考虑到单摄像头的运动目标跟踪，当运动目标因角度问题发生重叠时如何有效的区分开来也是一个重要的问题，而且多摄像头跟踪及协同通信都是面临解决的。

8.2 模型建立

在我们前面建立的混合高斯模型解决的动态背景前景目标提取的前提下。对于多摄像头下的信息融合我们采用单应性约束法，对目标进行定位后再采用粒子滤波法对目标进行跟踪，再将多个摄像头提取的目标进行协同^[5]。

8.2.1 基于单应性约束的目标定位

图 8.1 为两个平面中的点对投影示意图。假设世界坐标系的原点为 0，摄像机坐标系原点为 C，从右边的平面中的点 p_0 投影到摄像机成像面的点 p_c ，即是一次透视投影变换过程。

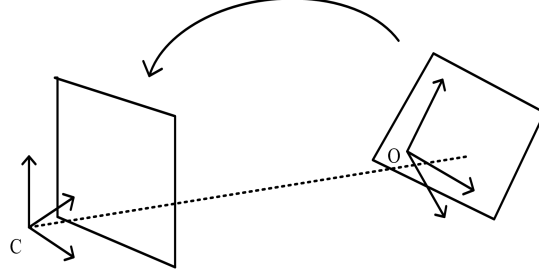


图 8.1 平面投影示意图

在三维空间中，旋转可看成是分别绕 x, y, z 轴进行了 Ψ, φ, θ 角度的旋转，总的旋转可以通过连乘三个单旋转矩阵得到

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & \sin \psi \\ 0 & -\sin \psi & \cos \psi \end{pmatrix} * \begin{pmatrix} \cos \varphi & 0 & -\sin \varphi \\ 0 & 1 & 0 \\ \sin \varphi & 0 & \cos \varphi \end{pmatrix} * \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (8.2)$$

设成像画面的点的齐次坐标有 $p_c = (x, y, 1)$,

$$p_c = K(R + tn^T)p_o = H_{o_c}p_o \quad (8.3)$$

其中 t 为两个坐标系的平移向量，而 n 示该平面在空间坐标系下的法向量，有 $n^T p_o = 1$ ， H_{o_c} 表示从空间平面点到成像面的点的投影矩阵。

如图 8.2 所示，当两个摄像头 c_1, c_2 分别对世界坐标系的某个平面中的同一点 p_o 进行成像投影是，在各自地成像面上得到对应点 p_{c1}, p_{c2} 。

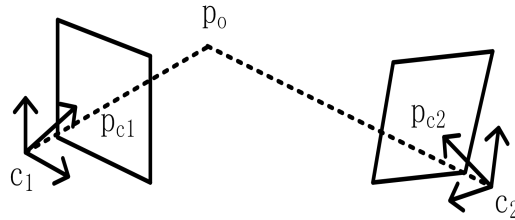


图 8.2 两个视角成像面的投影变换

由上述可知 $p_{c1} = H_1 p_o$ ， $p_{c2} = H_2 p_o$ ，于是可得从 p_{c1} 到 p_{c2} 的变换为

$$p_{c2} = H_2 H_1^{-1} p_{c1} = H_{3 \times 3} p_{c1} \quad (8.4)$$

H 即为单应性矩阵。再运用单应性约束进行目标定位。

8.2.2 粒子滤波目标跟踪

粒子滤波是根据运动目标状态的先验以及系统的状态转换函数，并在时间和观测数据两个方面用贝叶斯滤波器进行更新迭代，并在此基础上不断预测并修正状态的后验概率密度，最后通过求取后验概率密度分布的最优解获取运动目标的状态特征值^[6]。

在目标的检测基础上得到运动目标 X_{target} ，并计算颜色分布 $\{q^{(u)}\}_{u=1,2,\dots,B}$ ，并建立初始样本集 $\left\{X_0^{(i)}, \frac{1}{N}\right\}_{i=1}^N$ 。根据随机漂移模型计算粒子状态转移 $X_{t-1}^{(i)} \rightarrow \tilde{X}_t^{(i)}$ ，计算粒子 $\tilde{X}_t^{(i)}$ 的权重 $w_t^{(i)}$ ，并对其进行归一化处理 $w_t^{(i)} = w_t^{(i)} / \sum_{i=1}^N w_t^{(i)}$ 。估计目标状态 $\hat{X}_t = \sum_{i=1}^N w_t^{(i)} \tilde{X}_t^{(i)}$ 再根据粒子的权重 $w_t^{(i)}$ 从样本集 $\{\tilde{X}_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ 重新抽取 N 个粒子。

8.2.3 多摄像头协同跟踪

通过单应性变换，各个角度的成像画面都投影到参考成像面上，在投影过程中，将整个成像面的前景画面进行投影变换并叠加，得到二值化足部区域图，再分别对每个角度及二值化图的单跟踪结果自动分配跟踪置信度^[7]，设定用 ω_{c_n} 表示目标第 n 个视角的置信度，用 ω_f 表示其对应的足部区域的置信度。在某一视角或足部区域得到了跟踪结果为 n ，表示该目标在该视角与上一帧第 n 个目标相匹配。根据上述权值，综合计算每个目标在各视角和足部区域跟踪结果。

该模型的流程图如图 8.3 所示

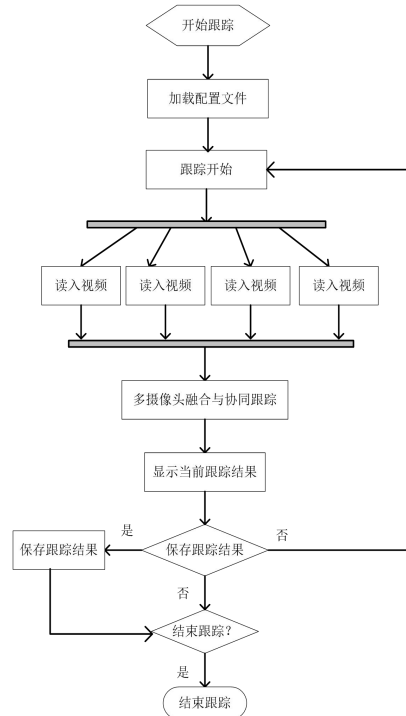


图 8.3 系统主要工作流程图

8.3 实验结果与分析

我们将下载四个摄像头多人运动视频作为检测对象。如图 8.4 所示，这是

四个角度的单人运动，我们采用蓝色的边框作为跟踪工具，并提取出前景目标；随着时间推移，又出现第二人，如图 8.5 所示，我们分别采用蓝色与红色边框跟踪两人运动，而且在不同角度的摄像头中两人身影发生重叠现象时，也能很好地分辨出来，如图 8.6 所示；如图 8.7 所示，三人在两角度摄像头中也能较好的进行前景目标跟踪与提取。

