



## “华为杯”第十四届中国研究生 数学建模竞赛

学 校 上海对外经贸大学，清华大学

参赛队号 K0059

队员姓名 1.邢光珣  
2.张飞  
3.刘佳敏

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：

随着中国城市化发展,城市人口数量急剧增加,对城市安防提出了新的挑战。目前,监控系统在平安城市建设中发挥着极其关键的作用,因此各个城市广泛布控,摄像头的数目成井喷式增长。面对海量的监控数据,如何实时在复杂场景下发现和跟踪目标成为广泛研究的问题。本文通过不同的模型依次解决了题目中提出的静态背景下前景目标提取、动态背景下前景目标提取、晃动镜头下前景目标提取、多镜头下前景目标提取以及人群聚散、爆炸、建筑物倒塌等异常事件发现问题,为视频监控中的各类复杂问题提供了有效的解决方案,具有很强的现实意义。

对于问题一,本文采用了**混合高斯模型**对含有静态背景的监控视频进行前景提取。本文从背景模型初始化、背景减除、背景更新三个方面详细描述了模型的建立过程和参数选取过程,并运用该模型完成了多个场景下的前景提取(图 2.1-2.4),选取了多个关键帧和其它算法进行对比。在此基础上,本文提出了一系列评价指标,通过比较不同算法的召回率、准确率和 F 值等指标(表 2.1-2.4),发现混合高斯模型有较好的效果。

对于问题二,本文采用了 **ViBe 模型**对含有动态背景的监控视频进行前景提取,并详细阐述了模型建立过程。本文运用 ViBe 模型对含有水面波动的场景进

行了前景体局，并和其它算法在关键帧（图 3.3）和各项指标上进行了详细对比（表 3.1）。

对于问题三，本文首先运用**像移补偿技术**对视频进行了稳像处理，然后再进行前景提取。本文主要从运动模型与运动估计、背景匹配、图像补偿多个方面介绍了像移补偿模型的建立过程，最后对比了采用稳像技术和不采用稳像技术在前景提取上的差异，并发现采用稳像技术的前景提取技术在各个指标上有显著提升（表 4.1-4.4）。

对于问题四，本文根据各个场景进行了不同分类，然后依据分类采用了前文已验证具有较好效果的模型，即对不含晃动的静态背景采用混合高斯模型、对不含晃动的动态背景采用 ViBe 模型。在获得前景提取照片后，分析对应帧的变化情况和密度，找到包含显著目标前景的关键帧。本文给出了所有场景的关键帧信息（表 5.1），并选取了部分帧进行验证。

对于问题五，本文分析了多摄像头下目标匹配和跟组等关键问题，并采用了**坐标转换**的方法来校对各个视频的前景提取。本文选用了公开数据中 Campus 和 Laboratory 两个场景，分析了各个视频的拍摄角度，然后选取了多个关键帧展示了算法的效果（图 6.1-6.2）。

对于问题六，本文分析了人群或其他异常检测的难点，并采用了**灰度共生矩阵和光流法相结合**的模型来发现异常事件。首先本文利用正常的视频作为输入，在上面进行模型训练，然后输入需要被检测的视频，如果包含异常事件，根据光流分析会给出异常事件出现位置。本文选取了人群突然扩散，爆炸，建筑物倒塌等多个场景进行测试，并给出了过个关键帧的详细分析（图 7.3-7.7）。

最后，本文对于各类问题和本文实现的模型进行总结和评价，并探讨了模型推广的可能性。

**关键词：**前景提取    混合高斯模型    ViBe 模型    多摄像    异常事件监测    灰度共生矩阵    光流法

# 目录

<b>1.前言 .....</b>	<b>6</b>
1.1 研究目的与意义.....	6
1.2 问题重述与分析.....	6
1.3 前景提取的主要衡量指标 .....	7
1.4 基本假设.....	7
<b>2.问题一：静态背景下前景目标提取 .....</b>	<b>8</b>
2.1 问题分析.....	8
2.2 基于混合高斯模型的求解方法 .....	8
2.2.1 背景模型初始化 .....	8
2.2.2 背景减除 .....	9
2.2.3 背景更新 .....	9
2.3 算法仿真与结果分析 .....	10
2.3.1 Airport 场景中的前景目标提取 .....	10
2.3.2 Smoke 场景中的前景目标提取.....	11
2.3.3 Corridor 场景中的前景目标提取.....	12
2.3.4 Highway 场景中的前景目标提取.....	13
2.4 实验验证与对比.....	14
2.4.1 Airport 场景中的实验验证与对比.....	14
2.4.2 Smoke 场景中的实验验证与对比 .....	15
2.4.3 Corridor 场景中的实验验证与对比.....	17
2.4.4 Highway 场景中的实验验证与对比 .....	19
<b>3. 问题二：动态背景下前景目标提取 .....</b>	<b>21</b>
3.1 问题分析 .....	21
3.2 基于 ViBe 模型的求解方法 .....	21
3.2.1 模型实现机制.....	21
3.2.2 背景模型初始化 .....	22
3.2.3 背景更新 .....	22
3.3 算法仿真与结果分析 .....	23
3.4 实验验证与对比.....	24
3.4.1 实验效果对比.....	24
3.4.2 评价指标对比.....	25
<b>4.问题三：晃动或偏移下视频前景目标提取.....</b>	<b>26</b>
4.1 问题分析.....	26
4.2 基于电子稳像技术的求解方法 .....	26
4.2.1 模型实现机制.....	26
4.2.2 运动模型与运动估计 .....	27
4.2.3 背景匹配 .....	28
4.2.4 图像补偿.....	28
4.3 算法仿真与结果分析.....	29

4.3.1 Cars6 场景中的前景目标提取 .....	29
4.3.2 Cars7 场景中的前景目标提取 .....	30
4.3.3 People1 场景中的前景目标提取 .....	31
4.3.4 People2 场景中的前景目标提取 .....	31
4.4 实验验证与比对 .....	32
4.4.1 Cars6 场景中的实验验证与比对 .....	32
4.4.2 Cars7 场景中的实验验证与比对 .....	34
4.4.3 People1 场景中的前景目标提取 .....	36
4.4.4 People2 场景中的前景目标提取 .....	38
<b>5.问题四：给定视频显著前景目标帧号 .....</b>	<b>40</b>
5.1 问题分析 .....	40
5.2 场景分类与模型应用 .....	41
5.3 帧数提取 .....	41
5.3.1 包含动态背景信息的五种场景 .....	41
5.3.2 包含静态背景信息的三种场景 .....	48
<b>6.问题五：多镜头下目标前景的提取 .....</b>	<b>53</b>
6.1 问题分析 .....	53
6.2 基于几何约束的目标匹配 .....	53
6.3 算法仿真与结果分析 .....	54
6.3.1 Campus 场景下多角度视频的前景目标提取 .....	54
6.3.2 Laboratory 场景下多角度视频的前景目标提取 .....	57
<b>7. 问题六：异常事件的发现 .....</b>	<b>58</b>
7.1 问题分析 .....	58
7.2 基于灰度矩阵和光流法的人群异常检测 .....	58
7.2.1 灰度共生矩阵 .....	58
7.2.2 基于光流法的人群运动估计 .....	59
7.2.3 算法设计及流程 .....	60
7.3 实验结果及分析 .....	61
7.3.1 人群异常时的异动事件检测 .....	61
7.3.2 发生物体爆炸时的异动事件检测 .....	62
7.3.3 发生建筑物倒塌时的异动事件检测 .....	63
<b>8. 问题总结与评价 .....</b>	<b>64</b>
8.1 问题总结 .....	64
8.2 模型评价 .....	65
8.3 模型推广 .....	65
<b>9 参考文献 .....</b>	<b>65</b>
<b>10 附录 .....</b>	<b>66</b>

## 1.前言

根据国家统计局的数据,我国城市数量已经从新中国成立时的 132 个增加到 2014 年的 860 个,其中 50 万人以上的城市从 1980 年的 51 个发展到 2014 年的 322 个;100 万人口以上的城市从 1980 年的 10 个发展到 2014 年的 303 个;千万级人口城市数量达到 13 个。随着城市现代化进程发展,人口快速增长,引发城市社会治安、重点区域安全防范、交通管理等一系列城市问题,城市治安管理的日常防范和应对重大恐怖、灾害等特殊突发公共事件的安全防范压力也越来越大。

### 1.1 研究目的与意义

对于某些敏感场景,如银行、超市、机场、停车场、交通要道等,出于管理和安全的需要,我们必须采用视频监控系统对该场景进行全天候的监控。当发生异常事件时,监控人员必须在第一时间内记录该异常情况并及时地做出报警。但在这个快速发展的社会里,人类所面临的突发事件和异常事件越来越复杂,因此监控的难度也越来越突出。

因此视频监控系统的发展面临着智能化的问题,智能化的方式不仅可以减少系统的运行维护成本,而且还可减少人为资源的投入。让计算机给视频监控系统提供智能,代替人类的眼睛对视频数据进行分析理解,不但可以减轻监控人员的劳动强度,而且还可以实现许多人工监控不能实现的功能。

智能化视频监控系统的研究逐渐成为近年来计算机视觉领域研究的一个重要方向,而视频中的前景提取和多目标跟踪和识别是智能监控技术中的一个至关重要的课题。由于视频监控中的背景往往是固定的,而人等运动物体进入形成前前景,因此可以通过对背景建模的方法来对场景中的运动前景进行提取。本文将重点研究在不同场景下如何能准确快速的提取视频中前景部分,以及监测视频中人群异常、爆炸或建筑物倒塌等异常事件。

### 1.2 问题重述与分析

问题一: 对一个不包含动态背景、摄像头稳定拍摄大约 5 秒的监控视频,构造提取前景前景如人、车、动物等的数学模型,并对该模型设计有效的求解方法。

问题二: 对包含动态背景信息的监控视频,设计有效的前景目标提取方案。

问题三: 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)。对这种类型的视频,如何有效地提取前景目标?

问题四: 在提供的 8 组视频中,请利用构造的建模方法,从每组视频中选出包含显著前景目标的视频帧标号。

问题五: 如何通过从不同角度同时拍摄的近似同一地点的多个监控视频中有效检测和提取视频前景目标?

问题六：利用所获取前景目标信息，能否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）、物体爆炸、建筑物倒塌等异常事件？

针对每个子问题的应用场景进行详细的分析，发现各个子问题并不是相互独立的。问题二是对问题一的扩展，问题三依赖于问题一和二的求解，问题四的求解依赖于问题一、二、三的求解。

对于问题一、二和三问题的求解，本文将详细阐述问题需要解决的难点，并详细阐述模型建立过程，然后在多个场景下进行试验验证，最后与其他算法进行对比。

对于问题四的求解，本文将借助前面已有的模型，对给定场景进行分类，运用最为合适的模型求解。

对于问题五的求解，本文将借助已有信息来完成坐标变换，并给出求解方法。

对于问题六的求解，本文将在真实数据集上进行模型训练，并给定不同场景来验证异常检测的准确性。

最后，本文将对各个模型进行总结和评价。

### 1.3 前景提取的主要衡量指标

如何评价本文对于前景提取效果的主要衡量指标如下：

表 1.1 前景提取效果衡量指标

指标缩写	指标全称	含义
TP	True Positive	前景目标被算法正确判为前景点的个数
FP	False Positive	背景点被算法误判为前景点的个数
FN	False Negative	前景目标被算法误判为背景点的个数
TN	True Negative	背景点被算法正确判为背景点的个数
Re	Recall	召回率， $Re = TP / (TP + FN)$
Sp	Specificity	特指度， $Sp = TN / (TN + FP)$
FPR	False Positive Rate	假正率， $FPR = FP / (FP + TN)$
FNR	False Negative Rate	假负率， $FNR = FN / (TP + FN)$
PWC	Percentage of Wrong Classifications	错误分类率， $PWC = (FN + FP) / (TP + FN + FP + TN)$
P	Precision	准确率， $Precision = TP / (TP + FP)$
F	F-Measure	分类准确性， $F = (2 * P * Re) / (P + Re)$

### 1.4 基本假设

一个视频是由多个帧组成的，我们可以用一个三维数据 $X \in \mathbb{R}^{w \times h \times t}$ ，其中 $w, h$ 代表视频帧的长，宽， $t$ 代表视频帧的帧数。视频也可等价理解为逐帧图片的集合，即 $X = \{x_1, x_2, \dots, x_t\}$ ，其中 $x_i \in \mathbb{R}^{w \times h} (i = 1, 2, \dots, t)$ 为一张长宽分别为 $w, h$ 的图片。在处理视频的过程中，我们假设下面结论成立：

- 假设处理的视频是采集好的短视频，而不是无限视频流
- 假设处理的视频都是灰度视频，不含有色彩信息
- 假设处理的视频清晰度较低，在计算机处理范围内
- 假设处理的视频背景结构比较稳定，变化幅度较小
- 假设处理的视频是连续的，不存在跳帧、坏帧等情况
- 假设处理后的视频的前景提取结果输出是黑白帧，不带有原视频的灰度信息。

## 2.问题一：静态背景下前景目标提取

### 2.1 问题分析

问题一要求对不包含动态背景、摄像头稳定拍摄时间大约 5 秒的监控视频进行前景提取。传统提取算法中比较常用的是帧间差分算法，通常前景区域的像素值相对于上一帧的变化比较明显。比较朴素的一类做法是人为设定一个阈值，如果当前帧的某一像素点大于该阈值，即认为该像素点为前景。但是该类算法的缺点是即使在静态背景下，对光照、阴影等容错不够好。因此可以直接使用背景建模的方法，通过对背景建立模型来减除一些其他因素的影响。

### 2.2 基于混合高斯模型的求解方法

由于静态背景下的前景目标提取技术已经比较成熟，本文直接选用处理该类模型效果较好的混合高斯模型（Gaussian Mixture Model）进行求解。该模型的原理是：在图像中每一个像素点上，通过构建  $K$  个高斯分布来描述像素值的变化。该模型首先由 Stauffer 等人提出，并得到了广泛的认可和应用，大量实验结果显示，在处理复杂的动态背景时，混合高斯模型具有较好的鲁棒性，并且具备一定的实时性，运算复杂度也不是很高。

假设在某一时刻  $t$ ，在位置  $x$  处像素的像素值可以用变量  $X_t$  来表示，且可以用  $K$  个高斯分布来表示其概率密度：

$$f(X_t) = \sum_{i=1}^K w_{i,t} \eta(x_{i,t}, \mu_{i,t}, \Sigma_{i,t}, t) \quad (2.1)$$

上式中， $\eta(x_{i,t}, \mu_{i,t}, \Sigma_{i,t}, t)$  表示第  $i$  个高斯概率密度函数。其中  $\mu_{i,t}$  和  $\Sigma_{i,t}$  分别为均值和协方差矩阵， $w_{i,t}$  表示第  $i$  个高斯分布的权重。基于混合高斯模型的背景减除算法的具体步骤如下：

#### 2.2.1 背景模型初始化

背景模型的初始化问题，实质上是指在开始构建混合高斯背景模型时如何设置模型中的四个参数， $K$ 、 $\mu$ 、 $\sigma^2$ 、 $w$ 。 $K$  值的大小会直接影响到算法的效率，在



实际的应用中K的取值在 3~5 之间，而在本文中，混合高斯模型的K值取为 3。对于 $\mu$ 和 $\sigma^2$ ，则是取多帧视频图像序列中每个像素点的 N 个像素值，利用式(2.2)和(2.3)，即可得到高斯分布函数的均值和方差。

$$\mu_0 = \frac{1}{N} \sum_{i=0}^{N-1} I_t \quad (2.2)$$

$$\sigma_0^2 = \frac{1}{N} \sum_{i=0}^{N-1} (I_t - \mu_0)^2 \quad (2.3)$$

但上述方法由于需要计算 N 个像素值的均值和方差，会占用到较大的计算机内存空间。另一种方法则是初始化一个较大的方差值，均值和权重按式(2.4)进行计算。

$$\mu_t = 255 * \frac{i}{K} \quad (1 \leq i \leq K) \quad w_i = \frac{1}{K} \quad (2.4)$$

本文中采取的是第二种方法，即在初始化上的时候直接将首帧图像的像素值作为第一个高斯分布的均值，方差则取一个相对较大的值，其他高斯分布的均值设为零，方差同理的取一个较大的值，在权重方面，第一个高斯分布的权重值要远远大于其他高斯分布的权重。这样一来能够快速实现背景减除，虽然初始多帧可能存在前景提取不够准确的问题，但是随着背景模型的更新，能够很快达到一个较好的背景减除效果。

### 2.2.2 背景减除

初始化背景模型之后，通过判断当前像素点是否与 K 个高斯模型匹配，来进一步决定像素值属于前景还是背景区域。在 t 时刻，若像素点满足式(2.5)的约束条件，则认为其能够和背景模型匹配。

$$|X_t - \mu_{i,t}| \leq M\sigma_t \quad (2.5)$$

其中， $X_t$ 是像素值，M 为匹配系数，本文中 M 取 2.5。将每一个像素的 K 个高斯分布按 $w_{i,t}/\sigma_{i,t}$ 由小到大进行排序。由于视频图像的背景中通常会存在各种噪声干扰，只是其存在时间不会太久，故其权重相对应的较小，因此我们设置一个权重参数 T 来解决该问题，其中 T 取经验值为 0.85。由于最可能描述背景模型的高斯分布位于序列顶部，取前 $\beta$ 个高斯分布为背景模型，故根据式(2.6)来判断像素点属于前景还是背景。

$$\beta = \arg \min_j \sum_{j=i_1}^{i_w} w_{j,t} \geq T \quad (2.6)$$

上式中，若像素点满足该不等式的高斯模型的序号小于 $\beta$ ，则该像素为背景，否则其判定为前景。

### 2.2.3 背景更新

混合高斯模型由于存在背景更新的策略，故可以较好地适应背景扰动的影响。背景模型更新主要是指各类系数的更新，如均值、方差、权重等。若 $X_t$ 与对应的 K 个高斯模型中某个模型匹配，则需要对高斯分布的权值按照式(2.7)更新，同时按照式(2.8)和式(2.9)对匹配的高斯分布的方差和均值进行更新。

$$w_{i,t} = (1 - \alpha)w_{i,t-1} + \alpha \quad (2.7)$$

$$\mu_{i,t} = (1 - \rho)\mu_{i,t-1} + \rho I_t \quad (2.8)$$

$$\sigma_{i,t}^2 = (1 - \rho)\sigma_{i,t-1}^2 + \rho(I_t - \mu_{i,t})^2 \quad (2.9)$$

上式中 $\alpha$ 为背景的学习率，决定了背景模型更新的速率。若 $\alpha$ 越大，则相应的模型更新速度越快，但同时会出现空洞等问题，会无法完整地提取出前景运动目标；反之，则会牺牲模型更新速率来换取较好的背景减除效果。 $\alpha$ 值需根据实际的应用场景选择， $\rho$ 值为参数学习率，一般情况下取 $\rho \approx \alpha/w_{i,t}$ 。

若 $X_t$ 与对应的所有高斯模型均不匹配，则会初始化一个以 $X_t$ 为均值，较大方差 $\sigma^2$ 和较小权重 $w_0$ 的新模型。其余高斯分布保持均值和方差不变，权值按照式(2.10)更新。

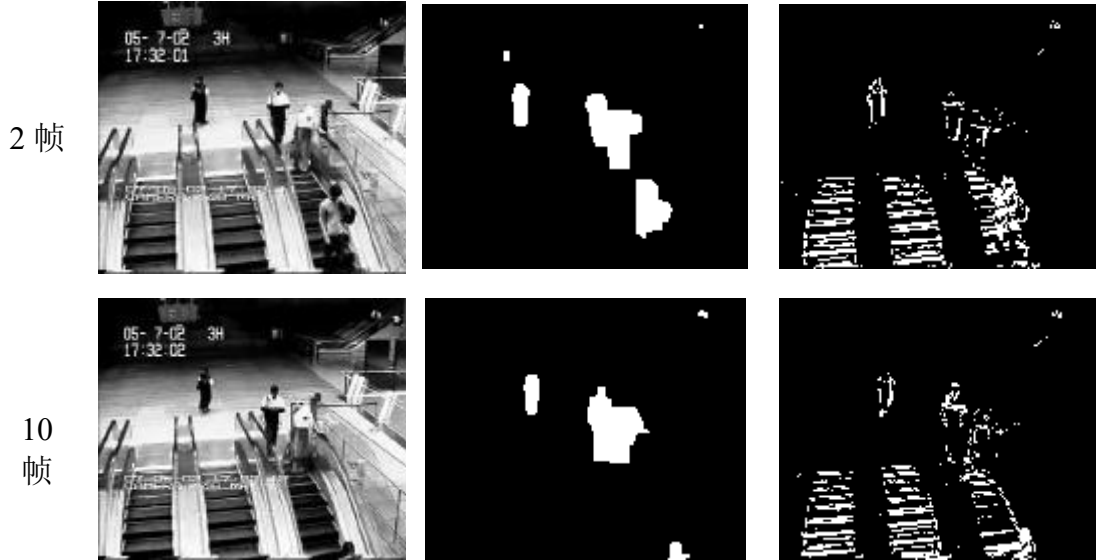
$$w_{i,t} = (1 - \alpha)w_{i,t-1} \quad (2.10)$$