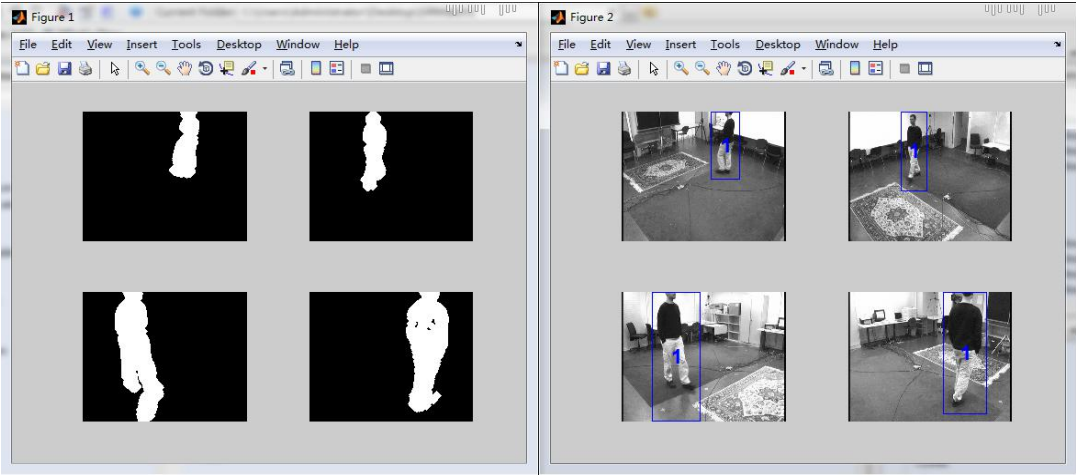


图 8.4 单人四角度跟踪与目标提取

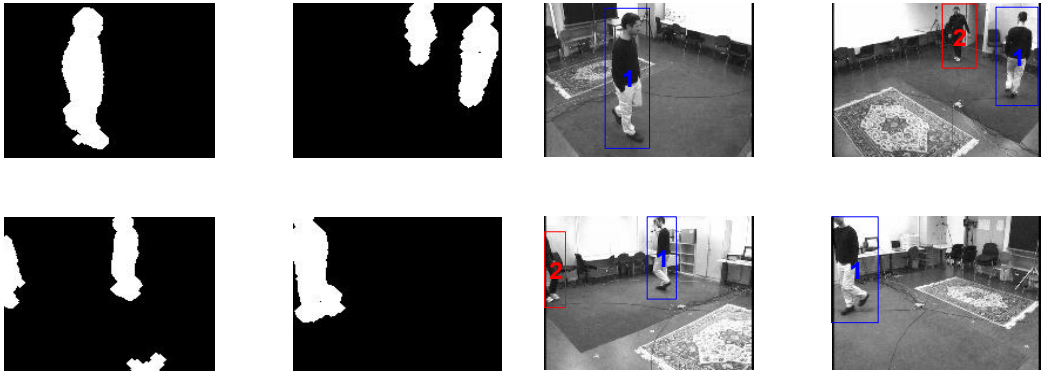


图 8.5 双人四角度目标跟踪与提取

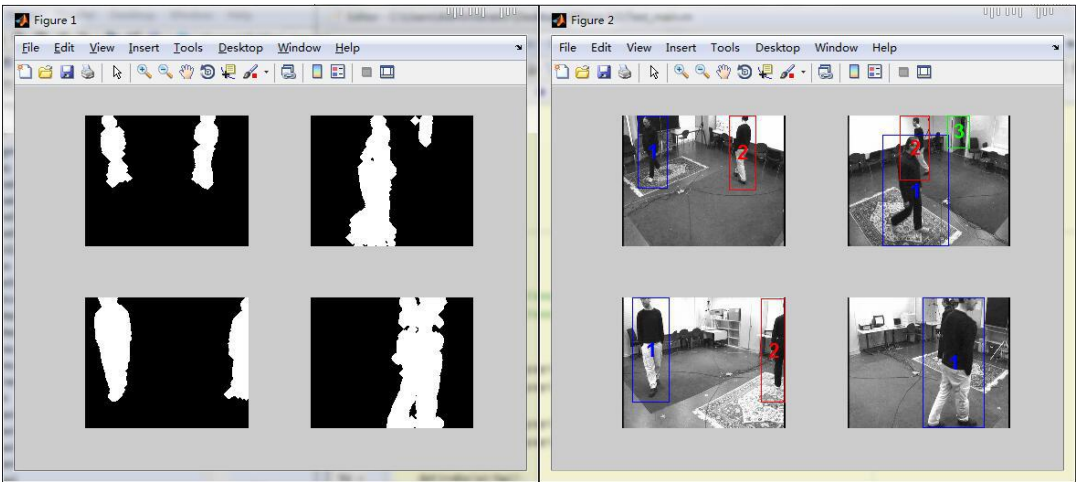


图 8.6 双人四角度目标跟踪与提取（人影重叠）

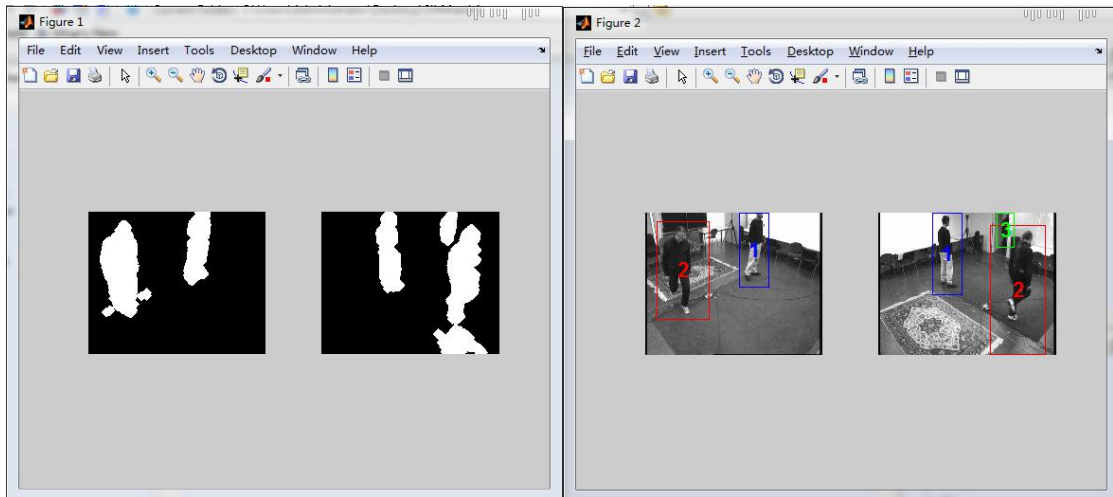


图 8.7 三人两角度目标跟踪与提取

9 问题六

9.1 问题描述及分析

让智能算法与传统的监控系统相结合必将成为未来监控发展的主流，使其能够在没有人为干预的情况下自动分析出监控场景中的事件，一旦发生异常事件就能立刻发出报警信号或者提供有价值的信息，从而有效地提高了部门的应急响应速度。

利用所获取的前景目标的运动信息，判断目标所在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（如跳舞、列队排练等）、物体爆炸、建筑物倒塌等异常事件。本文提出了一种基于混合高斯和稀疏表示的模型，引入混合高斯模型用于学习不同类型的运动特征规律，然后通过各个单高斯模型中的均值建立一个相似矩阵作为字典。训练阶段获取混合高斯模型中的均值矩阵并对其建立相似矩阵作为字典。测试阶段，生成局部特征的核矢量并以该核矢量计算基于稀疏表示的重构误差，并将其与已设定的阈值进行比较，如果重构误差大于阈值，则判为异常。

9.2 模型的建立与求解

本文提出了一种基于混合高斯和稀疏表示的模型，异常事件检测系统的基本流程：

训练阶段具体步骤：首先对输入训练视频序列分块并提取光流信息，以光流信息表示块的局部特征，然后采用混合高斯模型表达不同类型的运动特征规律，最后由各个单高斯模型中的均值组成一个均值矩阵并对其建立相似矩阵来作为字典。

测试阶段具体步骤：对输入的测试视频序列分块并提取光流信息，以光流信息表示块的局部特征，求出块的局部特征与均值矩阵中每一列的核函数值，以此建立核矢量，然后采用 OMP 算法求出它的稀疏表示并计算出重构误差值，最后将其与已设定的阈值进行比较，如果重构误差大于阈值，则判为异常。本文的算