## 2.3 算法仿真与结果分析

本文使用上述模型对附件二中不带晃动-静态背景下场景 airport、smoke 及自行从网站上下载的 corridor、highway 四个场景提取了前景目标，并选取关键帧（如：有明显人物出现等）进行结果分析。

### 2.3.1 Airport 场景中的前景目标提取

Airport 场景视频分解共 48 帧，以下选取第 2、10、26、45 帧为关键帧来展示视频提取的效果。其中，第 2 帧为四人同时在场景内，第 10 帧为其中一人即将离开场景。第 26 帧为其中两人处于场景中并列位置，第 43 帧为其中另一人即将离开场景。选取该 4 帧作为关键帧可以较为明显的看出本模型在关键时点的前景目标提取效果。



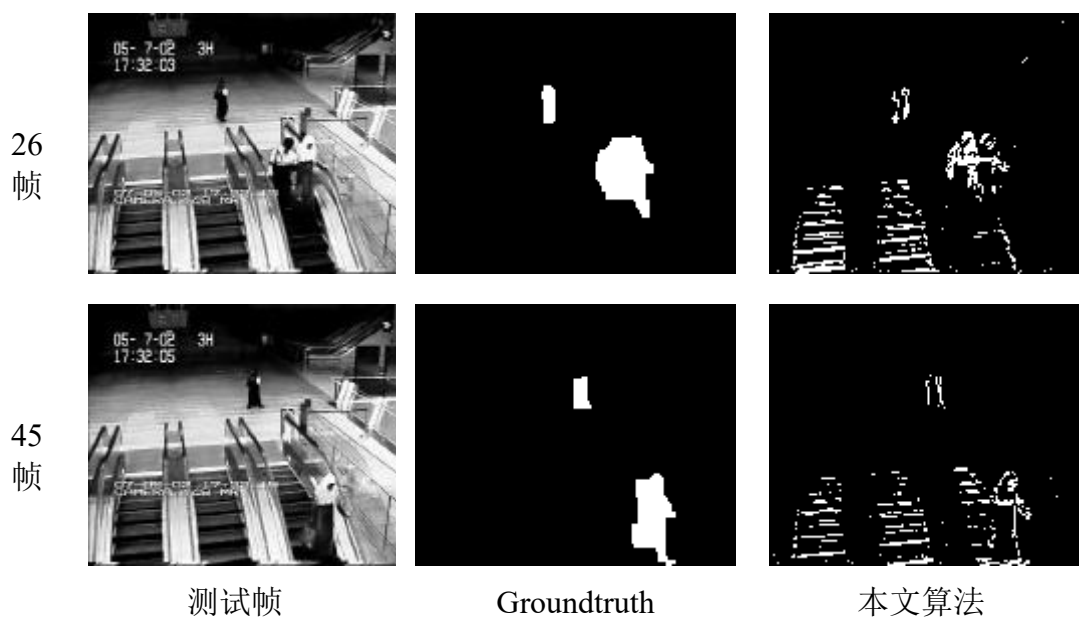
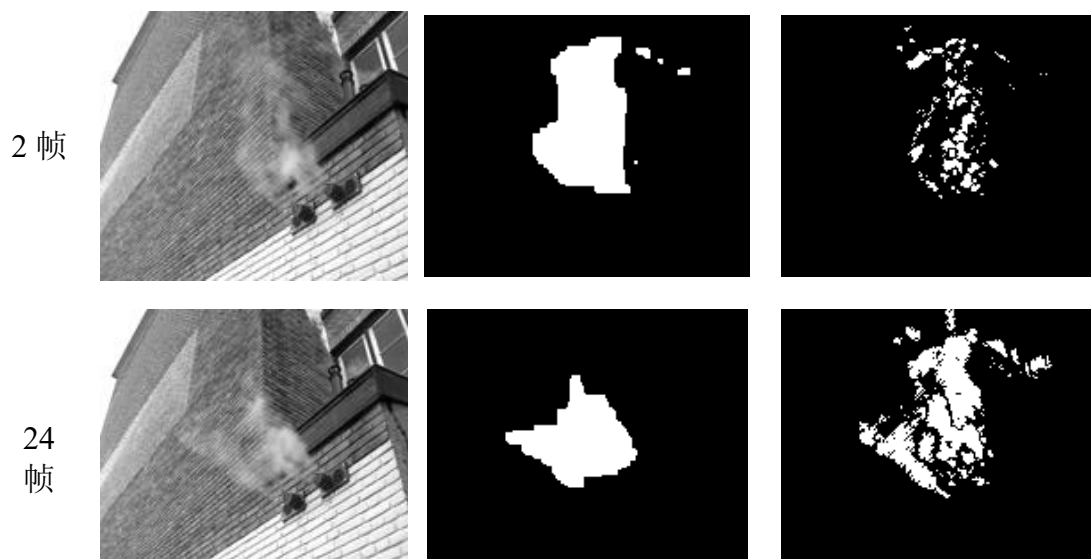


图 2.1 Airport 场景中的前景目标提取

从提取效果上看，本文算法（混合高斯模型）出现的噪声点较少，前景目标提取较为完整，能够看出前景目标轮廓，但对于动态背景的剔除仍存在一些缺陷。

### 2.3.2 Smoke 场景中的前景目标提取

Smoke 场景视频分解共 50 帧，以下选取第 2、24、33、48 帧为关键帧来展示视频提取的效果。选取该 4 帧作为关键帧可以较为明显的看出本模型在关键点的前景目标提取效果。



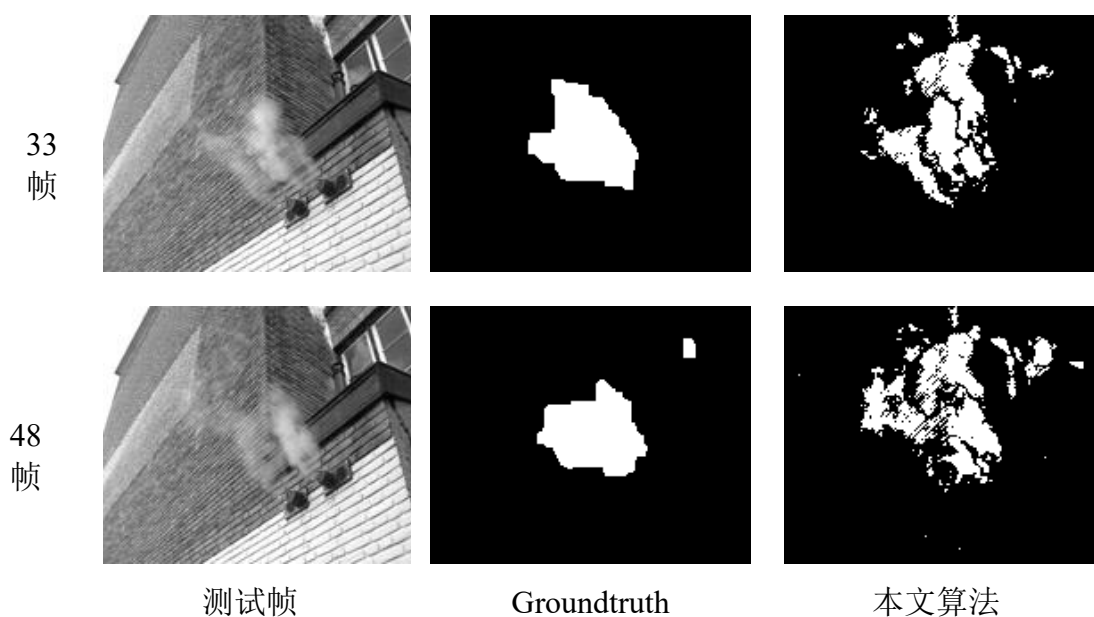
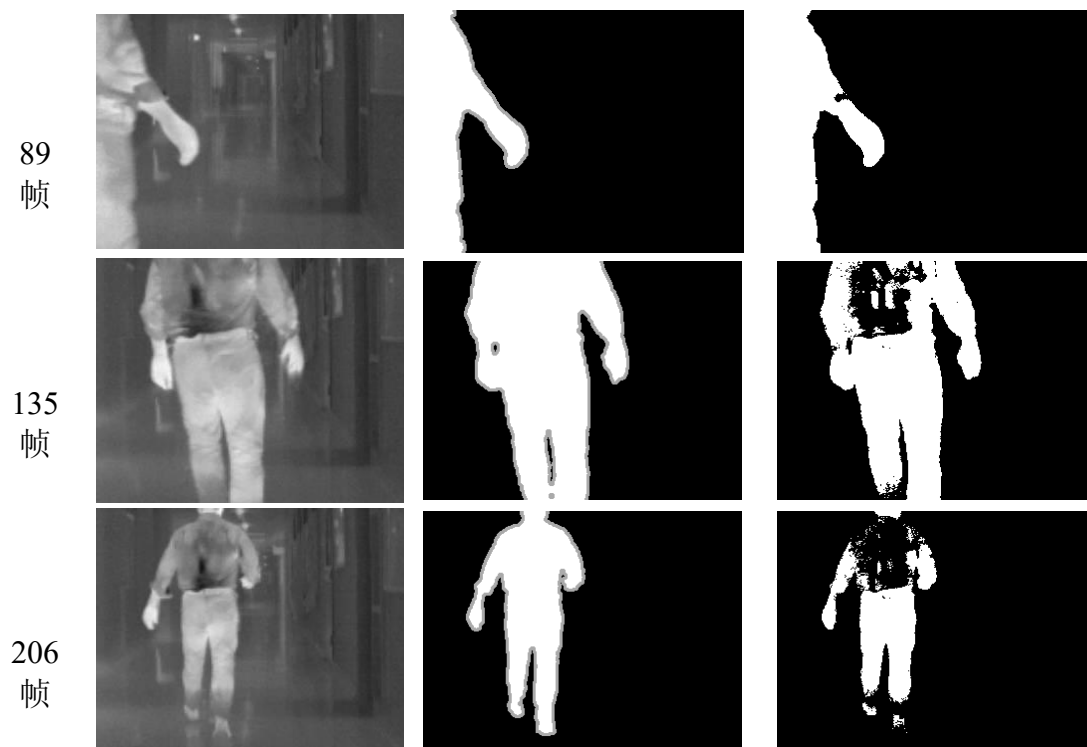


图 2.2 Smoke 场景中的前景目标提取

从提取效果上看，本文算法（混合高斯模型）出现的噪声点较少，前景目标提取较为完整，能够看出前景目标轮廓，但仍存在部分误检的情况。

### 2.3.3 Corridor 场景中的前景目标提取

Corridor 场景视频分解共 300 帧，按人物由近及远分别选取第 89、135、206、270 帧为关键帧来展示视频提取的效果。



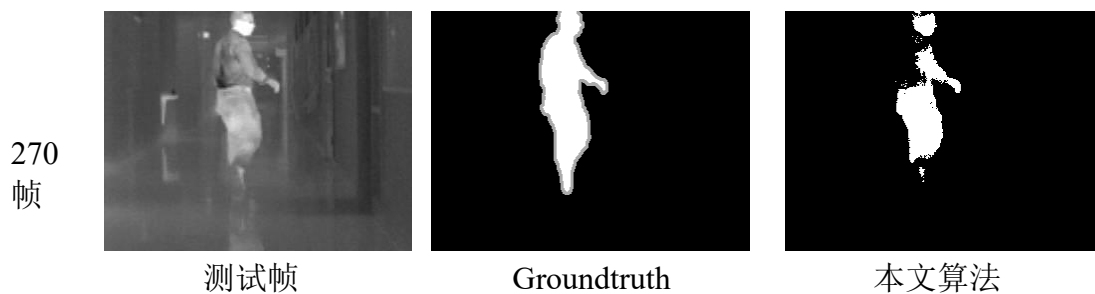


图 2.3 Corridor 场景中的前景目标提取

从提取效果上看，本文算法（混合高斯模型）出现的噪声点较少，前景目标提取较为完整，能够看出前景目标轮廓，但仍存在部分误检的情况。

### 2.3.4 Highway 场景中的前景目标提取

Highway 场景视频分解共 300 帧，按前景目标出现顺序依次选取第 66、156、216、286 帧为关键帧来展示视频提取的效果。具体结果见图 2.4：

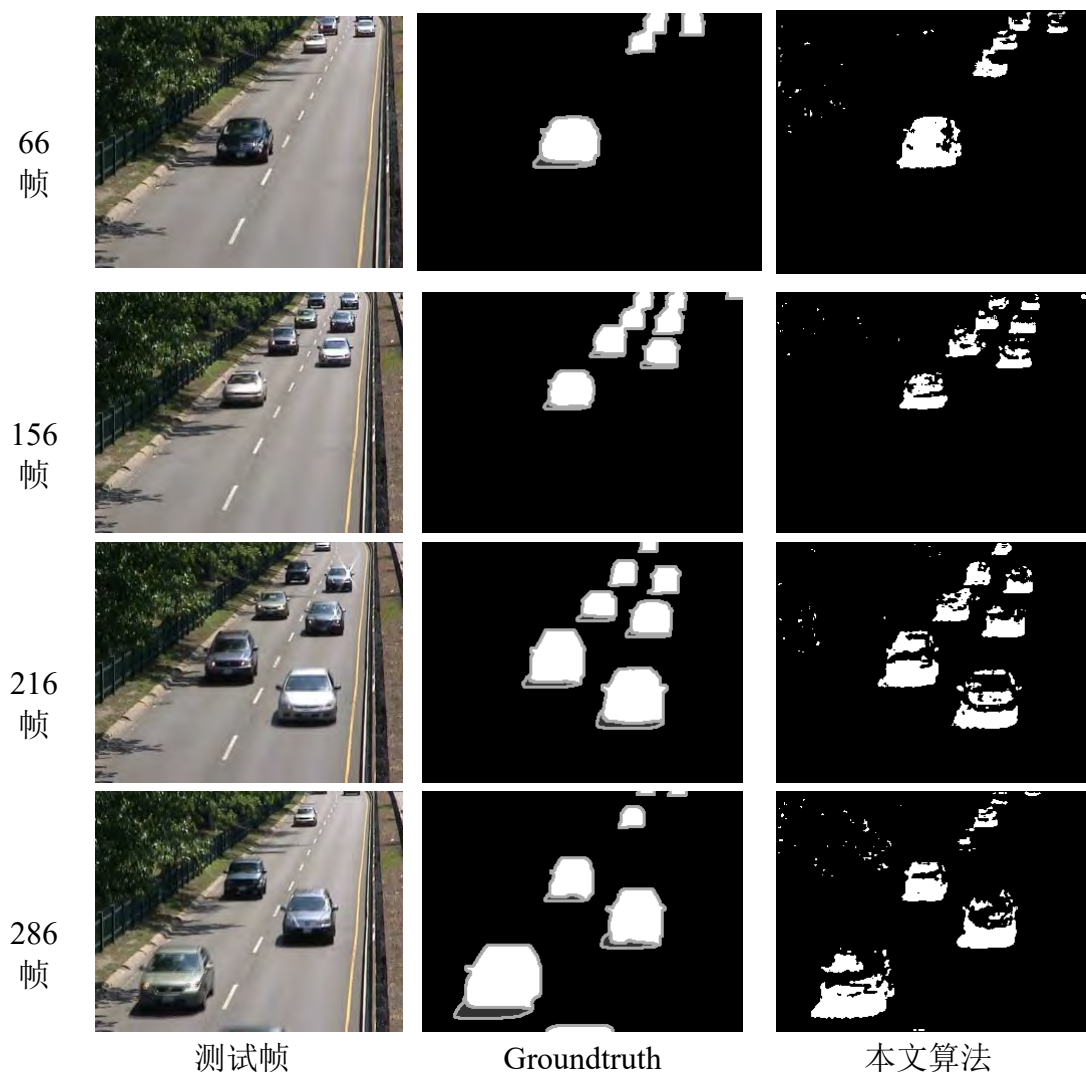


图 2.4 Office 场景中的前景目标提取

从提取效果上看，本文算法（混合高斯模型）出现的噪声点较少，前景目标提取较为完整，能够看出前景目标轮廓，但仍存在部分误检的情况。

## 2.4 实验验证与对比

### 2.4.1 Airport 场景中的实验验证与对比

#### 2.4.1.1 Airport 场景中的不同模型实验效果对比

除混合高斯模型外，我们还采用了单高斯模型和帧间差分法提取前景目标，效果对比如下：

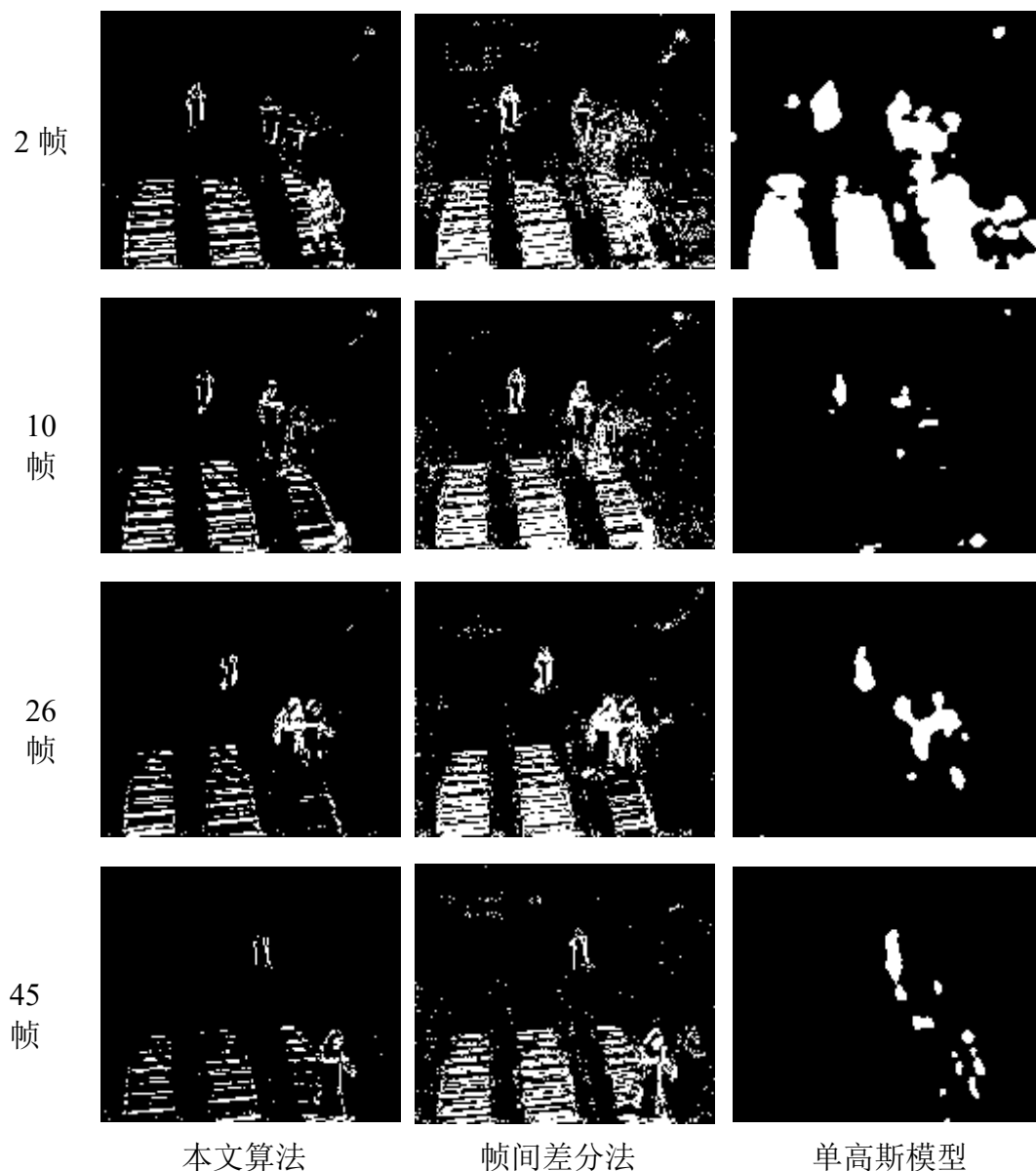


图 2.5 Airport 场景中的不同模型实验效果对比

从对比图上看，帧间差分法存在着噪声点较多的问题，单高斯模型虽然噪声点较少、空洞较少，但在完整性上不及混合高斯模型，反映不出图形的轮廓，从图形对比效果上看，混合高斯模型在解决静态背景的前景目标提取问题时效果更好。