法分为离线训练与在线测试两部分，具体流程如图 9.1 与图 9.2 所示：

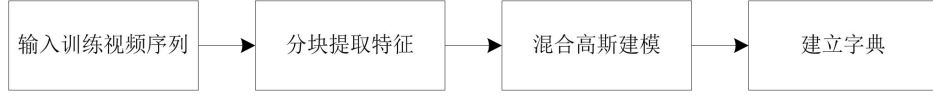


图 9.1 训练过程工作流程图

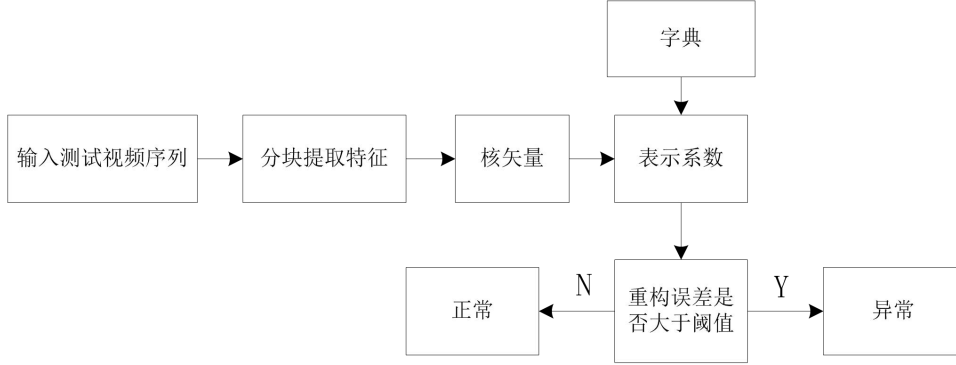


图 9.2 测试过程工作流程图

9.2.1 混合高斯模型

单高斯模型是指使用一个高斯函数去拟合样本空间中各数据点的概率密度分布情况，并构建出对应的概率数学模型。而混合高斯模型是指由多个单高斯函数加权组合而成，相比单高斯模型能更准确地拟合数据点的多个分布，目前普遍地应用在语音识别、人体特征检测、目标跟踪等领域并取得了良好的效果。

(1) 特征提取

将视频段中每一帧分割成互不重叠且大小为 $N \times N$ 的块，并连续取 M 帧，得到大小为 $N \times N \times M$ 的立方体块，每个立方体块称为原子。设视频帧的分辨率大小为 $W \times H$ ，每一帧得到块的数量 $f_block = \lfloor W/N \rfloor \times \lfloor H/N \rfloor$ ，其中 $\lfloor \cdot \rfloor$ 表示向下取整。 t 时刻位置 $loci$ 上块的运动信息用一个直方图表示

$$h_{locit} = [h_1, h_2, h_3, h_4], 1 \leq loci \leq f_block$$

其中 h_{locit} 中的各个分量是根据光流方向按 90 度间隔量化获得的 4 个方向上的光流幅值之和。结合历史时刻的直方图信息， t 时刻位置 $loci$ 上的原子可以表示为 $x_{locit} = [h_{locit-(M-1)} \dots h_{locit}]^T, 1 \leq loci \leq f_block$ 。取一段视频，以 M 帧为单位分成 P 小段，得到原子数量为 $P \times f_block$ ，由这些原子构成原子集合。

(2) 特征学习

训练阶段，采用混合高斯模型表示不同类型的运动特征规律。采用谱聚类算法对训练数据进行聚类，可以得到每一类局部特征的类中心和协方差，并用一个单高斯模型对这一类局部特征进行描述。前面所述混合高斯模型就是由若干个单高斯模型的加权概率和，其中每个单高斯模型的权重值由式 (9.3) 的 $w_{k,loci}$ 给出。下文将给出建立混合高斯模型的算法步骤。

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N_k} x_i \quad (9.1)$$

$$\sigma_k = \frac{1}{N_k} \sum_{i=1}^{N_k} (x_i - \mu_k)^2 \quad (9.2)$$

$$w_{k,loci} = \frac{n_{k,loci}}{\sum_{\varepsilon} n_{\varepsilon,loci}}, \varepsilon = \{1, 2, \dots, Num\} \quad (9.3)$$

Num 表示高斯分布总数； N_k 表示属于第 k 类单高斯的局部特征样本个数； $n_{k,loci}$ 表示位置 $loci$ 上属于第 k 类单高斯的局部特征样本个数， $1 \leq loci \leq f_block$ 。在不同的空间位置上，第 k 类单高斯分量的权重值不同。

在位置上建立混合高斯检测模型如下：

(a) 建立初始混合高斯模型，对 (1) 提取出的样本集合进行谱聚类得到 $Num=Num_0$ 类单高斯，根据式 (9.1)、式 (9.2) 以及式 (9.3) 可以求得参数 $\theta = \{w_{k,loci}, \mu_k, \Sigma_k = diag(\sigma_k)\}$ ，同时保存 $n_{k,loci}$ ， $0 < k \leq Num$ ，和每一类局部特征的训练数据。

(b) 输入新的训练数据 x_{loci}^{new} ，将它与位置 $loci$ 上混合高斯模型进行比较，若满足 $\log(p(x_{loci}^{new} | \theta)) < threshold$ 时，将 x_{loci}^{new} 加入到一个新的集合 U 中，否则把 x_{loci}^{new} 加入到位置 $loci$ 上最相似类局部特征的训练数据中，假定 $index$ 类局部特征最相似，则对 $n_{index,loci}$ 加 1，并将 $\theta = \{w_{k,loci}, \mu_k, \Sigma_k\}$ 按式 (9.1)、式 (9.2) 以及式 (9.3) 更新；其中 $\log(p(x_{loci}^{new} | \theta))$ 是按式 (9.4) 计算的对数似然函数估计值；若训练视频全部输入完毕，则终止整个循环，训练结束。

$$\log(p(x_{loci}^{new} | \theta)) = \log\left(\frac{1}{\sqrt{(2\pi)^{4M}}} \sum_{k=1}^{Num} w_{k,loci} \frac{1}{\sqrt{\det(\Sigma_k)}} \exp\left(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)\right)\right) \quad (9.4)$$

(c) 当集合 U 中的数据数量未达到 Q 时，重复步骤 (b)；当步骤 (b) 中的集合 U 中的数量达到 Q 时，重新对集合 U 进行谱聚类，聚类成 Num_1 类，此时又新增了 Num_1 个单高斯，单高斯的类目总数变为 $Num=Num+Num_1$ ，更新 $\theta = \{w_{k,loci}, \mu_k, \Sigma_k\}$ ， $0 < k \leq Num$ ，清空集合 U ，继续重复步骤 (b)。

9.2.2 生成字典

训练结束后，将每个单高斯模型中的均值参数组成集合 $A = \{\mu_1, \mu_2, \mu_3, \dots, \mu_{Num}\}$ ，再建立字典 $\Phi \in R^{Num \times Num}$ ：

$$\phi_{ij} = \exp(-(1 - d_f(\mu_i, \mu_j))) \quad (9.5)$$

其中 $d_f(\mu_i, \mu_j)$ 为 μ_i 与 μ_j 的直方图交距离。

9.2.3 视频异常事件检测

输入测试数据 x^{test} ，求出它与每个单高斯模型中的均值矢量的核函数值，并组成它的一个核矢量 $y^{test} = [y_1, y_2, \dots, y_j, \dots, y_{Num}]^T$ ，其中 y_j 的表达式为：

$$y_j = \exp(-(1 - d_f(x^{test}, \mu_j))) \quad (9.6)$$

对 y^{test} 的稀疏重建模型可表示为:

$$\min \|s\|_0, s.t. \|\Phi s - y^{test}\|_2^2 \leq \varepsilon \quad (9.7)$$

求解上式的非凸优化问题, 可以采用 OMP 算法, 其原理是在每一步迭代过程中, 将信号投影到由所有被选基生成的子空间上, 并更新所选基的系数, 以使得生成的残差与被选基都正交。因此在迭代中根据所选的基, 利用最小二乘的方法确定系数矢量的元素。采用 OMP 算法求得 s 代到式 (9.8) 中, 若 $err < th$, 则测试数据 x^{test} 被判为正常事件, 否则被判为异常事件。整个测试过程可以见算法 9.2。

$$err = \|y^{test} - \Phi s\|_2 \quad (9.8)$$

算法 9.2:

输入: 从测试视频帧中提取出局部运动特征集合

$X = [x_1, x_2, \dots, x_l, \dots, x_{Ne}], x_l \in R^n$, 训练阶段学习得到的每类单高斯的均值

$\mu_1, \mu_2, \dots, \mu_{Num}$ 和对应的合成字典 Φ 以及阈值 th ;

输出: 测试样本异常标志 $F \in [f_1, f_2, \dots, f_l, \dots, f_{Ne}], f_l \in \{0, 1\}$, 0: 正常; 1: 异常;

初始化: 令 $j=1$;

循环执行如下步骤 1-3

1: 根据公式 (9.6) 求出 x_j 与 $\mu_1, \mu_2, \dots, \mu_{Num}$ 的核函数值, 并组成一个核矢量 $y_j = [y_1, y_2, \dots, y_{Num}]^T$ 。稀疏矢量的初始值为 $s=0$, 信号表示的允许误差 $E1$, 残差初始化为 $r=y_n$, 所选基序号的集合 Ω 初始化为空集, 迭代次数 $t=1$ 。

迭代过程:

1) 选择基: 选择与上一步残差最匹配的基, 即所选择的基满足

$$\xi = \arg \max |\langle \phi_i, r \rangle|;$$

2) 更新所选基: 所选基序号的集合为 $\Omega = \Omega \cup \xi$;

3) 更新稀疏矢量: 根据最小二乘方法得 $s_\Omega = (\Phi_\Omega^H \Phi_\Omega)^{-1} \Phi_\Omega^H x, s_\Omega = 0$;

4) 更新残差: $r = x - \Phi s$;

5) 若 $t > Num$ 或者参差值 $\|r\|^2 < n * E1^2$, 转至步骤 6); 否则 $t=t+1$, 转至步骤 1);

6) 迭代结束, 输出信号 x 关于字典 ϕ 的编码 s 。

2: 计算 $f_j = (\|\Phi s - y_j\|_2 \leq th ? 0 : 1)$;

3: $j=j+1$

直到: $j > Ne$

9.3 实验结果与分析

本题采用 UMN 数据集中的人群异常行为视频进行检测, UMN 数据集包括了 3 个不同的场景分别为草坪、大厅、广场。如图 9.3、图 9.4、图 9.5 所示的正

常行走事件和异常逃散事件，其中场景为草坪的数据集总共有 1450 帧，场景为大厅的数据集总共有 4415 帧，场景为广场的数据集总共有 2145 帧。各个场景的训练数据来自于相应数据集的前 500 帧图像，其余图像作为测试数据。

当视频中出现异常逃散事件时，用文字“Abnormal Crowd Activity”标示在图片中，用以警示。由图 9.3-9.5 可知，本题算法可以有效检测出场景中发生逃散、慌乱等群体异常事件。

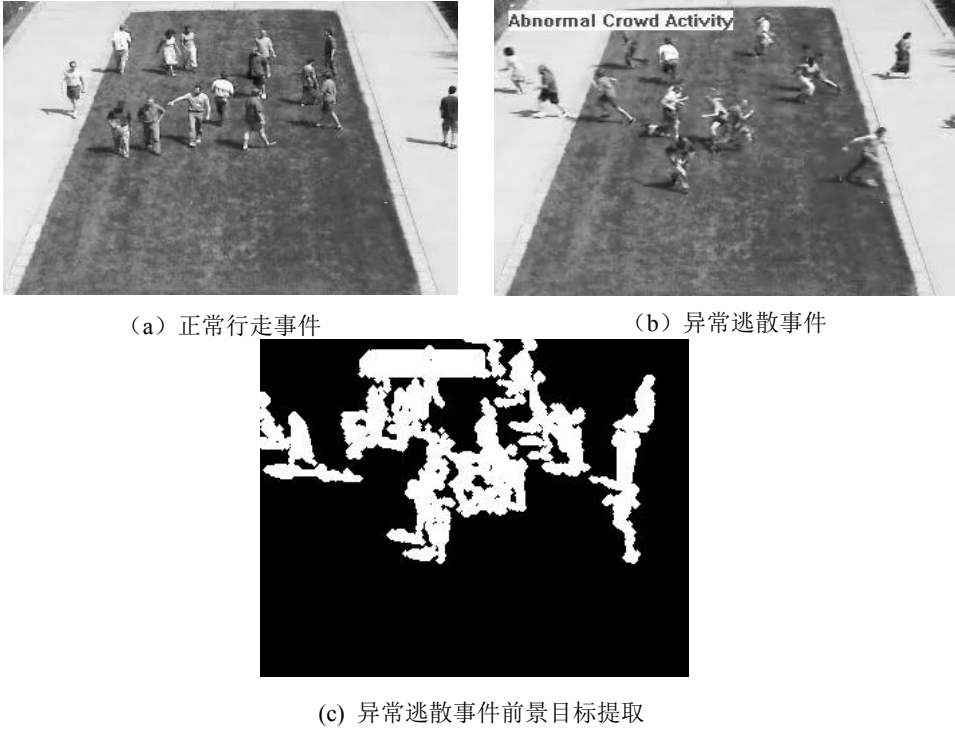


图 9.3 UMN 数据集中的草坪场

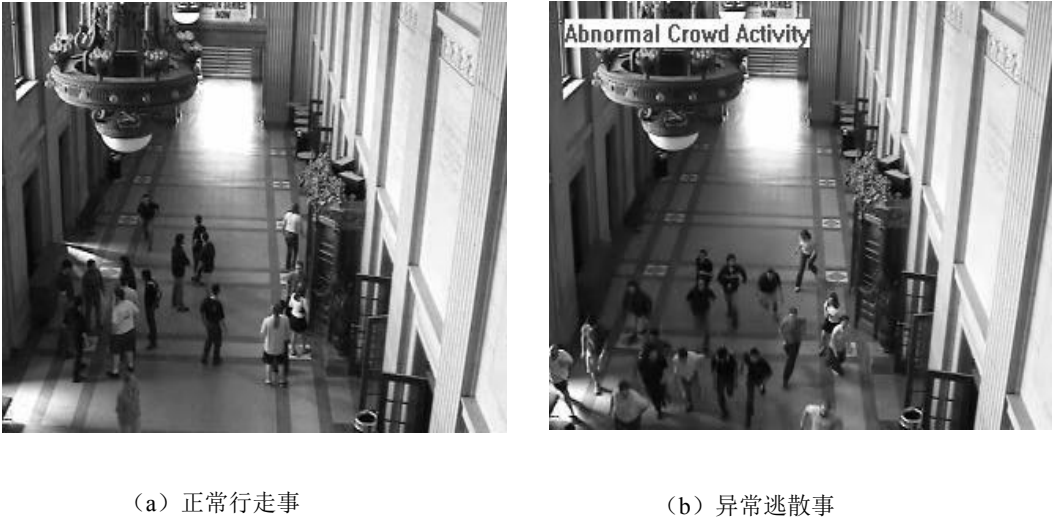


图 9.4 UMN 数据集中的大厅场



(a) 正常行走事

(b) 异常逃散事

图 9.5 UMN 数据集中的广场

10 模型的评价

10.1 模型的优点

问题一中，采用改进的 Vibe 算法能较好的抑制因提取前景目标而引起的鬼影，去噪方面也得到了优化。

问题二中，本文使用混合高斯模型建立背景模型，采用改进的 ViBe 算法，继承了 ViBe 算法查全率较高的特点，保证了检测结果的查全率，并且利用建立区分性表观模型的方法进一步抑制了动态幅度较大的水面区域，并在超像素分割的后处理中消除了整个场景中出现的噪声，本文方法对动态背景干扰因素有很好的抑制作用。

问题三中，采用基于仿射变换模型的改进的 RANSAC 算法，能有效地消除视频抖动对目标提取的影响，而且经过特征点提取与匹配后，使得视频较之前晃动减小，利于提取目标。

问题四中，提出了一种基于概率学的统计算法，针对问题一问题二提出的前景目标提取模型具有较高的显著目标提取率，说明了本文所给出的模型能够较为有效的提取前景目标及其所在帧数。

问题五中，单应性约束对目标能很好的对目标进行定位，而粒子滤波能较好的跟踪目标。引入了多相机协同工作的相机视觉融合。

问题六中，本文提出了一种基于混合高斯和稀疏表示的模型，引入混合高斯模型用于学习不同类型的运动特征规律，然后通过各个单高斯模型中的均值建立一个相似矩阵作为字典，用以判断是否为异常事件。当视频中出现异常逃散事件时，用文字“Abnormal Crowd Activity”标示在图片中，用以警示。本题算法可以有效检测出场景中发生逃散、慌乱等群体异常事件。