### 2.4.1.2 Airport 场景中的评价指标对比

使用前文提到的评价指标计算出结果如下图所示，可以看出，混合高斯模型在准确度（Precision）、特指度（Specificity）、错误分类率（PWC）、F 值（F-Measure）等方面均显著优于其他模型。再次印证了在解决无晃动-静态背景的前景提取问题方面，混合高斯模型确实为最优模型。

表 2.1 模型评价指标数值表

	混合高斯模型	单高斯模型	帧间差分法
TP	18552	19694	10945
FP	17741	95240	31336
TN	919266	841767	905671
FN	22041	20899	29648
Recall	0.4570	0.4852	0.2696
Precision	0.5112	0.1714	0.2589
Specificity	0.9802	0.9771	0.9881
FPR	0.0189	0.1016	0.0334
FNR	0.5430	0.5148	0.7304
PWC	0.0407	0.1188	0.0624
F-Measure	0.4826	0.2533	0.2641

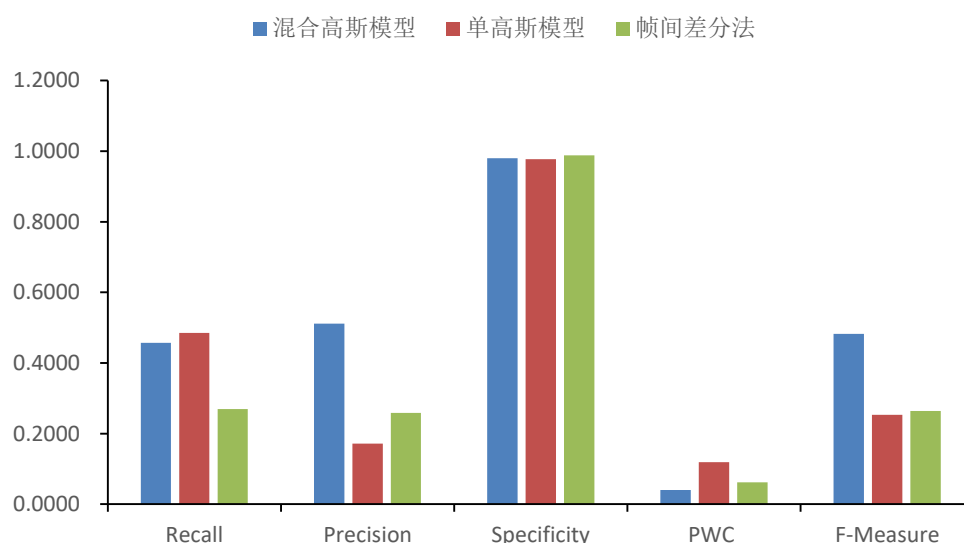


图 2.6 模型主要评价指标直方图

### 2.4.2 Smoke 场景中的实验验证与对比

#### 2.4.2.1 Smoke 场景中的不同模型实验效果对比

除混合高斯模型外，我们还采用了单高斯模型和帧间差分法提取前景目标，效果对比如下：

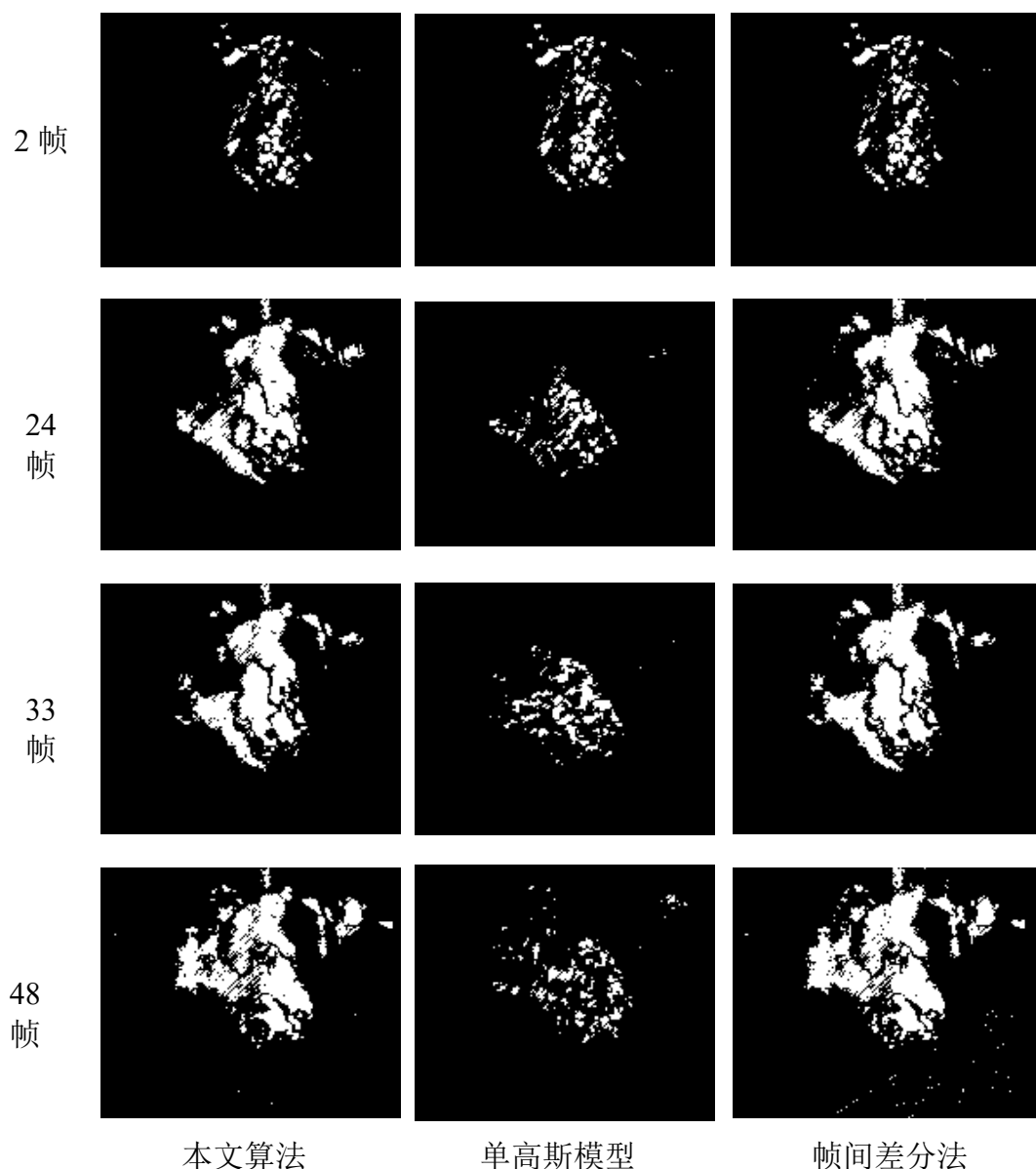


图 2.7 Smoke 场景中的不同模型实验效果对比

从对比图上看，单高斯模型存在着噪声点较多的问题，帧间差分法虽然噪声点较少、空洞较少，但在完整性上不及混合高斯模型，反映不出图形的轮廓，从图形对比效果上看，混合高斯模型在解决静态背景的前景目标提取问题时效果更好。

#### 2.4.2.2 Smoke 场景中的评价指标对比

使用前文提到的评价指标计算出结果如下图所示，可以看出，混合高斯模型在召回率（Recall）、错误分类率（PWC）、F 值（F-Measure）等方面均显著优于其他模型。再次印证了在解决无晃动-静态背景的前景提取问题方面，混合高斯模型确实为最优模型。



表 2.2 模型评价指标数值表

	混合高斯模型	单高斯模型	帧间差分法
TP	78372	33516	74152
FP	67807	4125	56862
TN	1077564	1141246	1088509
FN	46337	91193	50557
Recall	0.6284	0.2688	0.5946
Precision	0.5361	0.8904	0.5660
Specificity	0.9322	0.9715	0.9362
FPR	0.0592	0.0036	0.0496
FNR	0.3716	0.7312	0.4054
PWC	0.0899	0.0750	0.0846
F-Measure	0.5786	0.4129	0.5799

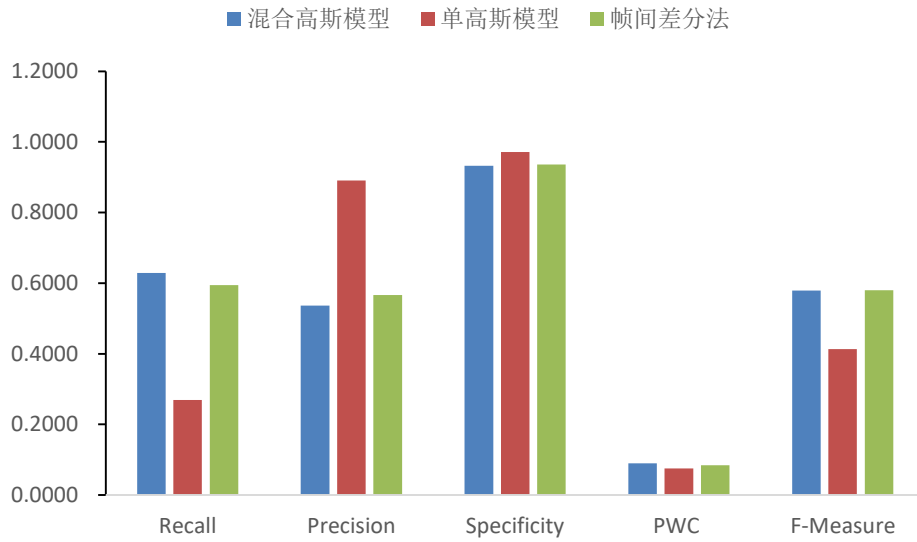


图 2.8 模型主要评价指标直方图

### 2.4.3 Corridor 场景中的实验验证与对比

#### 2.4.3.1 Corridor 场景中的不同模型实验效果对比

除混合高斯模型外，我们还采用了单高斯模型和帧间差分法提取前景目标，效果对比如下：



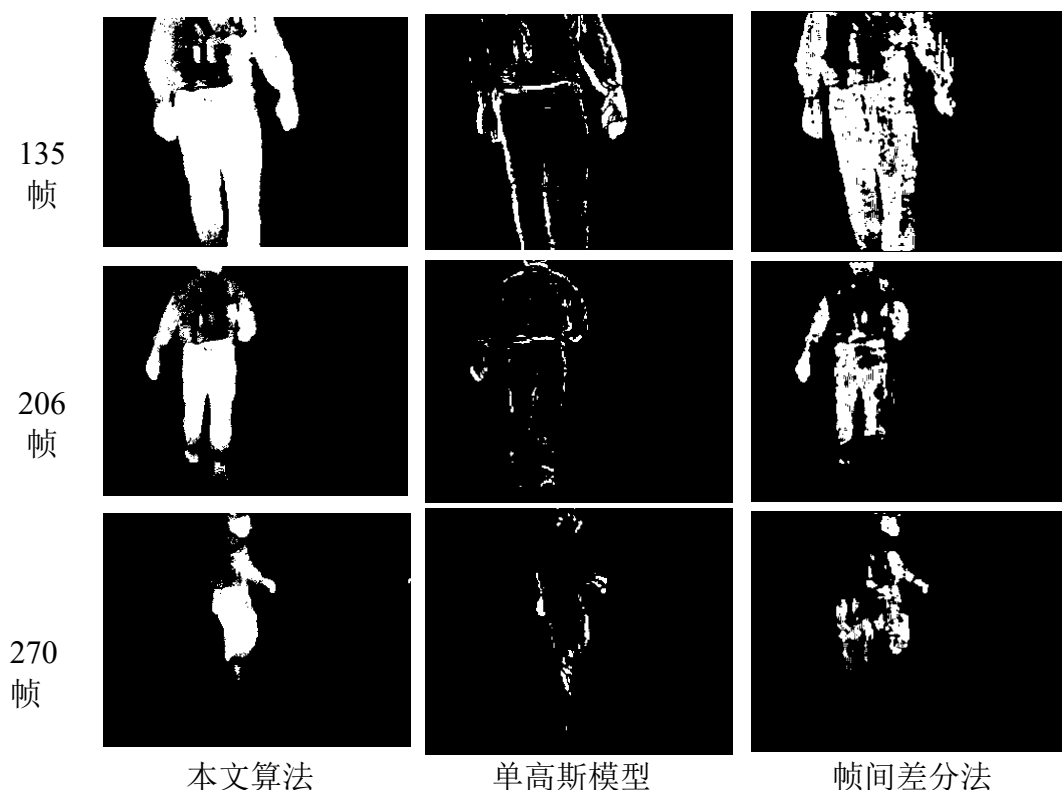


图 2.9 Corridor 场景中的不同模型实验效果对比

从对比图上看，单高斯模型存在着噪声点较多的问题，帧间差分法虽然噪声点较少、空洞较少，但在完整性上不及混合高斯模型，反映不出图形的轮廓，从图形对比效果上看，混合高斯模型在解决静态背景的前景目标提取问题时效果更好。

#### 2.4.3.2 Corridor 场景中的评价指标对比

使用前文提到的评价指标计算出结果如下图所示，可以看出，混合高斯模型在召回率（Recall）、准确度（Precision）、F 值（F-Measure）等方面均显著优于其他模型。再次印证了在解决无晃动-静态背景的前景提取问题方面，混合高斯模型确实为最优模型。

表 2.3 模型评价指标数值表

	混合高斯模型	单高斯模型	帧间差分法
TP	2229416	276445	1646065
FP	2561	18139	42714
TN	19605696	19590118	19565543
FN	652523	2605494	1235874
Recall	0.7736	0.0959	0.5712
Precision	0.9989	0.9384	0.9747
Specificity	0.8979	0.9861	0.9224
FPR	0.0001	0.0009	0.0022
FNR	0.2264	0.9041	0.4288
PWC	0.0291	0.1167	0.0569
F-Measure	0.8719	0.1741	0.7203

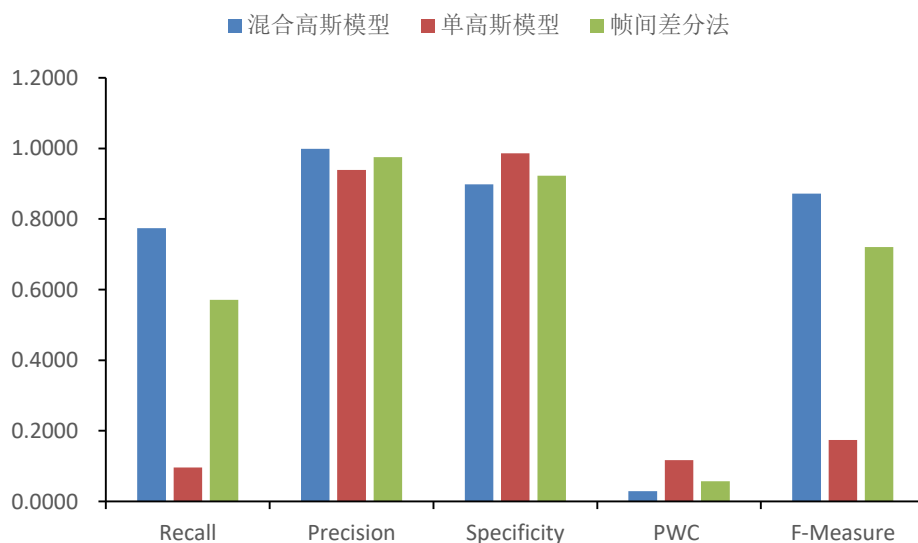
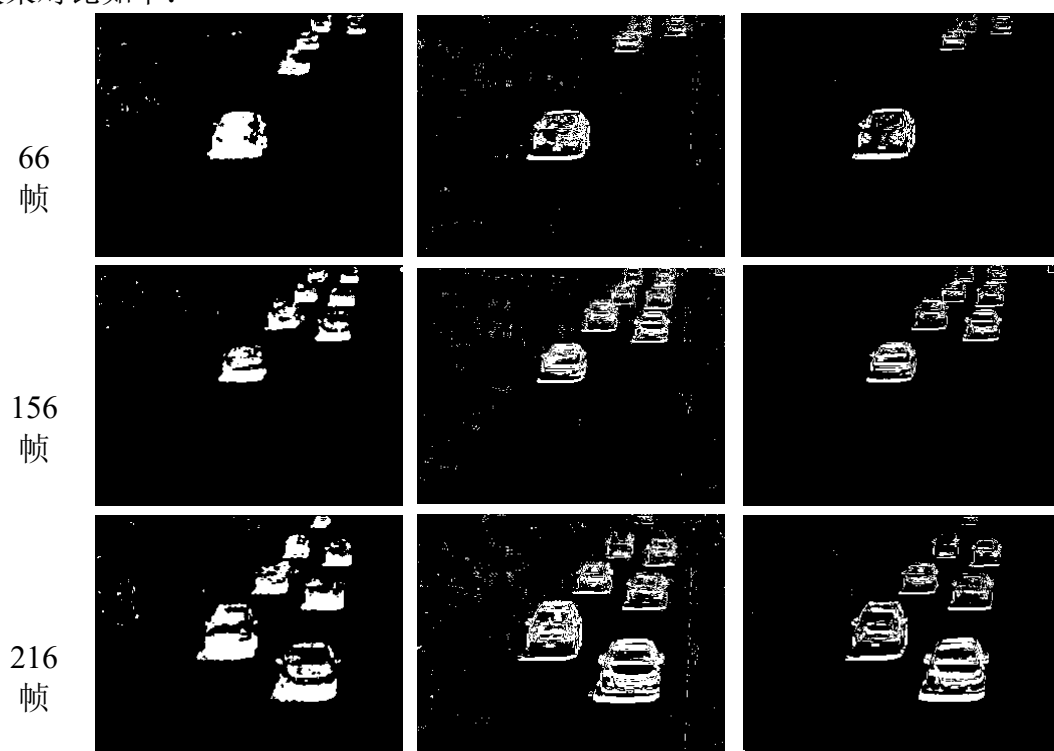


图 2.10 模型主要评价指标直方图

## 2.4.4 Highway 场景中的实验验证与对比

### 2.4.4.1 Highway 场景中的不同模型实验效果对比

除混合高斯模型外，我们还采用了单高斯模型和帧间差分法提取前景目标，效果对比如下：



286  
帧

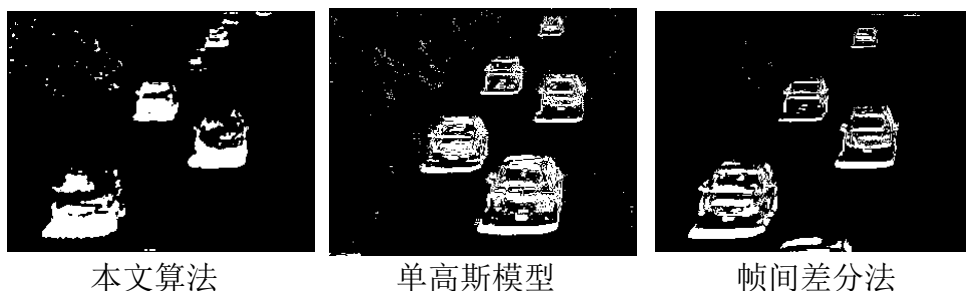


图 2.11 Highway 场景中的不同模型实验效果对比

从对比图上看，单高斯模型存在着噪声点较多的问题，帧间差分法虽然噪声点较少、空洞较少，但在完整性上不及混合高斯模型，反映不出图形的轮廓，从图形对比效果上看，混合高斯模型在解决静态背景的前景目标提取问题时效果更好。

#### 2.4.4.2 Highway 场景中的评价指标对比

使用前文提到的评价指标计算出结果如下图所示，可以看出，混合高斯模型在召回率（Recall）、特指度（Specificity）、F 值（F-Measure）等方面均显著优于其他模型。再次印证了在解决无晃动-静态背景的前景提取问题方面，混合高斯模型确实为最优模型。

表 2.4 模型评价指标数值表

	混合高斯模型	单高斯模型	帧间差分法
TP	143659	105659	67877
FP	32896	25721	7843
TN	7139817	7146992	7164870
FN	84038	122038	159820
Recall	0.6309	0.4640	0.2981
Precision	0.8137	0.8042	0.8964
Specificity	0.9908	0.9854	0.9906
FPR	0.0046	0.0036	0.0011
FNR	0.3691	0.5360	0.7019
PWC	0.1580	0.1997	0.2266
F-Measure	0.7107	0.5885	0.4474