10.2 模型的缺点

对于问题一中，没有考虑到实际情况中光影变化的问题，因此建立的模型在应用上具有局限性。

对于问题二中，对于视野远处缓慢行驶的车或船无法准确判断其为前景目标。

对于问题三中, 尽管减小了晃动, 但是对于提取目标来说仍然具有噪声不能完全抑制以及目标有孔洞的问题, 因此算法还有待进一步改进。

对于问题四中, 由于所给视频中并未出现相机抖动的现象, 对于相机抖动的场景中是否能够有效的提取前景目标及其所在帧号, 我们并没有进行实验。需要进一步的实验检测。

对于问题五中, 尽管两目标重合时能较好地分辨出两个目标并能提取出来, 但是对于 3 个乃至多个目标出现重叠时, 目标定位于跟踪便会出现混乱与延迟的现象。对于粒子滤波算法还需要改进。

对于问题六中, 现实中可能发生的异常事件具有复杂性, 如人群的逃散和建筑物的倒塌, 建立的模型可能不具有普适性, 由于时间限制, 我们只研究了人群逃散、慌乱这一类异常事件, 其他的异常事件还有待研究。

参考文献

- [1]陶声祥, 周亚军, 张江辉,等. 基于行灰度投影相关的快速角度矢量估计算法[J]. 光电工程, 2008, 35(11):91-95.
- [2]Barnich O, Van D M. ViBe: a universal background subtraction algorithm for video sequences.[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2011, 20(6):1709.
- [3]吴尔杰, 杨艳芳, 田中贺,等. 一种能快速抑制鬼影及静止目标的 ViBe 改进算法[J]. 合肥工业大学学报:自然科学版, 2016, 39(1):56-61.
- [4]甘玲, 赵华翔. 一种改进的 ViBe 算法结合多特征融合的阴影移除方法[J]. 微电子学与计算机, 2015(11):152-157.
- [5]徐宇. 多摄像机全景协同的运动目标跟踪算法研究[D]. 西华大学, 2014.
- [6]葛鹤银, 孙建红, 林楠,等. 融合小波变换及 SIFT 算法的去抖动运动目标检测[J]. 实验室研究与探索, 2016, 35(2):119-123.
- [7]Hess R, Fern A. Discriminatively trained particle filters for complex multi-object tracking[C]// Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009:240-247.
- [8]李亚伟, 曹凯, 王杰,等. 一种改进的 ViBe 鬼影抑制算法[J]. 广西大学学报(自然科学版), 2017, 42(2):712-719.
- [9]叶思;视频监控中多摄像头下的行人跟踪[D];华中科技大学;2012 年

附录

1. 静态背景

Vibe.m 文件

%%Time: 2017.9.17

%%Name: Michael Beechan(陈兵)

%%School: Chongqing university of technology

clear,clc;

% [filename,pathname] = uigetfile('*.avi','choose the video name:');%选择视频

video = mmReader('input.avi');

height = video.Height;

width = video.Width;

tom = zeros(height,width);

%参数

NumFrames = video.NumberOfFrames;

cardinality = 2;%基数

r = 20;%给定半径

n = 20;

%

%初始化 %%%取前 20 帧作为模型初始化

sample = zeros(height,width,n);

for nn = 1 : 15

imrgb = read(video,nn);

imgray = rgb2gray(imrgb);

video_dis = imgray;

if(mod(nn,2)==0)

sample(:, :, nn/2) = imgray;

end

end

bg = padarray(sample,[1 1],'replicate');

%随机跟新

%sd = zeros(height,width);

%r = zeros(height,width);

for f = 16 : NumFrames

imageRGB = read(video,f);

imageRGB1 = read(video,f-1);

imageRGB2 = read(video,f-2);

imageRGB3 = read(video,f-3);

imageRGB4 = read(video,f-4);

imageRGB5 = read(video,f-5);

imageRGB6 = read(video,f-6);

```

imageRGB7 = read(video,f-7);
imageRGB8 = read(video,f-8);

image = rgb2gray(imageRGB);
image1 = rgb2gray(imageRGB1);
image2 = rgb2gray(imageRGB2);
image3 = rgb2gray(imageRGB3);
image4 = rgb2gray(imageRGB4);
image5 = rgb2gray(imageRGB5);
image6 = rgb2gray(imageRGB6);
image7 = rgb2gray(imageRGB7);
image8 = rgb2gray(imageRGB8);
for i = 1:height
    for j = 1:width
        div = abs(sample(i,j,:) - double(image(i,j)));
        ma = find(div == (max(max(max(div)))));
        logic = div < r;
        bignum = sum(logic);

        if bignum > cardinality    %确认该点为背景
            video_dis(i,j) = 0;    %该点显示灰度为 0
            tom(i,j) = 0;
            %    随机选择初始化背景中的一个点用新点进行替换
            randz = randi(16);
            if(randz == 10)    %确定为背景点后 有 1/16 的概率更新背景样本和领域样本
                rands = randi(20);
                sample(i,j,ma) = image(i,j);    %要更新背景样本时从 20 个样本中选取差值最大的那个
更新
%                随机改变(i,j)某个背景邻域像素点的某个样本值
                randy = round(rand)*2-1;%产生 0-1 之间的数，四舍五入为 0 或 1，然后乘以 2，减去 1，
得到的不是-1 就是 1
                randx = round(rand)*2-1;
                sample(i+1+randy,j+1+randx,rands) = image(i,j);
            end
        else
            video_dis(i,j) = 255;
            tom(i,j) = tom(i,j)+1;
            if(tom(i,j) > 5)    %当前像素被判定为前景点超过一定次数后，用当前像素点替换到样本集中的某个样本
                a1=abs(double(image(i,j)) - double(image1(i,j)));
                a2=abs(double(image(i,j)) - double(image2(i,j)));
                a3=abs(double(image(i,j)) - double(image3(i,j)));
                a4=abs(double(image(i,j)) - double(image4(i,j)));
                a5=abs(double(image(i,j)) - double(image5(i,j)));

```

```

a6=abs(double(image(i,j)) - double(image6(i,j)));
a7=abs(double(image(i,j)) - double(image7(i,j)));
a8=abs(double(image(i,j)) - double(image8(i,j)));
s=0;
if(a1 < 5)
    s=s+1;
else %s=s;
end
if(a2 < 5)
    s=s+1;
else %s=s;
end
if(a3 < 5)
    s=s+1;
else %s=s;
end
if(a4 < 5)
    s=s+1;
else %s=s;
end
if(a5 < 5)
    s=s+1;
else %s=s;
end
if(a6 < 5)
    s=s+1;
else %s=s;
end
if(a7 < 5)
    s=s+1;
else %s=s;
end
if(a8 < 5)
    s=s+1;
else %s=s;
end
if(s>6) %当前帧某像素点和前 3 帧该位置的值相差不大, 说明该点是背景
    randk = randi(20);
    sample(i,j,randk) = image(i,j);
    tom(i,j)=0;
    video_dis(i,j) = 0;
else
    video_dis(i,j) = 255;
end

```

```

                else
                    video_dis(i,j) = 255;
                    tom(i,j) = tom(i,j)+1;
                end
            end
        end
    end

    end

    randbg = randi(n);
    out = bg(2:height+1,2:width+1,randbg);
    video_dis=imfill(video_dis,'holes');%将原图填充孔洞
    %video_dis=imreconstruct(imerode(video_dis,strel('ball',5,1)),video_dis);
    frameNum = 270 + f;    %这里的 0 表示截取的开始帧数
    path='D:\数学建模代码\问题 1\vibe_xiao_gui_ying\vibe_xiao_gui_ying';
    imwrite(video_dis, strcat(path,int2str(frameNum),'.jpg'));
    figure(1),subplot(1,2,1),imshow(image,[]);title(sprintf('第%d 帧视频', f), 'FontWeight', 'Bold', 'Color', 'r');
    subplot(1,2,2),imshow(out,[]);title(sprintf('第%d 帧背景', f), 'FontWeight', 'Bold', 'Color', 'r');
    subplot(1,2,2),imshow(video_dis,[]);title(sprintf('第%d 帧目标', f), 'FontWeight', 'Bold', 'Color', 'r');
    %%drawnow;
end