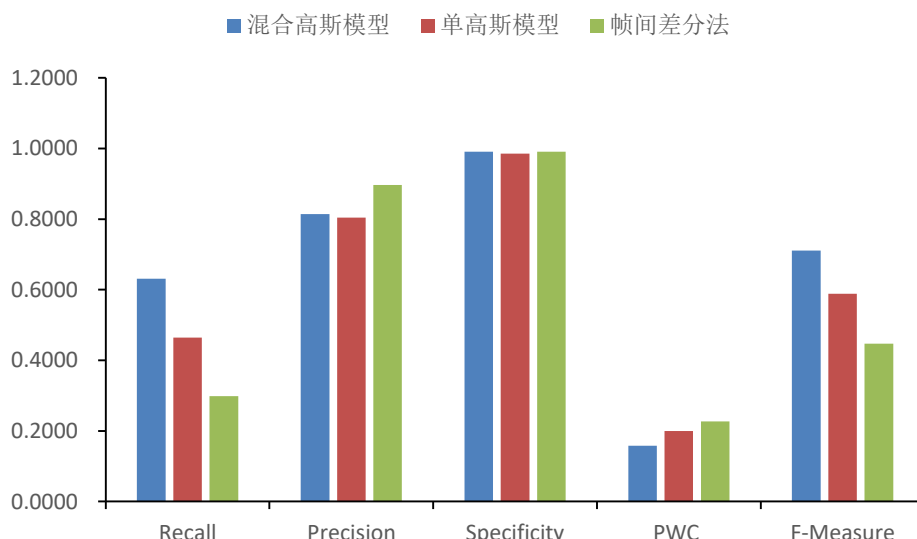


图 2.12 模型主要评价指标直方图

## 3. 问题二：动态背景下前景目标提取

### 3.1 问题分析

对于问题二，要求对包含动态背景信息的监控视频进行前景提取。对于包含例如晃动树叶、水面波动、喷泉变化、窗帘晃动等场景的视频，需要模型能够识别动态背景规律性的变化，并将这些变化进行消除。混合高斯模型虽然可以采用多个高斯函数的方法来对背景进行建模，但是需要设定学习速率来区分前景和背景，但如果设置不当则会引起较多误差。

### 3.2 基于 ViBe 模型的求解方法

ViBe 算法是最近几年提出来的一种非参数化颜色背景建模方法。该算法能够快速提取出目标前景，从而实现背景减除。该算法的基本思路是，图像序列中的每一个像素点均存储了一组历史样本集即背景模型，首先将当前位置像素点的像素值与该背景模型进行对比，来判定其归属；接着按照一定规则随机地替换背景模型中的像素值，从而实现模型的更新；最后，若当前像素值属于背景，则其值会影响到相邻像素的背景模型，从而更新邻近像素点的背景模型。本文将从以下三个方面来阐述该背景减除算法：模型实现机制、背景模型初始化及背景模型更新。

#### 3.2.1 模型实现机制

在智能视频监控中，背景部分绝大多数时间是静止不动的，因此可以通过建立背景模型，来判定当前的像素点属于背景或者前景。在 ViBe 算法中，图像帧中的每一个像素点均存有一个自己的样本集，若当前的像素值被判定为背景点，则其像素值和样本值较为相近。

问题一中使用的混合高斯模型法采取了估计像素点在时域上的概率分布函数，一旦密度函数确定完毕，则不会再发生改变，即当前像素点的像素值会根据确定好的密度函数下的概率来判断其归属。由于离中心值较远的像素点也被用来计算密度函数，故会导致估计得到的密度函数和实际函数之间存在较大的偏差。ViBe 背景建模法中，只有邻域内的像素值会对样本值产生较大的影响，且样本数量限制为  $N$  个，避免了上述问题的发生。

ViBe 算法的实现机制为：在欧式彩色空间中，定义  $v(x)$  为图像帧中第  $x$  个像素点的像素值，集合  $M(x) = \{v_1, v_2, \dots, v_n\}$  表示该像素点在先前  $N$  帧的样本集。定义  $S_R(v(x))$  为以像素点  $x$  为圆心， $R$  为半径的圆形区域。若  $(v(x))$  满足式(3.1)，则认为该像素点属于背景。

$$M(x)[S_R(v(x)) \cap \{v_1, v_2, \dots, v_n\}] \geq \#_{min} \quad (3.1)$$

如图 3.1 所示，表示的是在二维欧拉颜色空间  $(C_1, C_2)$  中将当前像素点与样本集进行比较，之后根据式(3.1)对该像素点进行分类，判断其到底是属于前景点或是背景点。

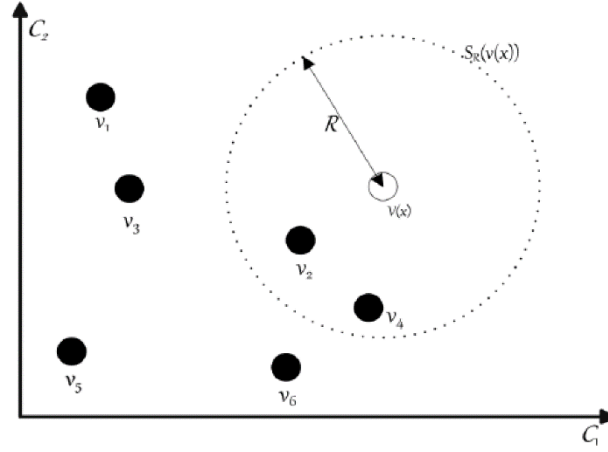


图 3.1 二维欧式空间匹配示意图

### 3.2.2 背景模型初始化

在很多背景减除算法中，均要求几十张图像帧来初始化背景模型，这将花费较长的时间，极大地影响到了背景减除算法的效率。但是 ViBe 算法只需要一帧图像就可以实现背景模型的初始化。利用相邻像素点拥有相似时空分布的特点，我们采用随机选择的策略，选取 8 邻域中像素点的像素值组成样本集来实现背景模型的初始化。实现的具体方式如(3.2)所示。

$$M_0(x) = \{v_0(y) | y \in N_G(x)\} \quad (3.2)$$

该初始化方法计算量较小，具有较好的实时性，能够快速提取出前景目标，但是对噪声等较为敏感，且有一个致命的缺陷便是，容易引入“鬼影”，随着模型不断地更新，在接下来的多帧中，才能逐渐消除“鬼影”。

### 3.2.3 背景更新

由于在智能视频监控中，随着时间推移，背景或多或少会发生变化，而更新背景模型就能够使之适应复杂多变的场景，例如水面波纹、光照突变、前景遮挡等，这样才能够使算法具有较好的鲁棒性。常见的比较保守的更新方法为：背景

模型将永远不会采取被断定为前景点的像素点来进行背景模型的更新，这样的做法存在不可逆的弊端；相反的，如果采用前景像素点和背景像素点来更新背景模型，虽避免了不可逆的缺陷，却会将视频中缓慢移动的物体判定成背景。

ViBe 算法采用的是上述两种更新方法相结合的更新策略，即采取无记忆的更新策略，在每一个像素点的 8 邻域中随机选取一个样本用来替换原背景模型中随机选取的样本，从而可以保证每个样本值在背景模型样本集中存在的时间呈平滑指数衰减。具体的更新过程为：若像素点  $x$  被断定为背景点，则其有  $p$  的概率同时更新自己背景模型中的样本值和其邻域像素点对应背景模型的样本值。通过这样一种像素点的空间特性，可以使背景模型逐渐扩展，可以快速消除“鬼影”。另一方面，若前景点被统计为背景点的次数达到了一定的阈值  $T$ ，则将该像素点判定成背景，并且以  $p$  的概率更新背景模型的样本值。正是由于采取了随机更新方法，保证了样本值在背景模型样本集中存在的时间呈平滑指数衰减，即  $t$  时刻某样本值被选取用来更新的概率为  $1/N$ ，那么在  $dt$  时间内，该样本值仍存在于样本集中的概率如式(3.3)。

$$P(t, t + dt) = \left(\frac{N-1}{N}\right)^{(t+dt)-1} \quad (3.3)$$

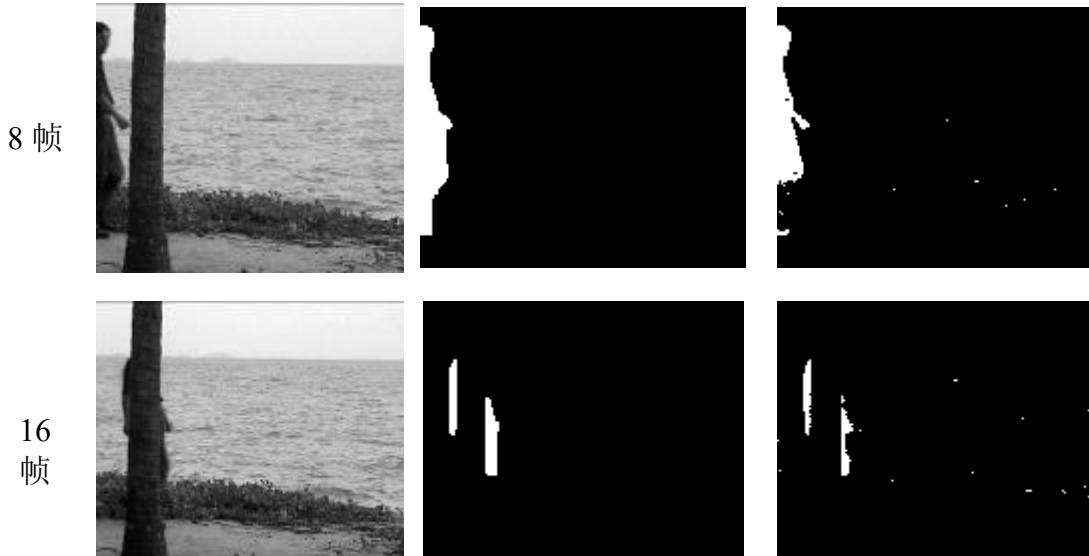
上式可以化简为如式(3.4)。

$$P(t, t + dt) = e^{-\ln\left(\frac{N}{N-1}\right)dt} \quad (3.4)$$

由上式得出的结论是某样本值存在于背景模型中的概率值和时间  $t$  无关，这样保证了其生命周期的平滑性。

### 3.3 算法仿真与结果分析

对于动态背景下前景目标提取我们直接选用附件 2 中不带晃动的动态背景——WaterSurface 场景进行算法仿真，并对结果进行分析。WaterSurface 视频可分解为 48 帧，以下选取第 8、16、21、48 帧为关键帧来展示视频提取的效果。其中，第 8 帧为人物走进场景内，第 16 帧为人物前有遮挡物的场景，第 21 帧为人物刚刚走出遮挡物，第 48 帧为人物无遮挡物站立在动态背景前。选取该 4 帧作为关键帧可以较为明显的看出本模型在关键时点的前景目标提取效果。



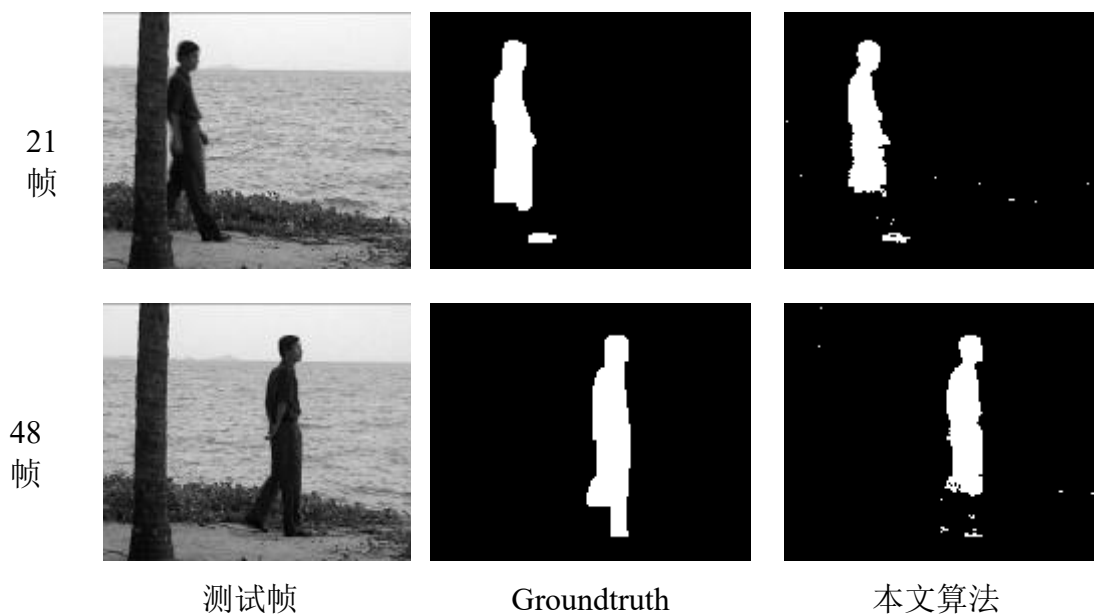


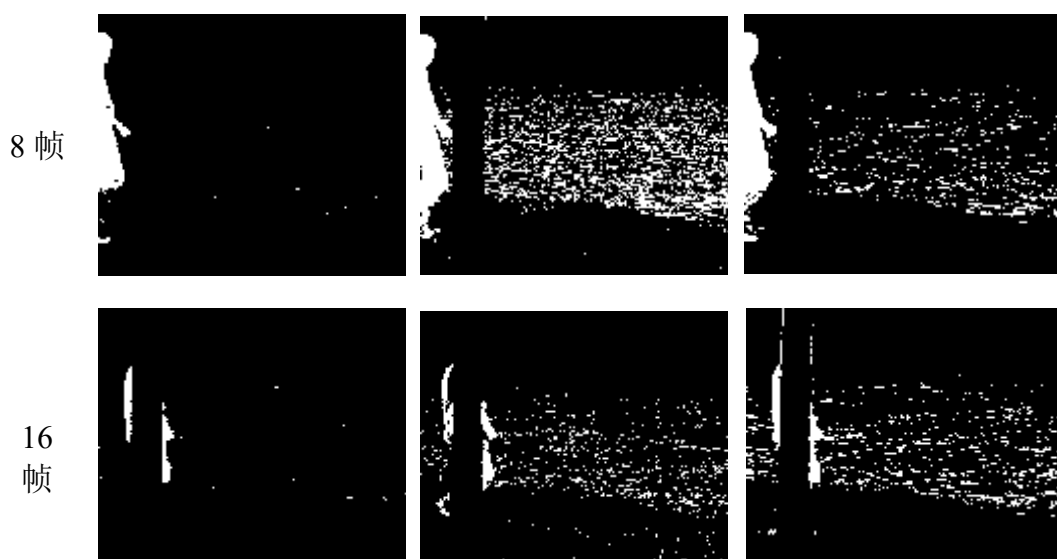
图 3.2 ViBe 模型仿真结果

从提取效果上看，本文算法（ViBe 算法）非常适用于动态无晃动背景的前景目标提取，达到了近似于 Groundtruth 的效果。该算法提取的前景目标背景中出现的噪声点很少，前景目标提取完整、轮廓清晰、空洞很少，背景剔除效果也较好。

### 3.4 实验验证与对比

#### 3.4.1 实验效果对比

除 ViBe 模型外，我们还采用了帧间差分法和混合高斯模型提取前景目标，效果对比如下：





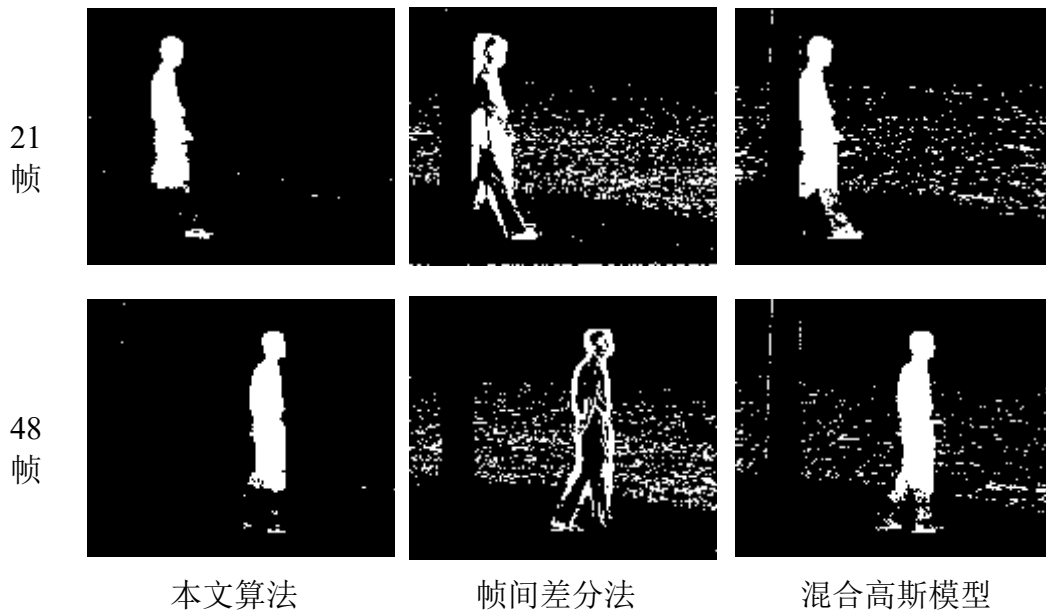


图 3.3 不同模型仿真结果对比

从对比图上看，帧间差分法存在着噪声点较多、空洞较多等问题，混合高斯模型较帧间差分算法虽然噪声点及空洞有所减少，但在完整性及背景剔除效果等方面仍和 ViBe 模型有较大差距。可以看出，在解决动态背景无晃动问题方面，ViBe 模型确实为最优模型。

### 3.4.2 评价指标对比

使用前文提到的评价指标计算出结果如下图所示，可以看出，ViBe 模型在召回率 (Recall)、准确度 (Precision)、特指度 (Specificity)、F 值 (F-Measure) 等方面均显著优于其他模型。再次印证了，在解决动态背景无晃动问题方面，ViBe 模型确实为最优模型。

表 3.1 模型评价指标数值表

	ViBe模型	帧间差分算法	混合高斯模型
TP	37842	21885	43744
FP	1503	62603	34065
TN	911935	850835	879373
FN	11280	27237	5378
Recall	0.7704	0.4455	0.8905
Precision	0.9618	0.2590	0.5622
Specificity	0.9602	0.9749	0.9526
FPR	0.0016	0.0685	0.0373
FNR	0.2296	0.5545	0.1095
PWC	0.0133	0.0933	0.0410
F-Measure	0.8555	0.3276	0.6893

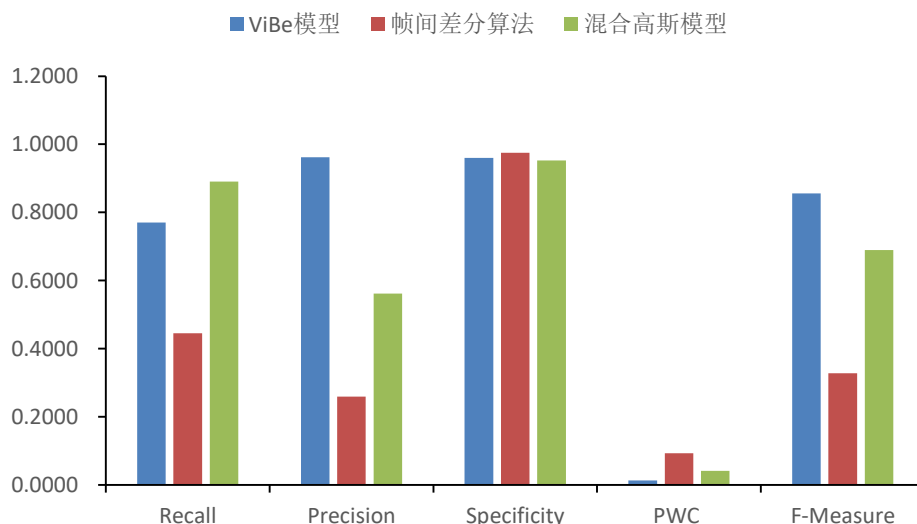


图 3.4 模型主要评价指标直方图

## 4.问题三：晃动或偏移下视频前景目标提取

### 4.1 问题分析

要求对于镜头发生晃动或者偏移的时候,仍能采用有效的算法对前景进行提取。如果在晃动的视频上直接运用各类前景提取算法时,整个背景都会被误判为前景,达不到预期的效果。而镜头的晃动可以看做短时间内一种线性的变换,例如平移,尺度变化等。我们在运用前景提取算法前,可以先将视频进行稳像处理,获得比较稳定的背景。

### 4.2 基于电子稳像技术的求解方法

对于晃动或偏移下视频前景目标提取问题,本文认为首先应进行稳定性处理,消除或减轻图像序列帧间运动中的随机运动,从而得到稳定的视频图像。在视觉应用中以此来提高目标运动估计的精度,降低模型的复杂性,改善图像质量。视频图像稳定方法是利用计算机图像处理的方法,是直接利用像面的图像信息来确定视频连续图像序列的帧间位移,并对其进行补偿的方法。

#### 4.2.1 模型实现机制

电子稳像最基本的技术是像移补偿技术,其基本原理如下:如图 4.1 所示,摄像机连续两帧成像焦平面(第 K 和 K+1 帧),每一格代表一个像素。摄像机成像过程中,由于摄像机位置或参数发生变化,导致曝光时间内相机与目标存在相对运动,目标在焦平面上所成的像不是静止的,而是运动变化的,此即像移。从监视器上来看,像移使目标成像相互混叠,导致监视器图像抖动、模糊退化及分辨率下降,大大降低了视频图像质量。