```

2.动态背景

gaosi.m

%%Time: 2017.9.17

%%Name:Michael Beechan(陈兵)

%%School:Chongqing university of technology

% mixture of Gaussians algorithm for background

%混合高斯模型适用于相机固定的运动目标检测，光流法适用于相机运动的运动目标检测

close all;

clear all;

source=dir('*.jpg');

% ----- frame size variables -----

% read in 1st frame as background frame

fr_bw=imread(source(1).name);

%fr_bw = rgb2gray(fr); % convert background to greyscale

fr_size = size(fr_bw);

width = fr_size(2);

height = fr_size(1);

fg = zeros(height, width);

bg_bw = zeros(height, width);

% ----- mog variables -----

```

C = 3; % number of gaussian components (typically 3-5)
Cgegaosimoxing
M = 3; % number of background components,
D = 2.5; % positive deviation threshold
alpha = 0.01; % learning rate (between 0 and 1) (from paper 0.01) the background
will change slowly with the time, so the mean(u)should be updated slowly.
thresh = 0.25; % foreground threshold (0.25 or 0.75 in paper)
sd_init = 6; % initial standard deviation (for new components) var = 36 in paper
w = zeros(height,width,C); % initialize weights array
mean = zeros(height,width,C); % pixel means , the u of gaosi(u,d)
sd = zeros(height,width,C); % pixel standard deviations , the d of gaosi(u,d)
u_diff = zeros(height,width,C); % difference of each pixel from mean
p = alpha/(1/C); % initial p variable (used to update mean and sd)
rank = zeros(1,C); % rank of components (w/sd)

% ----- initialize component means and weights -----
pixel_depth = 8; % 8-bit resolution
pixel_range = 2^pixel_depth -1; % pixel range (# of possible values)

for i=1:height
    for j=1:width
        for k=1:C

            mean(i,j,k) = rand*pixel_range; % means random (0-255)
            w(i,j,k) = 1/C; % weights uniformly dist
            sd(i,j,k) = sd_init; % initialize to sd_init

        end
    end
end

for n = 8:(length(source)-2) %there will be false route line,so it only include the pictures which
has the car

    fr_bw = imread(source(n).name); % read in frame
    %fr_bw = rgb2gray(fr); % convert frame to grayscale

    % calculate difference of pixel values from mean
    for m=1:C
        u_diff(:, :,m) = abs(double(fr_bw) - double(mean(:, :,m)));
    end
    sum_x=0;

```

```

sum_y=0;
num=0;
% update gaussian components for each pixel
for i=1:height %%%%%%%%%%%%%%%search each pixal of one image, if it is
in the C ge gaosimoxing, it is belong to the background,and undate the background.
    for j=1:width %%%%%%%%%%%%%% If it is not in the C ge gaosimoxing, create a
new gaosi and replace the least possible gaosi. Finally the the first several gaosi is background, and the last several
is foreground

        match = 0;

        for k=1:C
            if (abs(u_diff(i,j,k)) <= D*sd(i,j,k)) % pixel matches component

                match = 1; % variable to signal component match

                % update weights, mean, sd, p
                w(i,j,k) = (1-alpha)*w(i,j,k) + alpha;
                p = alpha/w(i,j,k);
                mean(i,j,k) = (1-p)*mean(i,j,k) + p*double(fr_bw(i,j));
                sd(i,j,k) = sqrt((1-p)*(sd(i,j,k)^2) + p*((double(fr_bw(i,j)) - mean(i,j,k))^2);
            else % pixel doesn't match component
                w(i,j,k) = (1-alpha)*w(i,j,k); % weight slighly decreases
            end
        end
        end
        bg_bw(i,j)=0;
        for k=1:C
            bg_bw(i,j) = bg_bw(i,j)+ mean(i,j,k)*w(i,j,k);
        end

        % if no components match, create new component
        if (match == 0)
            [min_w, min_w_index] = min(w(i,j,:));
            mean(i,j,min_w_index) = double(fr_bw(i,j));
            sd(i,j,min_w_index) = sd_init;
        end

        rank = w(i,j,:)./sd(i,j,:); % calculate component rank
        rank_ind = [1:1:C];

```

```

        % calculate foreground
        fg(i,j) = 0;
        while ((match == 0)&&(k<=M))

            if (abs(u_diff(i,j,rank_ind(k))) <= D*sd(i,j,rank_ind(k)))
                fg(i,j) = 0; %black = 0

            else
                fg(i,j) = fr_bw(i,j);
                sum_x=sum_x+j;
                sum_y=sum_y+i;
                num=num+1;
            end
            k = k+1;
        end

    end

end

if n==8||n==9||n==10
    route_x=[round(sum_x/num),round(sum_x/num)];
    route_y=[round(sum_y/num),round(sum_y/num)];
else

    next_x=round(sum_x/num);
    next_y=round(sum_y/num);
    route_x=[route_x,next_x];
    route_y=[route_y,next_y];
end
num=0;
figure(1);
%subplot(3,1,1);
imshow(fr_bw);
title('原始图像');
figure(2);
%subplot(3,1,2);
imshow(uint8(bg_bw));
title('背景图像');
figure(3);
%subplot(3,1,3);
imshow(uint8(fg));
%hold on;

```



```

referenceframe=g(:, :, 1);
refprojrow=zeros(row,1); %行投影 即水平方向每列的投影
refprojcol=zeros(1,col);

%参考帧行投影
refprojrowsum=0;
for i=1:row
    for j=1:col
        refprojrow(i)=refprojrow(i)+double(referenceframe(i,j));
    end
    refprojrowsum=refprojrowsum+refprojrow(i);
end
refprojrowmean=refprojrowsum/row;
refprojrow=refprojrow-refprojrowmean; %归一化? ? ? ?
% refprojrow=refprojrow';
% figure;plot(refprojrow);

%余弦滤波
for i=1:row
    if (i<frow)|| (i>row-frow)
        refprojrow(i)=refprojrow(i)*(1+cos(pi*(frow-1-i)/frow))/2;
    end
end
hold on
% plot(refprojrow,'r');
% xlabel('蓝色曲线为投影曲线,红色曲线为余弦滤波后的结果');
% title('参考帧的行投影曲线');

%参考帧列投影
refprojcolsum=0;
for j=1:col
    for i=1:row
        refprojcol(j)=refprojcol(j)+double(referenceframe(i,j));
    end
    refprojcolsum=refprojcolsum+refprojcol(j);
end
refprojcolmean=refprojcolsum/col;
refprojcol=refprojcol-refprojcolmean;
%refprojcol=refprojcol';
%figure;plot(refprojcol);

%余弦滤波
for j=1:col
    if (j<fcol)|| (j>col-fcol)

```



```

curprojrow=cat(1,curprojrow,kuozhanliang_row);

for j=1:61
    for i=1:row
        cr(j)=(refprojrow(j+i-1)-curprojrow(30+i))^2;
    end
end
% figure;plot(cr,'g'); xlabel('垂直方向')
[b,jmin]=min(cr);
dy=31-jmin;

% % % % % % % % % % % % % % 当前帧列投影
curprojcol=zeros(1,col);
curprojcolsum=0;
for j=1:col
    for i=1:row
        %refprojcol(j)=refprojcol(j)+double(referenceframe(i,j));
        curprojcol(j)=curprojcol(j)+double(currentframe(i,j));
    end
    curprojcolsum=curprojcolsum+curprojcol(j);
end
curprojcolmean=curprojcolsum/col;
curprojcol=curprojcol-curprojcolmean;
%curprojcol=curprojcol';
%figure;plot(curprojcol);

% % % % %余弦滤波
for j=1:col
    if (j<fcol)||(j>col-fcol)
        curprojcol(j)=curprojcol(j)*(1+cos(pi*(fcol-1-j)/fcol))/2;
    end
end
%figure;plot(refprojcol);hold on;
%plot(curprojcol,'r'); xlabel('蓝色曲线为投影曲线,红色曲线为余弦滤波后的结果')
%title('当前帧的列投影曲线');
%280 单位的相关运算
%水平方向最大偏移正负 20 个像素
cc=zeros(1,41);
kuozhanliang_col=zeros(1,41);
refprojcol=cat(2,refprojcol,kuozhanliang_col);
curprojcol=cat(2,curprojcol,kuozhanliang_col);

for i=1:41
    for j=1:col
        cc(i)=(refprojcol(j+i-1)-curprojcol(20+j))^2;
    end
end

```



```

    refprojrowsum=refprojrowsum+refprojrow(i);
end
refprojrowmean=refprojrowsum/row;
refprojrow=refprojrow-refprojrowmean;    %归一化? ? ? ?
% refprojrow=refprojrow';
% figure;plot(refprojrow);

%余弦滤波
for i=1:row
    if (i<frow)|| (i>row-frow)
        refprojrow(i)=refprojrow(i)*(1+cos(pi*(frow-1-i)/frow))/2;
    end
end
hold on
% plot(refprojrow,'r');
% xlabel('蓝色曲线为投影曲线,红色曲线为余弦滤波后的结果');
% title('参考帧的行投影曲线');

%参考帧列投影
refprojcolsum=0;
for j=1:col
    for i=1:row
        refprojcol(j)=refprojcol(j)+double(referenceframe(i,j));
    end
    refprojcolsum=refprojcolsum+refprojcol(j);
end
refprojcolmean=refprojcolsum/col;
refprojcol=refprojcol-refprojcolmean;
%refprojcol=refprojcol';
%figure;plot(refprojcol);

%余弦滤波
for j=1:col
    if (j<fcol)|| (j>col-fcol)
        refprojcol(j)=refprojcol(j)*(1+cos(pi*(fcol-1-j)/fcol))/2;
    end
end
%hold on ; plot(refprojcol,'r');
%xlabel('蓝色曲线为投影曲线,红色曲线为余弦滤波后的结果');
%title('参考帧的列投影曲线');

%   referenceframe=buchangframe;
%   refprojrow=curprojrow;
%   refprojcol=curprojcol;

```

```

end

H2 = figure; set(H2,'name','generating final movie ...')
for i = 1:length([gr(1,1,:)])
    imshow(gr(:,i),[0 255]);
    %pause(0.1);
    hold on;%我加入的
    plot(102,128, 'ro' )

movStab(i) =

getframe(H2);
end
close(H2)

H3 = figure; set(H3,'name','Final Stabilized Movie')
imshow(gr(:,1),[0 255]);
curPos = get(H3,'position'); %何意?
set(H3,'position',... % [left bottom width height]
    [60 scrz(4)-100-(movInfo.Height+50) curPos(3:4)]);
movie(H3,movStab,1,movInfo.FramesPerSecond)

% save out final movie & workspace
movie2avi(movStab,[fileName(1:end-4) '_out.avi'], ...
    'fps',movInfo.FramesPerSecond,'compression','None');
save(sprintf('Wkspace_at_d%d-%02d-%02d_t%02d-%02d-%02d',fix(clock)))

%profile report runtime
return

```

4.问题四

```

detectObject.m
%%Time: 2017.9.19
%%Name:Michael Beechan(陈兵)
%%School:Chongqing university of technology
function detectObject()
the_detect_num = 0;
video = mmReader('D:\数学建模代码\问题 4\office\office.avi');
numFrames = video.NumberOfFrames;
ref = imread('D:\数学建模代码\问题 4\office\运行\1020.jpg');
Double_ref = im2bw(ref);
S1 = sum(sum(Double_ref > 0));
Arr = [];
k = 1;
for num = 1 : numFrames-8
    temp1 = im2bw(imread(strcat('D:\数学建模代码\问题 4\office\运行\', num2str(num),'.jpg'),'.jpg'));

```

```

S2 = sum(sum(temp1 > 0));
if S1 <= S2
    path='D:\数学建模代码\问题 4\office\显著目标\';
    imwrite(temp1, strcat(path, int2str(num), '.jpg'));
    the_detect_num = the_detect_num + 1;
    Arr(k) = num;
    k = k + 1;
end
end
Percent = the_detect_num / 1438;
sprintf('%2.2f%%', Percent*100)
sprintf('%d', the_detect_num)
Arr

```

5.问题 5

%特征点的匹配，主要应用 harris 角点的检测,match 单向匹配 函数

%适合于有白边的图像，因为，在加窗滤波的时候，没有限定范围，尽量保证角点不在边上

```

clc;
clear;
close all;

```

```

%% % source1 = aviread('..\问题 5\4p-c0.avi');
%% % source2 = aviread('..\问题 5\4p-c1.avi');
source1 = VideoReader('4p-c0.avi');
source2 = VideoReader('4p-c1.avi');

```

```

source3 = VideoReader('4p-c2.avi');
source4 = VideoReader('4p-c3.avi');

```

```

nFrames1 = source1.NumberOfFrames;
nFrames2 = source2.NumberOfFrames;

```

```

nFrames3 = source3.NumberOfFrames;
nFrames4 = source4.NumberOfFrames;

```

```

bg1=double(rgb2gray(read(source1,1)));
bg2=double(rgb2gray(read(source2,1)));

```

```

bg3=double(rgb2gray(read(source3,1)));
bg4=double(rgb2gray(read(source4,1)));

```

```

[m,n]=size(bg1);

```



```

theld=40;

clors=['b' 'r' 'g' 'k' 'y' 'c' 'm'];

bFirst=1;
bFirst2=1;

bFirst3=1;
bFirst4=1;

for i = 500:nFrames1
    i
    if i==1000
        end
    % % %    imwrite(read(source1,i),['..\Video1\' num2str(i,'%05d') '.jpg']);
    % % %    imwrite(read(source2,i),['..\Video2\' num2str(i,'%05d') '.jpg']);
    % % %    continue;

    a1 = double(rgb2gray(read(source1,i)));    % read in frame
    a2 = double(rgb2gray(read(source2,i)));

    a3 = double(rgb2gray(read(source3,i)));    % read in frame
    a4 = double(rgb2gray(read(source4,i)));

    det1=abs(a1-bg1);
    det2=abs(a2-bg2);

    det3=abs(a3-bg3);
    det4=abs(a4-bg4);

    det1(det1<theld)=0;
    det1(det1>=theld)=1;
    se1=strel('diamond', 10);
    det1=imdilate(det1,se1);    %图像膨胀
    det1=bwareaopen(det1,1000);%去除图像中面积小于 500 的区域

    det2(det2<theld)=0;
    det2(det2>=theld)=1;

    se2=strel('diamond', 10);
    det2=imdilate(det2,se2);    %图像膨胀
    det2=bwareaopen(det2,1000);

```

```

det3(det3<theld)=0;
det3(det3>=theld)=1;
se3=strel('diamond', 10);
det3=imdilate(det3,se3);    %图像膨胀
det3=bwareaopen(det3,1000);%去除图像中面积小于 500 的区域

det4(det4<theld)=0;
det4(det4>=theld)=1;
se4=strel('diamond', 10);
det4=imdilate(det4,se4);    %图像膨胀
det4=bwareaopen(det4,1000);%去除图像中面积小于 500 的区域
%% % %      det2= imfill(det2,'holes');%将原图填充孔洞

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
割%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

目 标 分