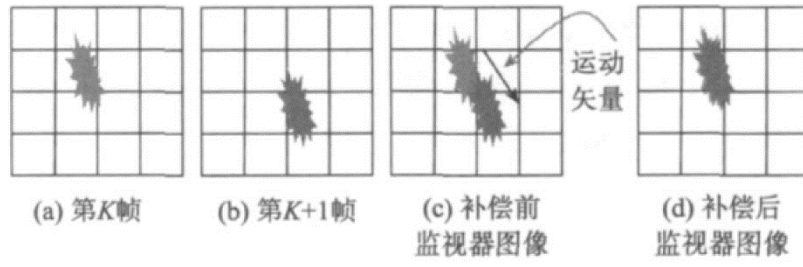


图 4.1 摄像机成像焦平面

对于面阵摄像机，它每一帧的图像信息是按行输出的，每行又是按照像元所排列的序号顺序输出的。因此，为了获得平稳、清晰的视频图像，首先检测出参考帧图像与当前帧图像之间的运动矢量，对运动矢量进行滤波、校正，转换为监视器图像的运动矢量，然后通过对 CCD 图像传感器的行、列序号重组，沿运动矢量反方向补偿第 K+1 帧图像，使监视器图像与第 K 帧图像近似相同或重合。

典型的稳像系统的结构如图 4.2 所示：

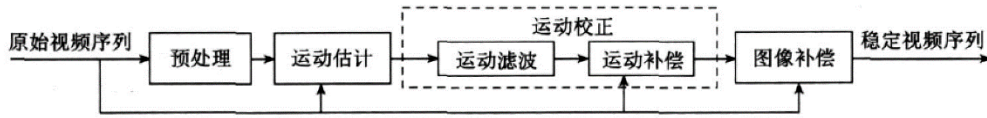


图 4.2 视频图像稳定系统结构图

#### 4.2.2 运动模型与运动估计

图像序列  $I(x, y, t)$  是三维场景投影到二维平面上产生的，每一帧对应于不同时刻同一场景的投影。所以在不同帧的图像像素之间存在着某种对应关系。人们用二维矢量场  $(u, v)$  来描述不同时刻  $t_1, t_2$  两帧之间的这种对应关系，表示如下：

$$I(x, y, t_1) = I(x + u, y + v, t_2) \quad (4.1)$$

我们把  $(u, v)$  称为运动场或位移矢量场，估算两帧之间的运动场的过程称为运动估计。实验中常采用逆向估计，即估算给定从  $t$  到  $t-1$ （一般对连续两帧处理取  $l=1$ ）的运动场。这样可以方便地由  $t-1$  时刻图像预测得到图像在  $t$  时刻图像  $I'(x, y, t_1)$ ，即有：

$$I'(x, y, t) = I(x + u, y + v, t - 1) \quad (4.2)$$

我们称这个预测过程为运动补偿。本文采用 6 参数透射模型对运动进行建模，描述背景的变化：

$$f(x, y, t) = f(m_1x + m_2y + m_5, m_3x + m_4y + m_6, t - 1) \quad (4.3)$$

其中，

$$A = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix}, \quad \vec{T} = \begin{pmatrix} m_5 \\ m_6 \end{pmatrix}$$

由式 (4.3) 可知，只需获得模型的 6 个参数就可知背景的运动信息，根据参数对整幅图像进行调整，最后再差分，在理想状态下得到的差分图像就是运动目标的边缘信息。

### 4.2.3 背景匹配

实验观察发现，在摄像机有抖动的情况下，只要摄像机的运动速度不至于过高，在适当的速度范围内（能满足实际的需要），序列图像帧与帧之间图像的背景变化不大，可以近似地或者直接认为连续两帧之间背景的运动变化可由一个 6 参数模型的仿射变换表示。另一方面，运动目标相对背景是一个小目标，也就是说，运动目标区域不能超过整个图像面积的 50%，这对于计算背景变换的仿射参数是很重要的。因为如果运动目标区域超过了背景区域，那么计算得到的仿射参数就不是背景的仿射参数，而是运动目标在图像中的变化的参数，这对于背景匹配没有丝毫作用。文提出了一种背景匹配结合帧间差分以解决在摄像机抖动情况下的运动目标分割。

仿射参数有 6 个未知数，理论上，只需要找出相对应的 3 对像素点  $(x_i(t-1), y_i(t-1))$  及  $(x_i(t), y_i(t))$ ，根据式 (4.3) 可求出仿射参数  $a = (a_0, \dots, a_5)^T$ ，其中， $(x_i(t), y_i(t))$  是像素点  $(x_i(t-1), y_i(t-1))$  在  $t$  时刻图像中的对应位置。然而由于图像本身的噪声以及匹配过程中的误差都会让参数  $a = (a_0, \dots, a_5)^T$  不能很好表示或者错误表示背景的变化情况。为此，本文在图像中寻找  $N$  个特征点，再取最小二乘的那三组特征点计算出的参数  $a = (a_0, \dots, a_5)^T$  为最后的结果。具体的步骤分为以下四步。

(1) 取  $N$  个特征点。选取这  $N$  个点的标准就是特征明显，这样匹配的效果才会好，否则只会给匹配结果带来噪声。点匹配本身就是易产生噪声的，本文中选取特征块匹配，然后再抽取特征块中心位置坐标作为点匹配的结果。设视频序列图像的相邻两帧为  $I(x, y, t-1)$  和  $I(x, y, t)$ 。将  $I(x, y, t-1)$  分成  $n \times n$  大小的子块  $W_{pq}^{t-1}$ 。匹配块的选择采用以下的评价函数：

$$f = \sum_{(x,y) \in W} [I(x, y, t-1) - \sum_{(x,y) \in W} I(x, y, t-1)/n^2] / n^2 \quad (4.4)$$

当  $f > \xi$  ( $\xi$  是阈值) 时，将此块标记为匹配块，否则为非匹配块。将  $N$  个匹配块分别在下一帧图像中用互相关匹配法寻找尺寸相同的最佳匹配块，取这  $N$  个对应匹配块的中心位置坐标，则得到  $N$  个特征点及其在下一帧中的匹配点  $(x_i(t-1), y_i(t-1))$  和  $(x_i(t), y_i(t))$ ， $i = 1, 2, \dots, N$ 。

(2) 在上一步找到的  $N$  个点中，用排列组合的方式任意选择 3 个点  $(x_i(t-1), y_i(t-1))$  和它们在下一帧中的对应点  $(x_i(t), y_i(t))$ ， $i=1,2,3$ 。然后由这三个点，根据式(4.3)计算仿射参数  $a = (a_0, \dots, a_5)^T$ 。

(3) 利用步骤 2 中获得的仿射参数  $a = (a_0, \dots, a_5)^T$  计算  $t-1$  时刻的  $N$  个特征点在  $t$  时刻图像中的位置坐标  $(x_i(t), y_i(t))$ ， $i = 1, 2, \dots, N$ 。比较  $N$  个特征点在  $t$  时刻图像中的匹配点  $(x_i(t), y_i(t))$  的值：

$$S = \sum_{i=1}^N [x_i(t) - x'_i(t)]^2 + [y_i(t) - y'_i(t)]^2 \quad (4.5)$$

(4) 重复步骤 2 和步骤 3，直到遍历  $N$  中的所有 3 个点的组合，分别依式 (4.5) 计算出  $S$  的值，取当  $S$  值最小时的那组数据求出的仿射参数  $a = (a_0, \dots, a_5)^T$ ，作为背景最后的仿射参数。

### 4.2.4 图像补偿

获得了前后两帧的背景运动参数  $a = (a_0, \dots, a_5)^T$  后，依式(4.3)把第  $t-1$  时刻图像中的像素  $(x', y')$  经过六参数仿射运动模型变换到第  $t$  时刻图像中对应的像素  $(x, y)$ ，即把图像  $I(x, y, t-1)$  逐像素点依式(4.3)计算其在  $t$  时刻的新坐标  $(x', y')$ ，

若计算得到的新坐标超出图像的大小范围，则将这一点抛弃，否则用该像素  $I(x, y, t - 1)$ ，得到图像  $I'(x, y, t - 1)$ 。具体如下式：

$$I'(x, y, t - 1) = I(x, y, t), \forall x \in (0, \text{Width}), \forall y \in (0, \text{Height})$$

若  $\forall x' \in (0, \text{Width})$  且  $y' \in (0, \text{Height})$

$$\text{则 } I'(x, y, t - 1) = I(x, y, t) \quad (4.6)$$

依次排列  $(x', y')$ ，则图像  $I((x', y', t - 1))$  就变成图像  $I'(x, y, t - 1)$ ，它的背景是图像  $I(x, y, t - 1)$  进行仿射变换得到的，所以它的背景理论上应该和图像  $I(x, y, t)$  的背景一致，把图像  $I'(x, y, t - 1)$  和  $I(x, y, t)$  作差分得到就是运动目标的区域。根据差分法对图像  $I'(x, y, t - 1)$  和  $I(x, y, t)$  进行差分运算，如下式：

$$D(x, y, t) = |I'(x, y, t - 1) - I(x, y, t)| \quad (4.7)$$

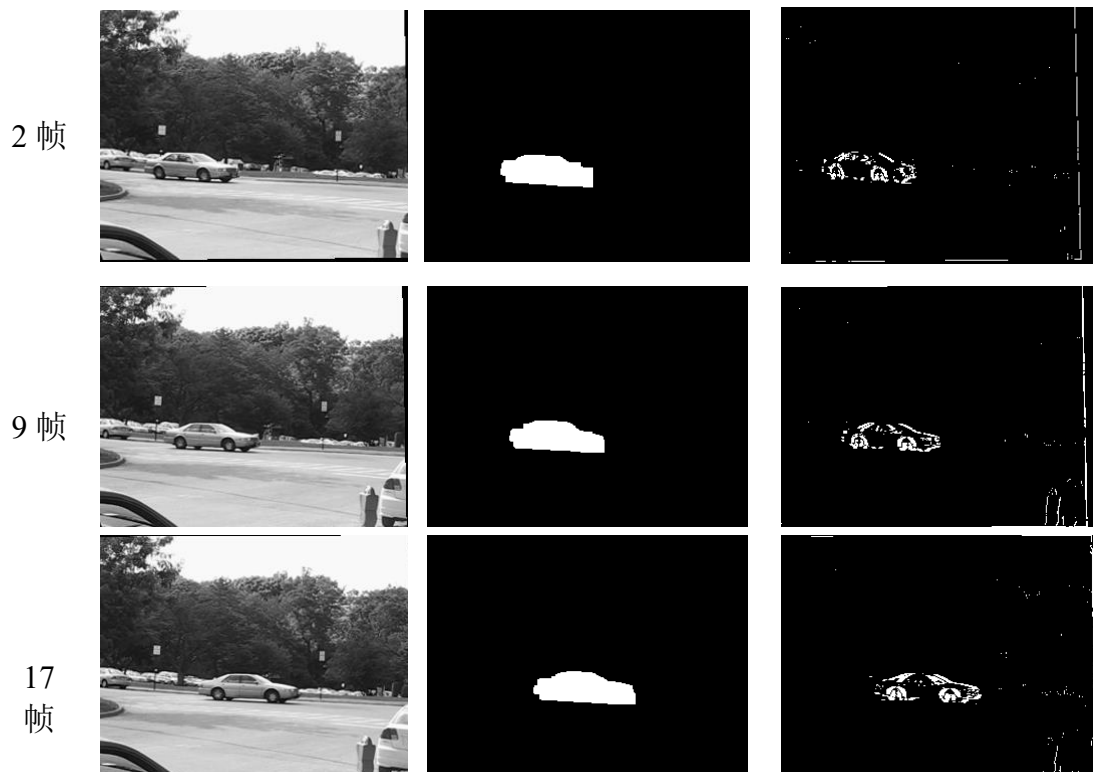
即可提取出背景基本上被消除，有显著前景目标的视频。

### 4.3 算法仿真与结果分析

本文使用上述模型对附件二中有晃动下的四个场景 (Cars6、Cars7、People1、People2) 均提取了前景目标，并选取关键帧 (如：有明显人物出现等) 进行结果分析。

#### 4.3.1 Cars6 场景中的前景目标提取

Cars6 场景视频分解共 30 帧，以下选取第 2、9、17、28 帧为关键帧来展示视频提取的效果。其中，第 2 帧为汽车位于画面约 1/5 处，第 9 帧为汽车位于画面约 2/5 处，第 17 帧为汽车位于画面约 1/2 处，第 28 帧为汽车位于画面约 4/5 处。选取该 4 帧作为关键帧可以较为明显的看出本模型在关键时点的前景目标提取效果。



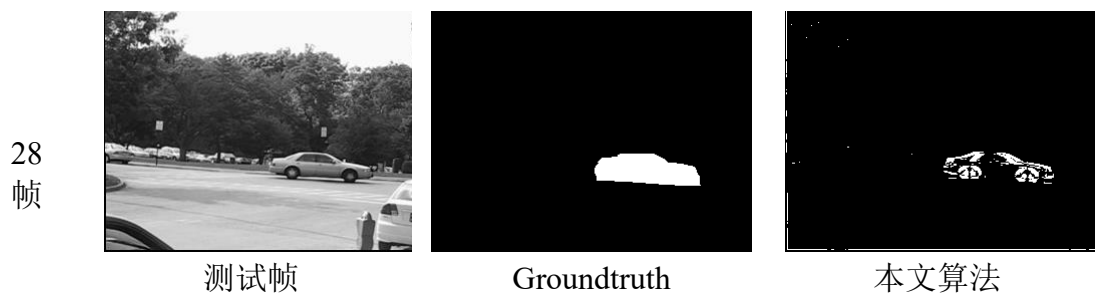


图 4.3 Cars6 场景中的前景目标提取

从提取效果上看，本文算法（对视频进行电子稳像技术处理）出现的噪声点较少，前景目标提取较为完整，能够看出前景目标轮廓，图像效果较为稳定。

#### 4.3.2 Cars7 场景中的前景目标提取

Cars7 场景视频分解共 24 帧，以下选取第 2、8、19、22 帧为关键帧来展示视频提取的效果。

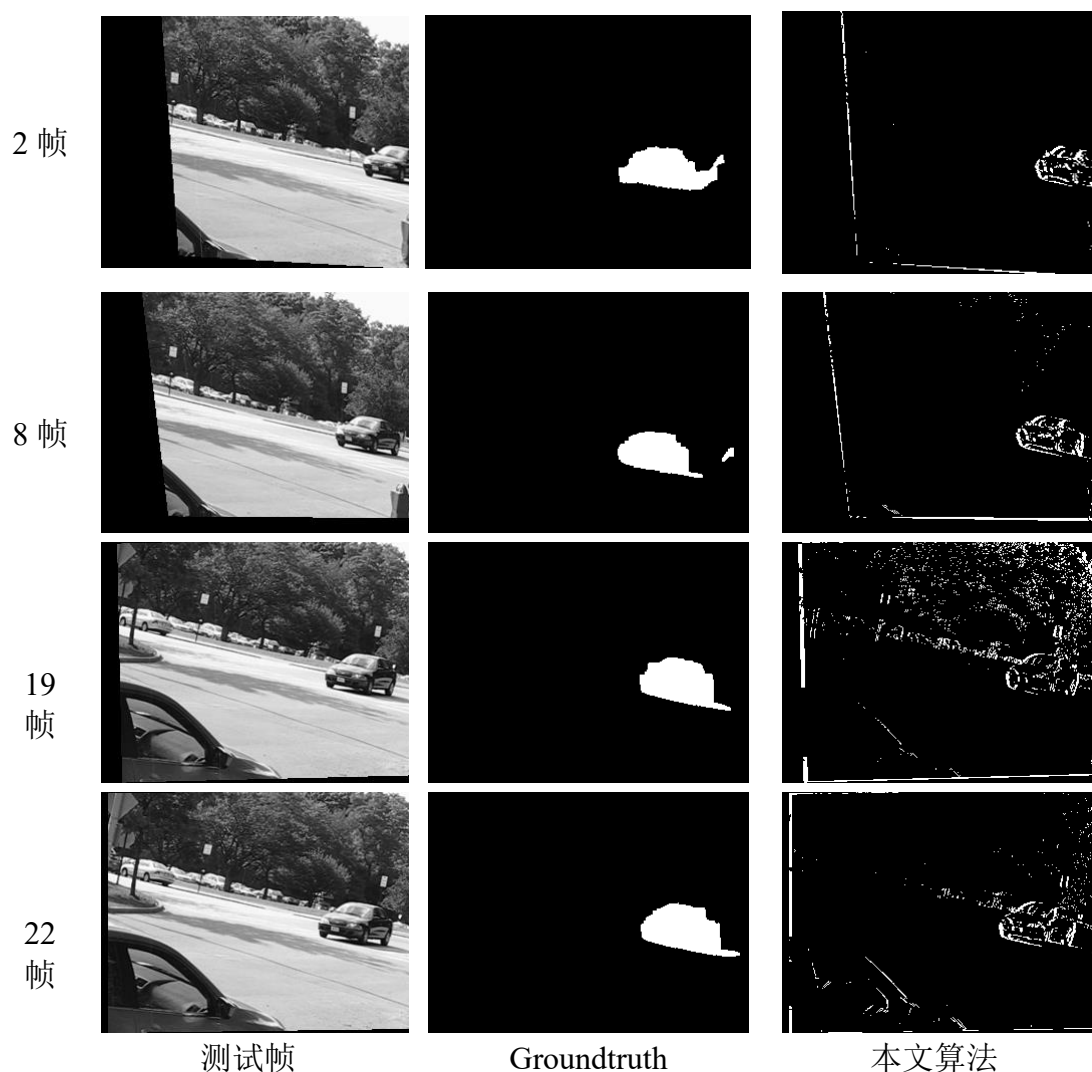


图 4.4 Cars7 场景中的前景目标提取

从提取效果上看，本文算法（对视频进行电子稳像技术处理）出现的噪声点较少，前景目标提取较为完整，能够看出前景目标轮廓，图像效果较为稳定。

### 4.3.3 People1 场景中的前景目标提取

People1 场景视频分解共 40 帧，以下选取第 2、18、28、38 帧为关键帧来展示视频提取的效果。

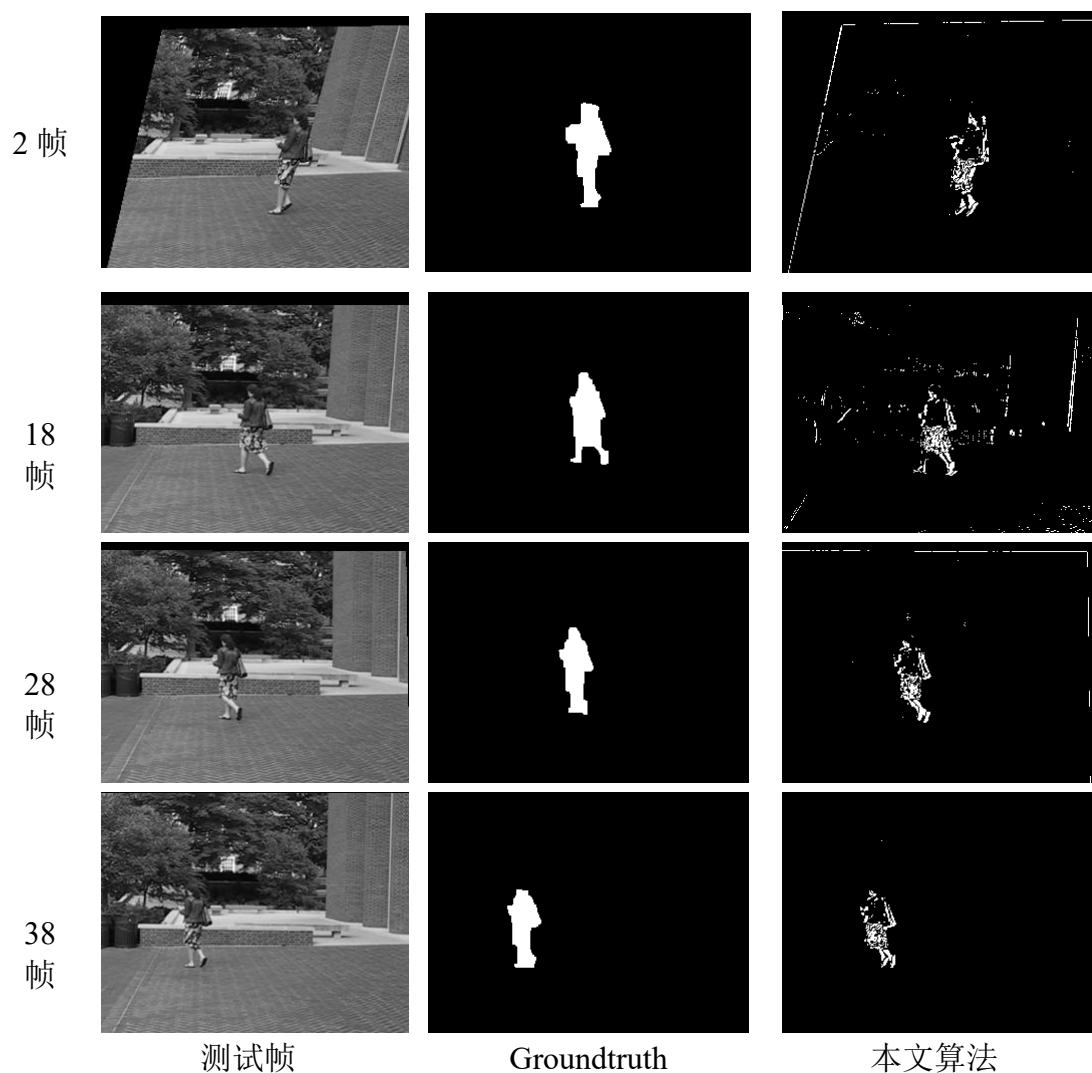


图 4.5 People1 场景中的前景目标提取

从提取效果上看，本文算法（对视频进行电子稳像技术处理）出现的噪声点较少，前景目标提取较为完整，能够看出前景目标轮廓，图像效果较为稳定。

### 4.3.4 People2 场景中的前景目标提取

People2 场景视频分解共 30 帧，以下选取第 2、9、17、28 帧为关键帧来展示视频提取的效果。其中，第 2 帧为二人大致处于同一水平线，第 9 帧为一人在前，一人在后，第 17 帧为一人即将离开画面，第 28 帧为画面中仅剩一人。选取该 4 帧作为关键帧可以较为明显的看出本模型在关键时点的前景目标提取效果。