```

box1=regionprops(det1,'BoundingBox');
box2=regionprops(det2,'BoundingBox');

box3=regionprops(det3,'BoundingBox');
box4=regionprops(det4,'BoundingBox');

for i=1:length(box1)
    box1(i).BoundingBox=round(box1(i).BoundingBox);
    if box1(i).BoundingBox(1)+box1(i).BoundingBox(3)>n
        box1(i).BoundingBox(3)=n-box1(i).BoundingBox(1);
    end
    if box1(i).BoundingBox(2)+box1(i).BoundingBox(4)>m
        box1(i).BoundingBox(4)=m-box1(i).BoundingBox(2);
    end
end

for i=1:length(box2)
    box2(i).BoundingBox=round(box2(i).BoundingBox);

    if box2(i).BoundingBox(1)+box2(i).BoundingBox(3)>n
        box2(i).BoundingBox(3)=n-box2(i).BoundingBox(1);
    end
    if box2(i).BoundingBox(2)+box2(i).BoundingBox(4)>m
        box2(i).BoundingBox(4)=m-box2(i).BoundingBox(2);
    end
end

for i=1:length(box3)

```

```

box3(i).BoundingBox=round(box3(i).BoundingBox);
if box3(i).BoundingBox(1)+box3(i).BoundingBox(3)>n
    box3(i).BoundingBox(3)=n-box3(i).BoundingBox(1);
end
if box3(i).BoundingBox(2)+box3(i).BoundingBox(4)>m
    box3(i).BoundingBox(4)=m-box3(i).BoundingBox(2);
end
end

```

```

for i=1:length(box4)
    box4(i).BoundingBox=round(box4(i).BoundingBox);
    if box4(i).BoundingBox(1)+box4(i).BoundingBox(3)>n
        box4(i).BoundingBox(3)=n-box4(i).BoundingBox(1);
    end
    if box4(i).BoundingBox(2)+box4(i).BoundingBox(4)>m
        box4(i).BoundingBox(4)=m-box4(i).BoundingBox(2);
    end
end

```

```

if length(box1)>0
    if bFirst==1
        boxOld1=box1;
        bFirst=0;
    else
        [boxOld1,num] = RectAna(boxOld1, box1,det1,m,n);
    end
end

```

```

else
    boxOld1=box1;
end

```

```

if length(box2)>0
    if bFirst2==1
        boxOld2=box2;
        bFirst2=0;
    else
        [boxOld2,num] = RectAna(boxOld2, box2,det2,m,n);
    end
end

```

```

else
    boxOld2=box2;
end

```

```

if length(box3)>0
    if bFirst3==1
        boxOld3=box3;
        bFirst3=0;
    else
        [boxOld3,num] = RectAna(boxOld3, box3,det3,m,n);
    end

else
    boxOld3=box3;
end

if length(box4)>0
    if bFirst4==1
        boxOld4=box4;
        bFirst4=0;
    else
        [boxOld4,num] = RectAna(boxOld4, box4,det4,m,n);
    end

else
    boxOld4=box4;
end

figure(1)
subplot(221),imshow(det1);
subplot(222),imshow(det2);

subplot(223),imshow(det3);
subplot(224),imshow(det4);

figure(2)
subplot(221),imshow(uint8(a1));
hold on
for j=1:length(boxOld1)
    rectangle('position',boxOld1(j).BoundingBox,'edgecolor',clors(j));

text(boxOld1(j).BoundingBox(1)+boxOld1(j).BoundingBox(3)/2-10,boxOld1(j).BoundingBox(2)+boxOld1(j).BoundingBox(4)/2,num2str(j),'FontSize',16,'FontWeight','Bold','Color',clors(j));
end
hold off
subplot(222),imshow(uint8(a2));
hold on
for j=1:length(boxOld2)

```

```

        rectangle('position',boxOld2(j).BoundingBox,'edgecolor',clors(j));

text(boxOld2(j).BoundingBox(1)+boxOld2(j).BoundingBox(3)/2-10,boxOld2(j).BoundingBox(2)+boxOld2(j).Bou
ndingBox(4)/2,num2str(j),'FontSize',16,'FontWeight','Bold','Color',clors(j));
    end
    hold off

    subplot(223),imshow(uint8(a3));
    hold on
    for j=1:length(boxOld3)
        rectangle('position',boxOld3(j).BoundingBox,'edgecolor',clors(j));

text(boxOld3(j).BoundingBox(1)+boxOld3(j).BoundingBox(3)/2-10,boxOld3(j).BoundingBox(2)+boxOld3(j).Bou
ndingBox(4)/2,num2str(j),'FontSize',16,'FontWeight','Bold','Color',clors(j));
    end
    hold off
    subplot(224),imshow(uint8(a4));
    hold on
    for j=1:length(boxOld4)
        rectangle('position',boxOld4(j).BoundingBox,'edgecolor',clors(j));

text(boxOld4(j).BoundingBox(1)+boxOld4(j).BoundingBox(3)/2-10,boxOld4(j).BoundingBox(2)+boxOld4(j).Bou
ndingBox(4)/2,num2str(j),'FontSize',16,'FontWeight','Bold','Color',clors(j));
    end
    hold off

end

return;

```

6.问题 6

```
function [rects,num] = RectAna(rectOld, rectNew,bw,m,n)
```

```

    num1=length(rectOld);
    num2=length(rectNew);

%%%%%%排除不符合规律的目标%%%%%%%%
    k=1;
    for i=1:num2
        if rectNew(i).BoundingBox(3)<rectNew(i).BoundingBox(4)*0.85

```

```

        rectTemp(k)=rectNew(i);
        k=k+1;
    elseif rectNew(i).BoundingBox(3)>n/2
        rectTemp(k)=rectNew(i);
        k=k+1;
    end
end
end
if k~=1
    rectNew=rectTemp;
    num2=length(rectNew);
end
%%%%%%%%%%排除不符合规律的目标%%%%%%%%%%

```

```

while num2<num1
    [rectNew,num2]=rectDetach(rectOld,rectNew,bw);
end

```

```

%原目标特征信息
feaOld=zeros(num1,4);
for i=1:num1
    feaOld(i,1)=rectOld(i).BoundingBox(3);
    feaOld(i,2)=rectOld(i).BoundingBox(4)/3;
    feaOld(i,3)=rectOld(i).BoundingBox(1)+rectOld(i).BoundingBox(3)/2;
    feaOld(i,4)=rectOld(i).BoundingBox(2)+rectOld(i).BoundingBox(4)/2;
end

```

```

%新目标特征信息
feaNew=zeros(num2,4);
for i=1:num2
    feaNew(i,1)=rectNew(i).BoundingBox(3);
    feaNew(i,2)=rectNew(i).BoundingBox(4)/3;
    feaNew(i,3)=rectNew(i).BoundingBox(1)+rectNew(i).BoundingBox(3)/2;
    feaNew(i,4)=rectNew(i).BoundingBox(2)+rectNew(i).BoundingBox(4)/2;
end

```

```

if num2==num1
    rects=rectOld;
    for i=1:num1
        [~,ind]=min(sum(( repmat(feaOld(i,:),num1,1)-feaNew).^2,2));
        rects(i)=rectNew(ind);
        feaNew(ind,:)=feaNew(ind,:)*100;
    end
end

```

```

end

num=num1;

end

if num2>num1
    bAdd=0;
    mask=zeros(1,num2);
    rects=rectOld;
    for i=1:num1
        [~,ind]=min(sum(( repmat(feaOld(i,:),num2,1)-feaNew).^2,2));
        rects(i)=rectNew(ind);
        feaNew(ind,:)=feaNew(ind,:)*100;
        mask(ind)=1;
    end

    for i=1:num2
        if mask(i)==0
            if rectNew(i).BoundingBox(1)<=5 ||
rectNew(i).BoundingBox(1)+rectNew(i).BoundingBox(3)>=n-4 || rectNew(i).BoundingBox(2)<=5 ||
rectNew(i).BoundingBox(2)+rectNew(i).BoundingBox(4)>=m-4
                if bAdd==0
                    rects(num1+1)=rectNew(i);
                else
                    if
rectNew(i).BoundingBox(3)*rectNew(i).BoundingBox(4)>rects(num1+1).BoundingBox(3)*rects(num1+1).Bound
ingBox(4)
                        rects(num1+1)=rectNew(i);
                    end
                end
            end

            bAdd=1;
        end
        if length(rects)>=4
            break;
        end
    end
end

num=length(rects);

```

end