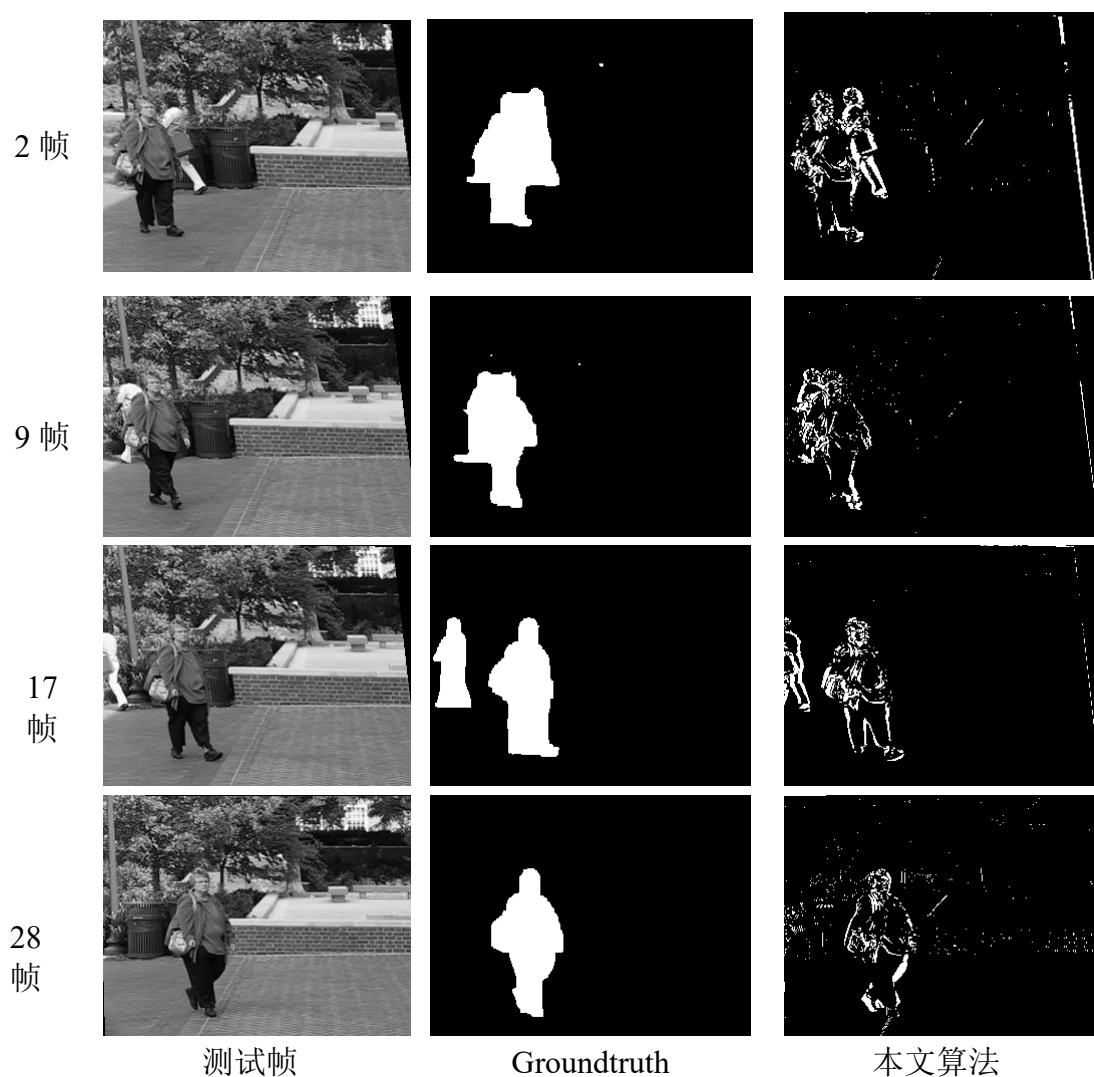


图 4.6 People2 场景中的前景目标提取

从提取效果上看，本文算法（对视频进行电子稳像技术处理）出现的噪声点较少，前景目标提取较为完整，能够看出前景目标轮廓，图像效果较为稳定。

#### 4.4 实验验证与比对

##### 4.4.1 Cars6 场景中的实验验证与比对

###### 4.4.1.1 Cars6 场景中的实验效果对比

除对视频进行电子稳像技术处理算法外，我们还采用了未经电子稳像技术处理的混合高斯模型和 ViBe 模型提取前景目标，效果对比如下：



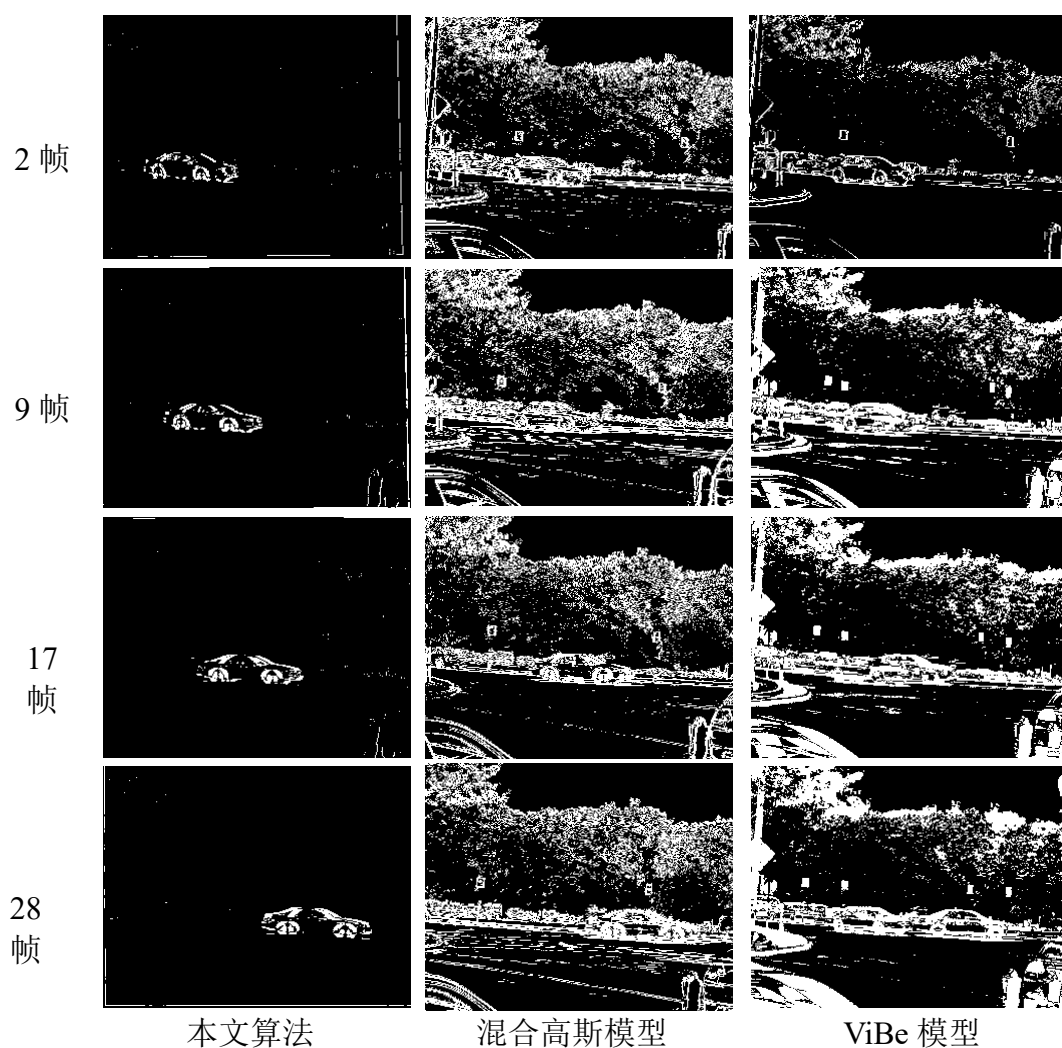


图 4.7 Cars6 场景中的不同模型实验效果对比

从对比图上看，对于有晃动的视频进行前景提取时，混合高斯模型存在着噪声点较多、背景剔除不彻底等问题，ViBe 模型虽然噪声点相对较少，但在背景提出和完整性上不及经电子稳像算法处理后的效果，从图形对比效果上看，电子稳像算法在解决有晃动的前景目标提取问题时效果更好。

#### 4.4.1.2 Cars6 场景中的评价指标对比

使用前文提到的评价指标计算出结果如下图所示，可以看出，电子稳像算法在准确度（Precision）、特指度（Specificity）、错误分类率（PWC）、F 值（F-Measure）等方面均显著优于其他模型。再次印证了在解决有晃动的前景提取问题方面，电子稳像算法确实为最优模型。

表 4.1 模型评价指标数值表

	本文算法	混合高斯模型	ViBe模型
TP	17201	40955	44538
FP	18607	405065	484911
TN	2139065	1752607	1672761
FN	52327	28573	24990
Recall	0.2474	0.5890	0.6406
Precision	0.4804	0.0918	0.0841
Specificity	0.9920	0.9772	0.9741
FPR	0.0086	0.1877	0.2247
FNR	0.7526	0.4110	0.3594
PWC	0.0318	0.1947	0.2289
F-Measure	0.3266	0.1589	0.1487

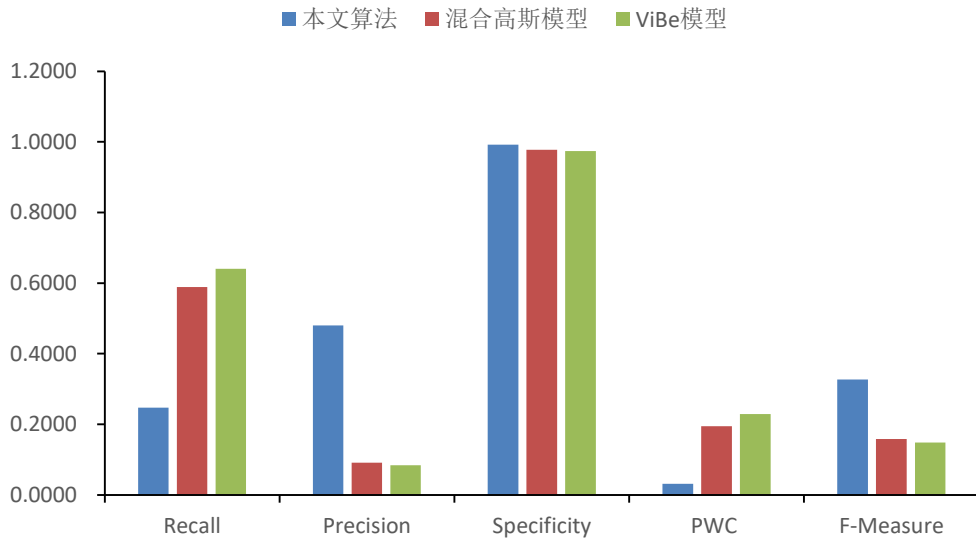


图 4.8 模型主要评价指标直方图

#### 4.4.2 Cars7 场景中的实验验证与对比

##### 4.4.2.1 Cars7 场景中的实验效果对比

除对视频进行电子稳像技术处理算法外,我们还采用了未经电子稳像技术处理的混合高斯模型和 ViBe 模型提取前景目标,效果对比如下:

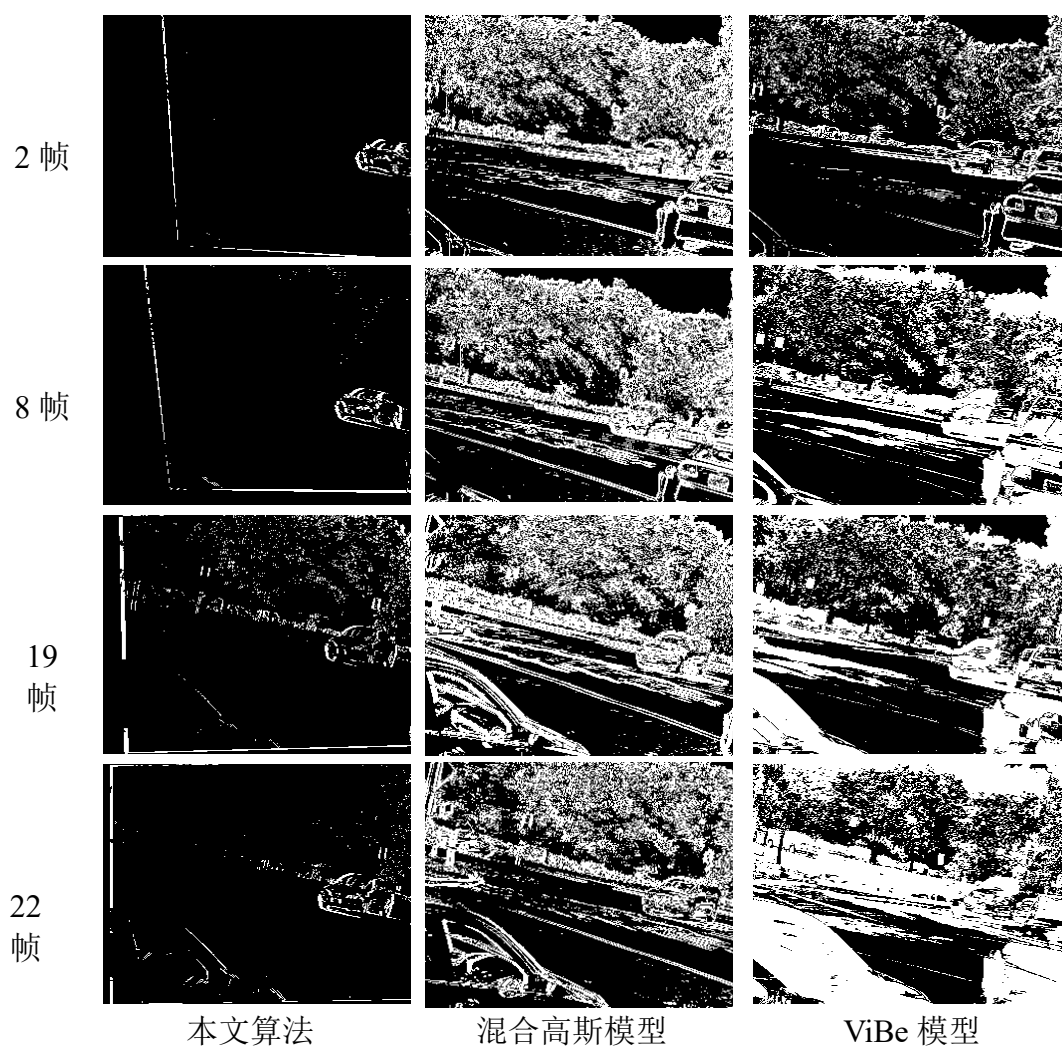


图 4.9 Cars7 场景中的不同模型实验效果对比

从对比图上看，对于有晃动的视频进行前景提取时，混合高斯模型存在着噪声点较多、背景剔除不彻底等问题，ViBe 模型虽然噪声点相对较少，但在背景提出和完整性上不及经电子稳像算法处理后的效果。从图形对比效果上看，电子稳像算法在解决有晃动的前景目标提取问题时效果更好。

#### 4.4.2.2 Cars7 场景中的评价指标对比

使用前文提到的评价指标计算出结果如下图所示，可以看出，电子稳像算法在准确度（Precision）、特指度（Specificity）、错误分类率（PWC）、F 值（F-Measure）等方面均显著优于其他模型。再次印证了在解决有晃动的前景提取问题方面，电子稳像算法确实为最优模型。

表 4.2 模型评价指标数值表

	本文算法	混合高斯模型	ViBe模型
TP	8382	43915	44500
FP	114865	689080	739525
TN	1589449	1015234	964789
FN	53704	18171	17586
Recall	0.1350	0.7073	0.7167
Precision	0.0680	0.0599	0.0568
Specificity	0.9948	0.9585	0.9559
FPR	0.0674	0.4043	0.4339
FNR	0.8650	0.2927	0.2833
PWC	0.0954	0.4004	0.4286
F-Measure	0.1905	0.1105	0.1052

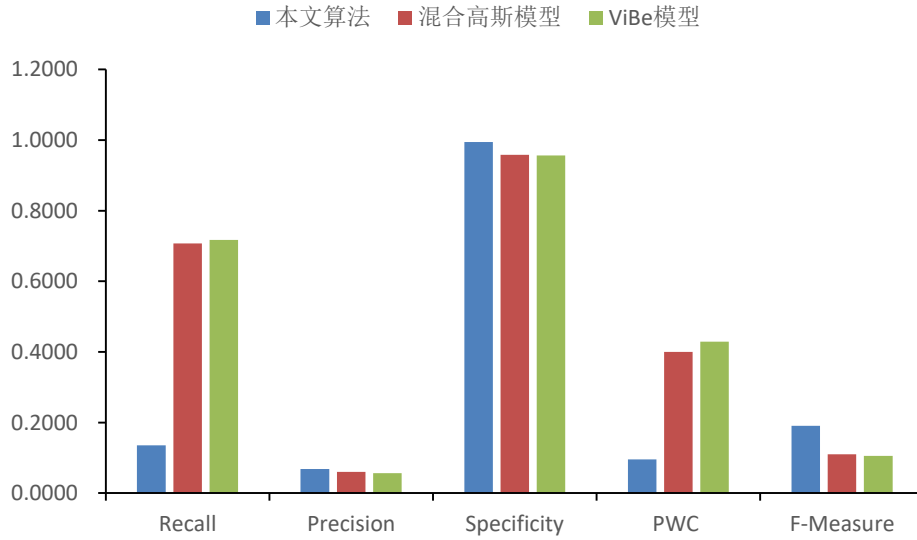


图 4.10 模型主要评价指标直方图

### 4.4.3 People1 场景中的前景目标提取

#### 4.4.3.1 People1 场景中的实验效果对比

除对视频进行电子稳像技术处理算法外,我们还采用了未经电子稳像技术处理的混合高斯模型和 ViBe 模型提取前景目标,效果对比如下:

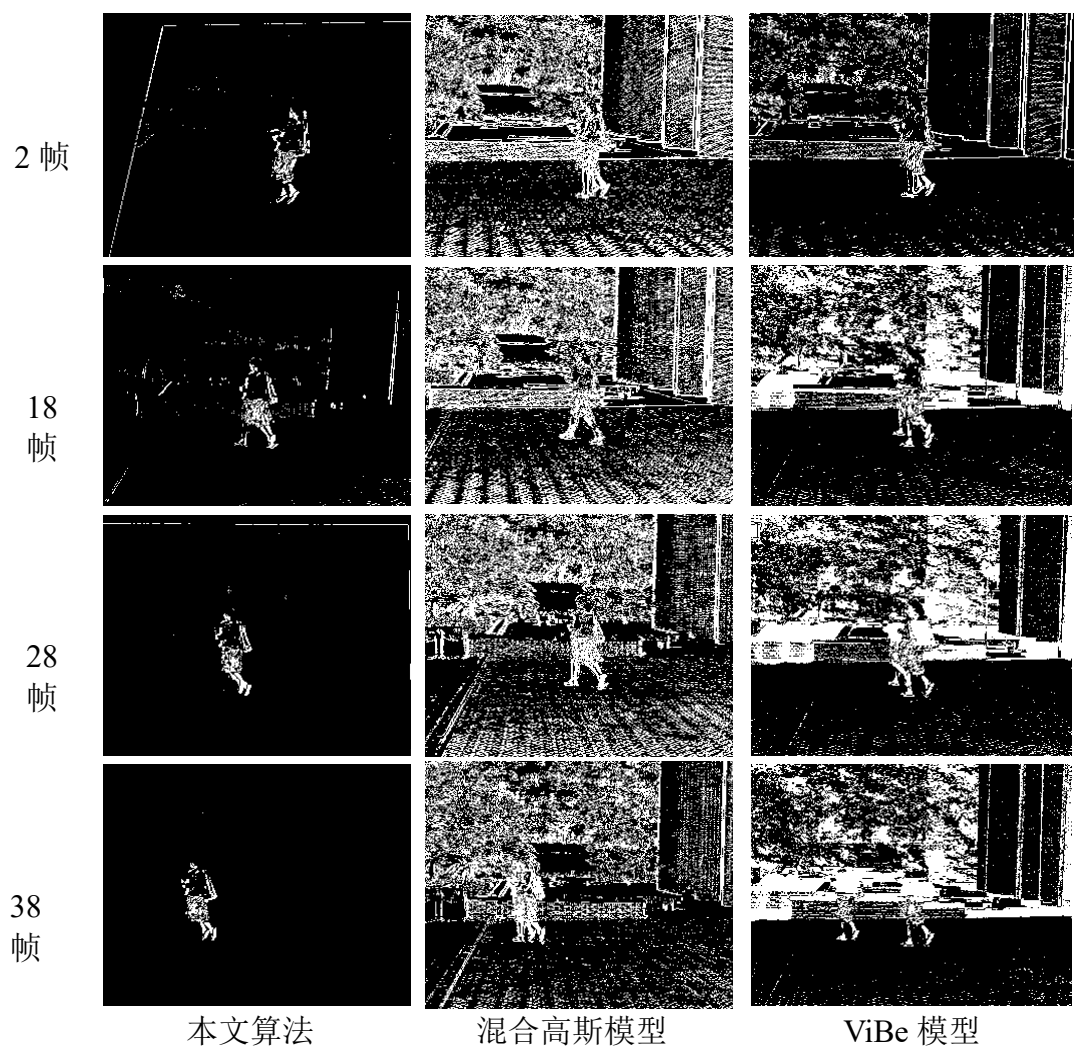


图 4.11 People1 场景中的不同模型实验效果对比

从对比图上看，对于有晃动的视频进行前景提取时，混合高斯模型存在着噪声点较多、背景剔除不彻底等问题，ViBe 模型虽然噪声点相对较少，但在背景提出和完整性上不及经电子稳像算法处理后的效果。从图形对比效果上看，电子稳像算法在解决有晃动的前景目标提取问题时效果更好。

#### 4.4.3.2 People1 场景中的评价指标对比

使用前文提到的评价指标计算出结果如下图所示，可以看出，电子稳像算法在准确度（Precision）、特指度（Specificity）、错误分类率（PWC）、F 值（F-Measure）等方面均显著优于其他模型。再次印证了在解决有晃动的前景提取问题方面，电子稳像算法确实为最优模型。

表 4.3 模型评价指标数值表

	本文算法	混合高斯模型	ViBe模型
TP	13129	44952	39779
FP	61330	947330	743223
TN	2857036	1971036	2175143
FN	63705	31882	37055
Recall	0.1709	0.5851	0.5177
Precision	0.1763	0.0453	0.0508
Specificity	0.9954	0.9777	0.9820
FPR	0.0210	0.3246	0.2547
FNR	0.8291	0.4149	0.4823
PWC	0.0417	0.3269	0.2605
F-Measure	0.1736	0.0841	0.0925

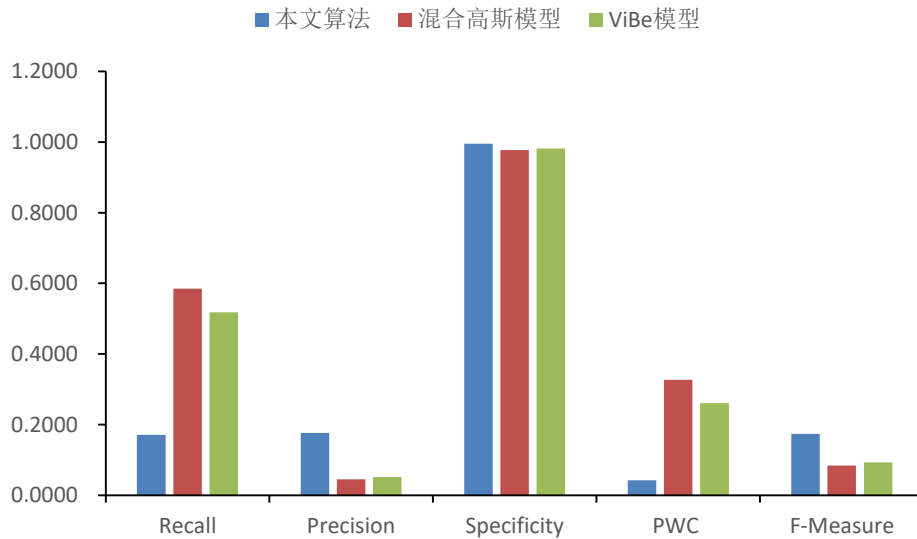


图 4.12 模型主要评价指标直方图

#### 4.4.4 People2 场景中的前景目标提取

##### 4.4.4.1 people2 场景中的实验效果对比

除对视频进行电子稳像技术处理算法外,我们还采用了未经电子稳像技术处理的混合高斯模型和 ViBe 模型提取前景目标,效果对比如下:

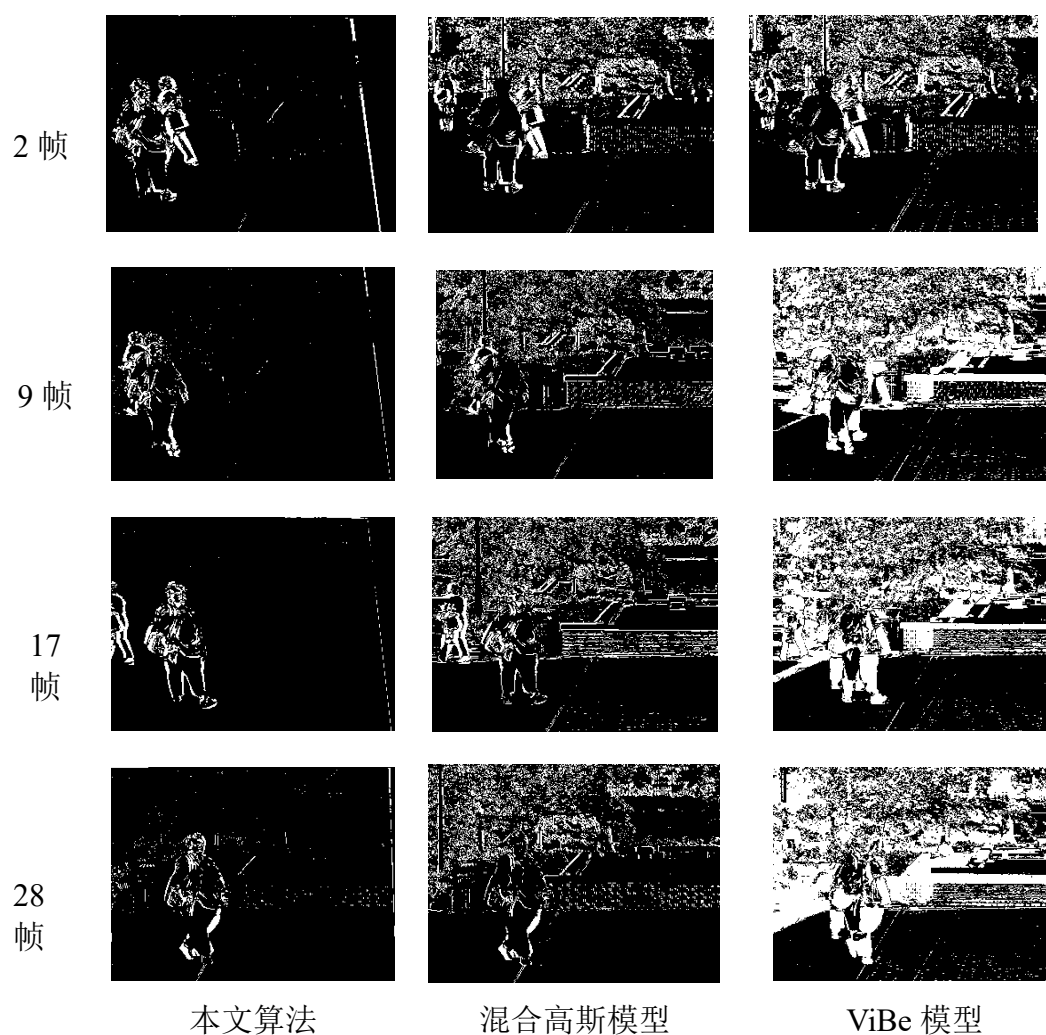


图 4.13 People2 场景中的不同模型实验效果对比

从对比图上看，对于有晃动的视频进行前景提取时，混合高斯模型存在着噪声点较多、背景剔除不彻底等问题，ViBe 模型虽然噪声点相对较少，但在背景提出和完整性上不及经电子稳像算法处理后的效果。从图形对比效果上看，电子稳像算法在解决有晃动的前景目标提取问题时效果更好。

#### 4.4.4.2 People2 场景中的评价指标对比

使用前文提到的评价指标计算出结果如下图所示，可以看出，电子稳像算法在准确度（Precision）、特指度（Specificity）、F 值（F-Measure）等方面均显著优于其他模型。再次印证了在解决有晃动的前景提取问题方面，电子稳像算法确实为最优模型。

表 4.4 模型评价指标数值表

	本文算法	混合高斯模型	ViBe模型
TP	27661	44398	99240
FP	110998	329113	723319
TN	1927091	1708976	1314770
FN	161450	144713	89871
Recall	0.1463	0.2348	0.5248
Precision	0.1995	0.1189	0.1206
Specificity	0.9858	0.9747	0.9298
FPR	0.0545	0.1615	0.3549
FNR	0.8537	0.7652	0.4752
PWC	0.1223	0.2127	0.3651
F-Measure	0.1688	0.1578	0.1962

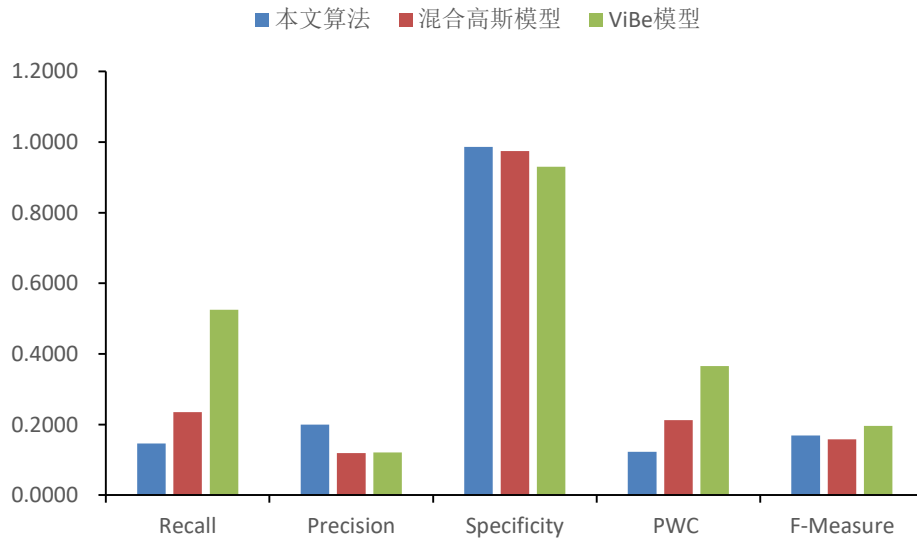


图 4.14 模型主要评价指标直方图

## 5.问题四：给定视频显著前景目标帧号

### 5.1 问题分析

问题四要求对给定的 8 组视频，利用构造的模型对找出显著前景的帧号。前面构造的模型包含了静态背景、动态背景以及摄像头晃动的情况，因此我们可以将各个场景进行分类采用不同模型进行求解。在完成前景提取，获得黑白 Mask 帧后，可以计算前后帧差值较大的作为有显著目标进入或退出视野的帧号。或者通过计算各帧的像素密度来判断是否有显著目标。



## 5.2 场景分类与模型应用

根据前文建模结果，对于不带晃动-静态背景视频的前景目标提取使用混合高斯模型效果最好，对于不带晃动-动态背景视频的前景目标提取使用 ViBe 模型效果最好。基于此，我们首先将附件三中的 8 组视频根据背景进行了分类，其中 Campus、Curtain、Escalator、Fountain 和 Overpass 属于包含动态背景信息的监控视频，对这 5 组视频我们采用了 ViBe 模型进行前景提取。Hall、Lobby 和 Office 属于包含静态背景信息的监控视频，对这三组视频我们采用了混合高斯模型进行前景提取。

## 5.3 帧数提取

首先我们总结出各个场景中显著目标进入及退出的帧号如下表所示：

表 5.1 不同场景下前景目标帧号提取

场景	模型	帧号
Campus	ViBe	200-225; 306-523; 351-392; 600; 643-682; 690-712; 737-875; 810-834; 759-905; 1001-1035; 1264; 1325-1355; 1343-1375; 1377-1406
Curtain	ViBe	967; 1756-1908; 2126; 2171-2318; 2642; 2765-2932
Escalator	ViBe	2-175; 220-2392; 2415; 2754; 2761-3417
Fountain	ViBe	141; 153-212; 259; 335; 403-523
Hall	混合高斯	2-141; 219-281; 578; 634-698; 819-848; 1154-1203; 1278-1375; 1380-1510; 1518-1543; 1556-1595; 1656-1706; 1859-2270; 2378-2585; 3025-3253; 3255-3318; 3467-3534
Lobby	混合高斯	79; 154-196; 259; 346-393; 521; 623-667; 870; 963-1037; 1161; 1239-1283; 1334-1538
Office	混合高斯	197; 372; 501; 581-2042; 2080
Overpass	ViBe	374; 547-636; 968; 1551; 1881; 2098; 2335-2965

接下来，我们将提取出的前景目标予以展示，仅在一帧中出现的前景目标我们将其列出，在多帧中均有出现的前景目标我们选择关键帧列出，具体结果如下。

### 5.3.1 包含动态背景信息的五种场景

#### 5.3.1.1 Campus 场景下前景目标的帧号提取

场景为 Campus，使用 ViBe 模型提取前景目标时，显著目标出现帧数及提取结果如下表所示：

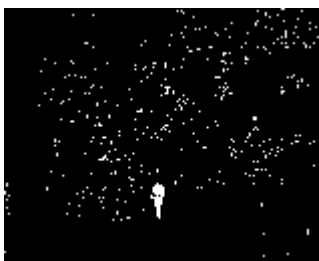
帧号

提取结果

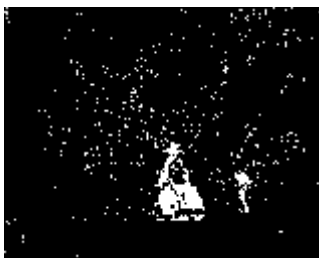
200-  
225



306-  
523



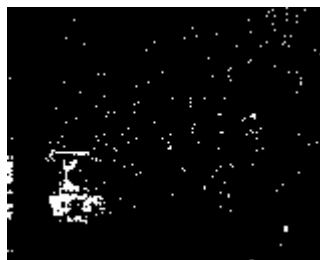
351-  
392



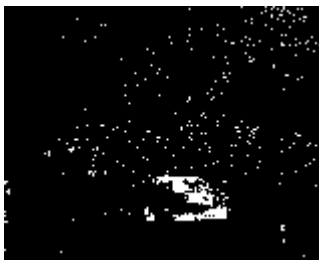
600



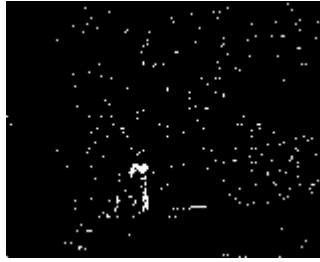
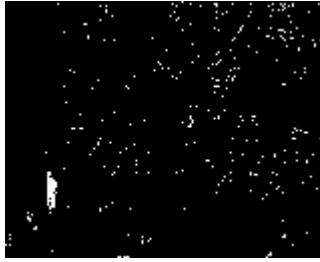
643-  
682



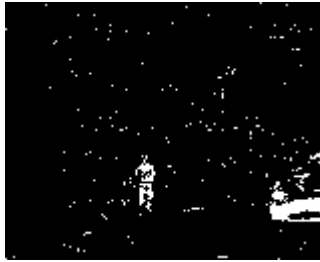
690-  
712



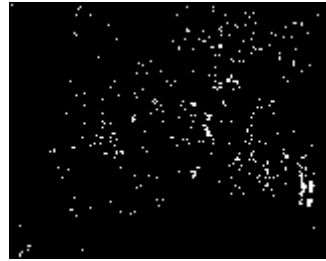
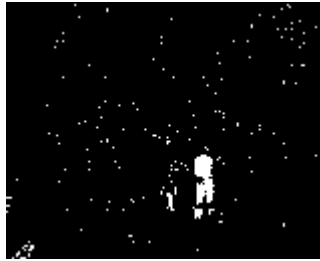
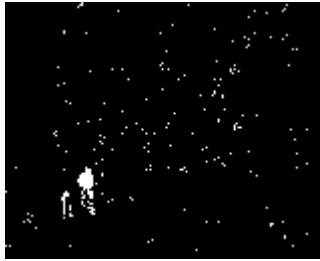
737-  
875



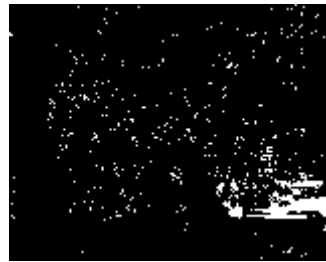
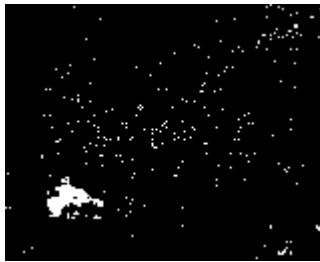
810-  
834



759-  
905



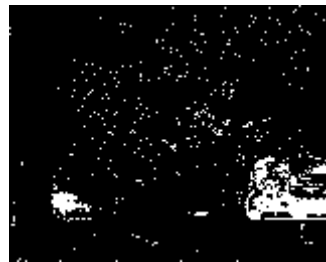
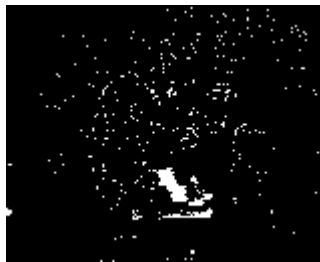
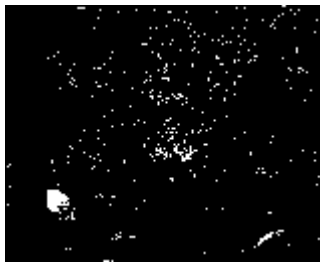
1001  
-  
1035



1264



1325  
-  
1355



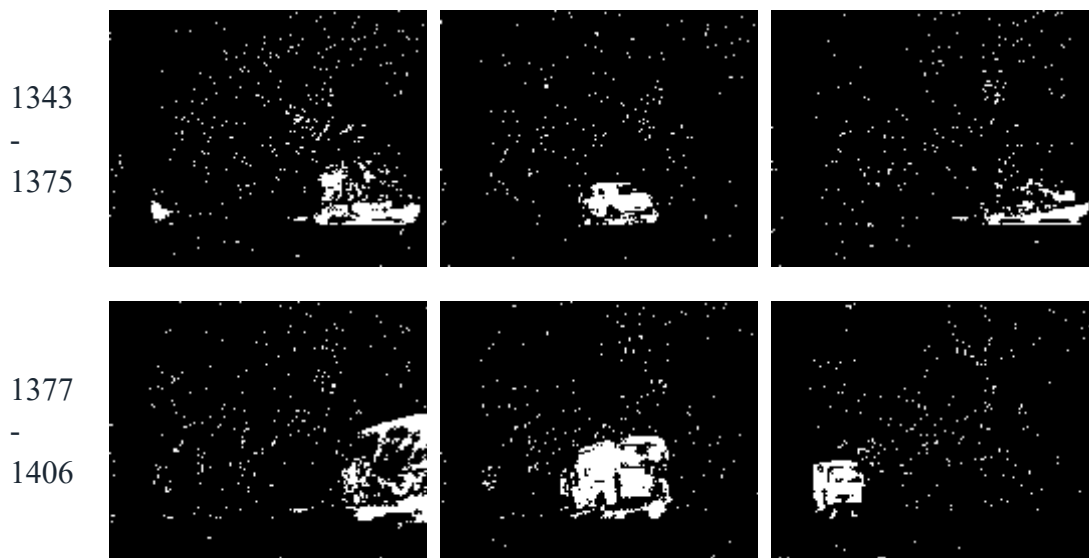


图 5.1 Campus 场景下前景目标的帧号提取

#### 5.3.1.2 Curtain 场景下前景目标的帧号提取

场景为 Curtain，使用 ViBe 模型提取前景目标时，显著目标出现帧数及提取结果如下表所示：


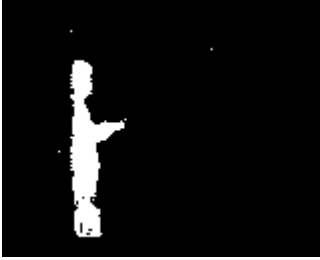






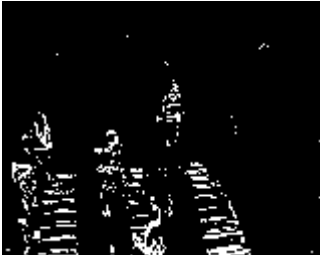
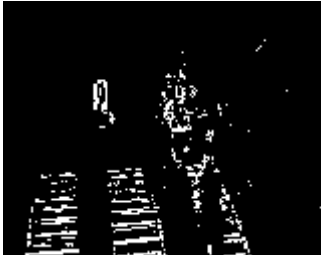
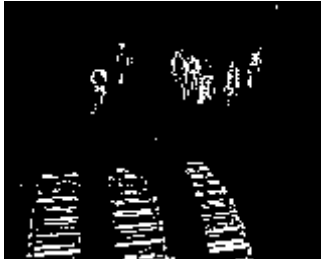
帧号	提取结果		
967			
1756 - 1908			
2126			



图 5.2 Curtain 场景下前景目标的帧号提取

### 5.3.1.3 Escalator 场景下前景目标的帧号提取

场景为 Escalator，使用 ViBe 模型提取前景目标时，显著目标出现帧数及提取结果如下表所示：

帧号	提取结果		
2- 175			
220- 2392			

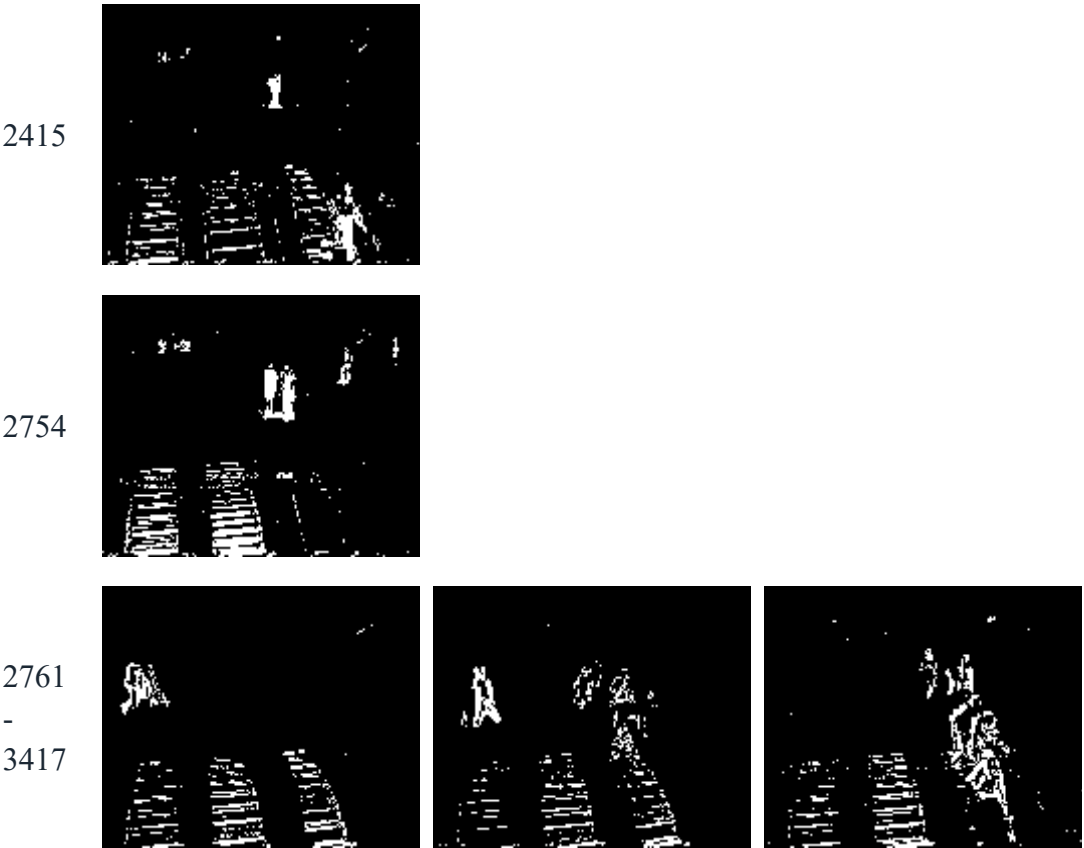


图 5.3 Escalator 场景下前景目标的帧号提取

5.3.1.4 Fountain 场景下前景目标的帧号提取

场景为 Fountain，使用 ViBe 模型提取前景目标时，显著目标出现帧数及提取结果如下表所示：

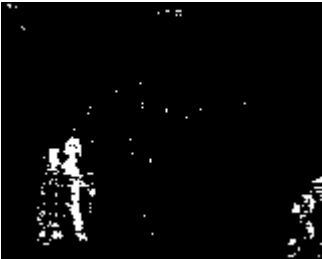
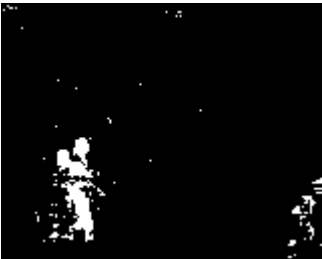

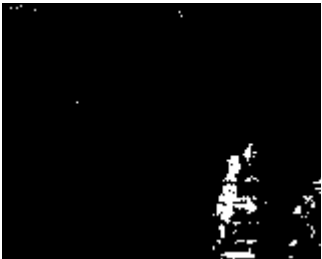
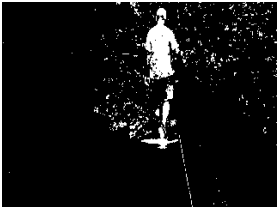




帧号	提取结果		
141			
153- 212			



图 5.4 Fountain 场景下前景目标的帧号提取

### 5.3.1.5 Overpass 场景下前景目标的帧号提取

场景为 Overpass，使用 ViBe 模型提取前景目标时，显著目标出现帧数及提取结果如下表所示：

帧号	提取结果		
374			
547-636			
968			

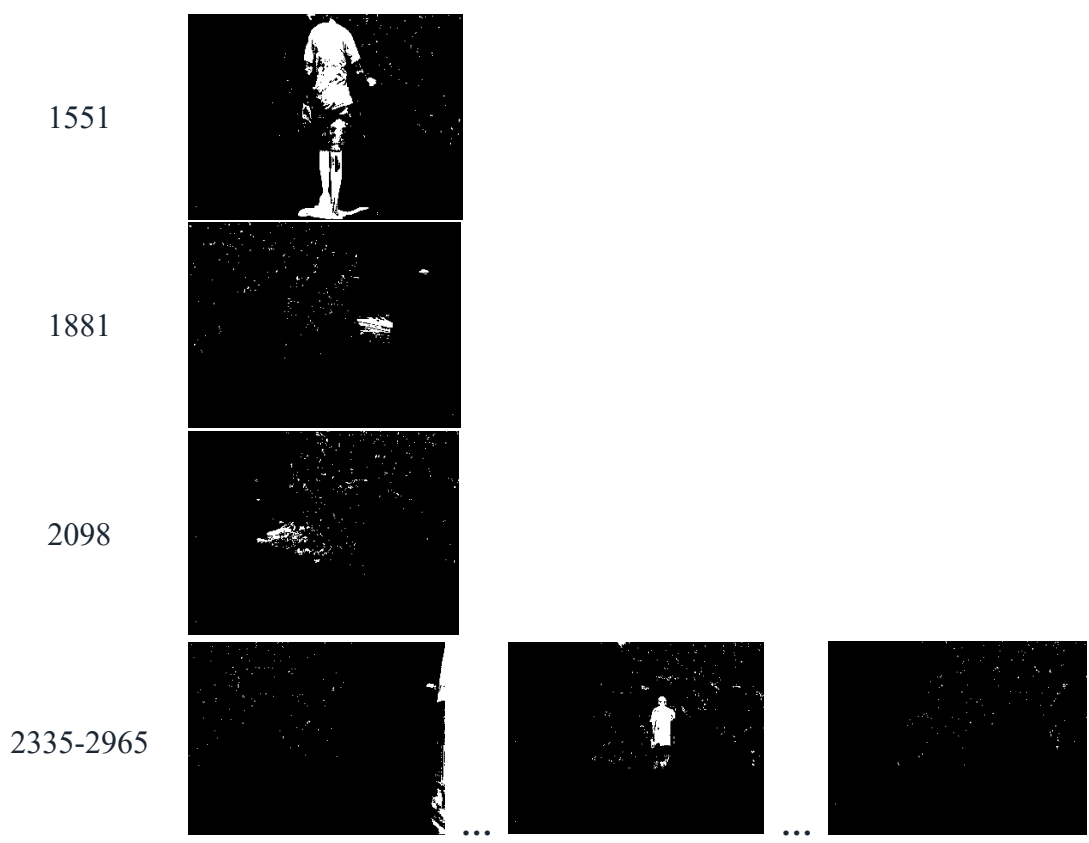



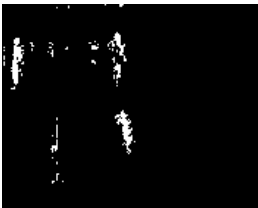


图 5.5 Overpass 场景下前景目标的帧号提取

### 5.3.2 包含静态背景信息的三种场景

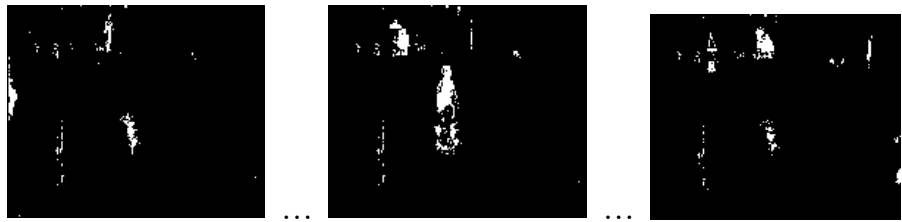
#### 5.3.2.1 Hall 场景下前景目标的帧号提取

场景为 Hall，使用混合高斯模型提取前景目标时，显著目标出现帧数及提取结果如下表所示：

帧号	提取结果		
2-141		...	 ... 
578			



634-700



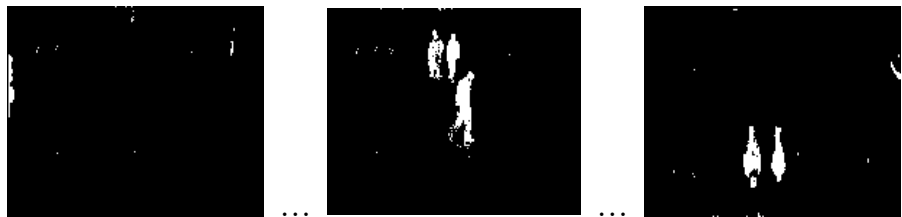
818-847



1153-1208



1278-1375



1378-1509



1517-1543



1556-1596


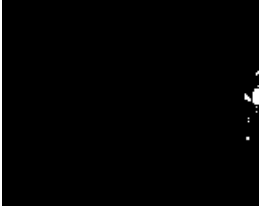

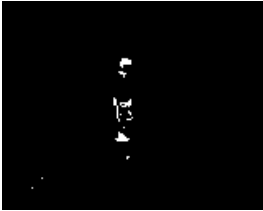
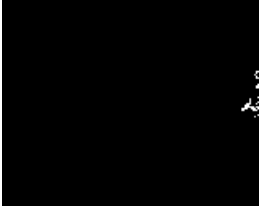
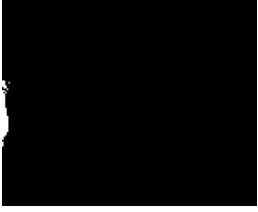



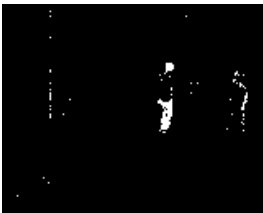




图 5.6 Hall 场景下前景目标的帧号提取

#### 5.3.2.2 Lobby 场景下前景目标的帧号提取

场景为 Lobby，使用混合高斯模型提取前景目标时，显著目标出现帧数及提取结果如下表所示：

帧号	提取结果		
79			
154-196		...	
259			
345-393		...	
521			
632-655		...	
870			

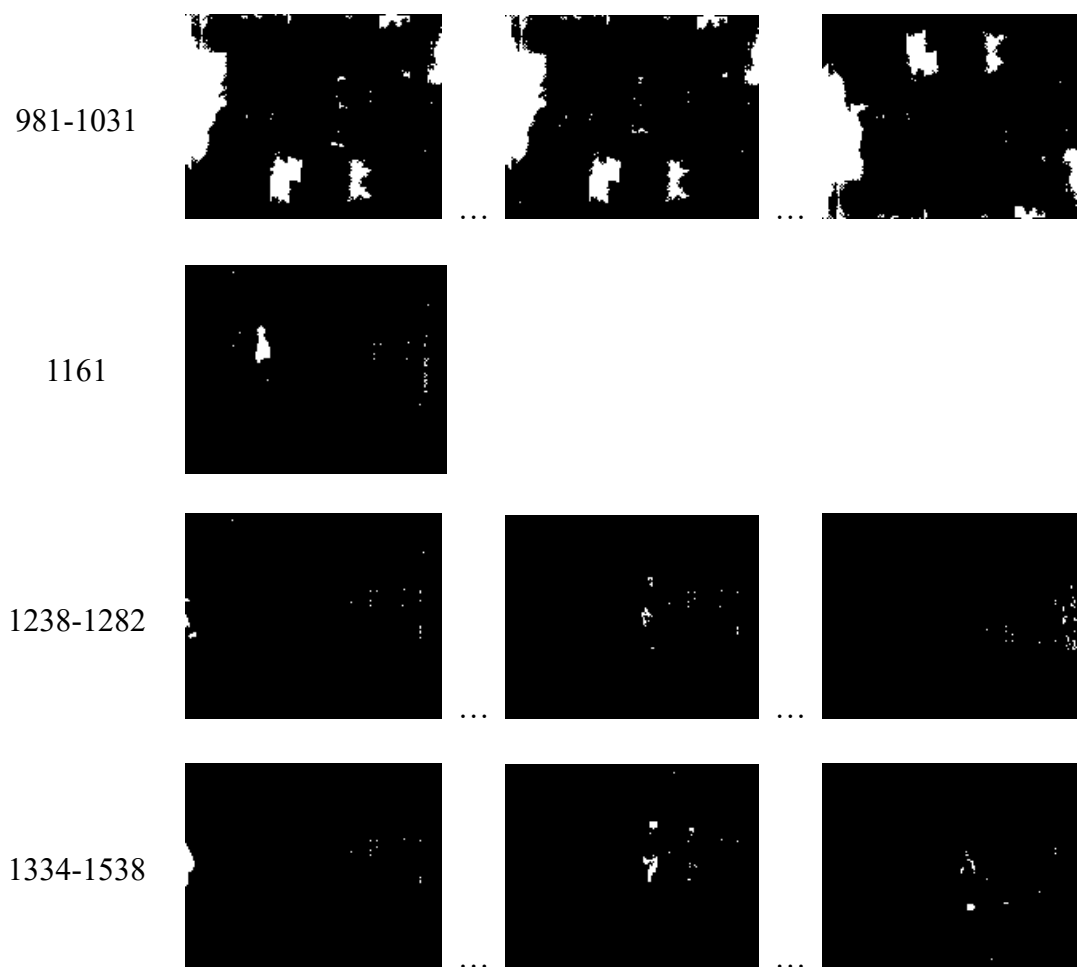


图 5.7 Lobby 场景下前景目标的帧号提取

### 5.3.2.3 Office 场景下前景目标的帧号提取

场景为 Office，使用混合高斯模型提取前景目标时，显著目标出现帧数及提取结果如下表所示：

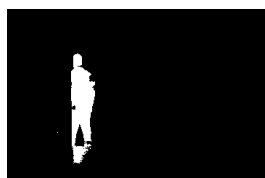
帧号

提取结果

197



372



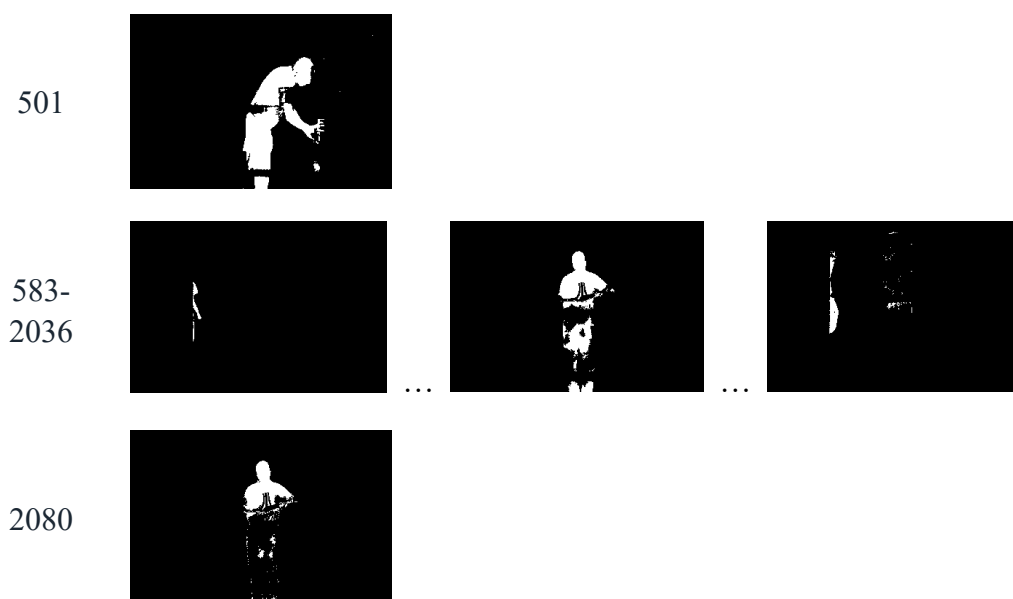


图 5.8 Office 场景下前景目标的帧号提取

## 6.问题五：多镜头下目标前景的提取

### 6.1 问题分析

对于问题五，要求在对同一地点不同角度的多个监控视频上进行目标前景提取。当单个摄像机中的目标检测和跟踪问题解决，扩展到多个摄像机的时候，遇到的主要难点是如何确定在不同摄像机中目标的观测，是否来源于同一个目标的问题。因为世界坐标系中的真实目标，会在不同的摄像机视图中产生不同的观测。确定不同摄像机中目标观测的对应关系称为匹配问题，是多镜头下跟踪和识别的研究要点。

### 6.2 基于几何约束的目标匹配

判断目标是否匹配，可以利用的约束条件可以分为三类：

（1）目标自身特征的一致性，即目标本身的特性在穿越多个摄像机视野时是不变的，如目标的颜色、形状等外观特征。但由于不同的摄像机的视角不同、场景光照的不同，这类特征在不同的摄像机视图中的观测存在着较大的差异，寻找这些差异中的规律性，如不同视图间亮度转移关系，仍是一个较难的问题。这类特征在目标匹配问题中应用较多，但准确度不高，算法耗时随着特征维数成级数增加。

（2）摄影几何约束，如空间目标的 3D 方位、对极约束、共面约束等，这类约束比较严格，有良好的数学基础，通过对单个摄像机的标定或多个摄像机间关

系的确定，能够准确的定位目标的空间位置，这类约束多应用于视野有重叠的问题中。

(3) 另外还有一类比较松的约束条件，如目标在不同摄像机之间运动的路径拓扑结构，目标进入或离开视图大致位置的分布，以及目标穿越摄像机视野的大致时间分布等，这类约束随机性强，一般需要通过大量观测数据进行统计学习来确定，多用于无法获得严格摄影几何约束的视野无重叠的问题上。

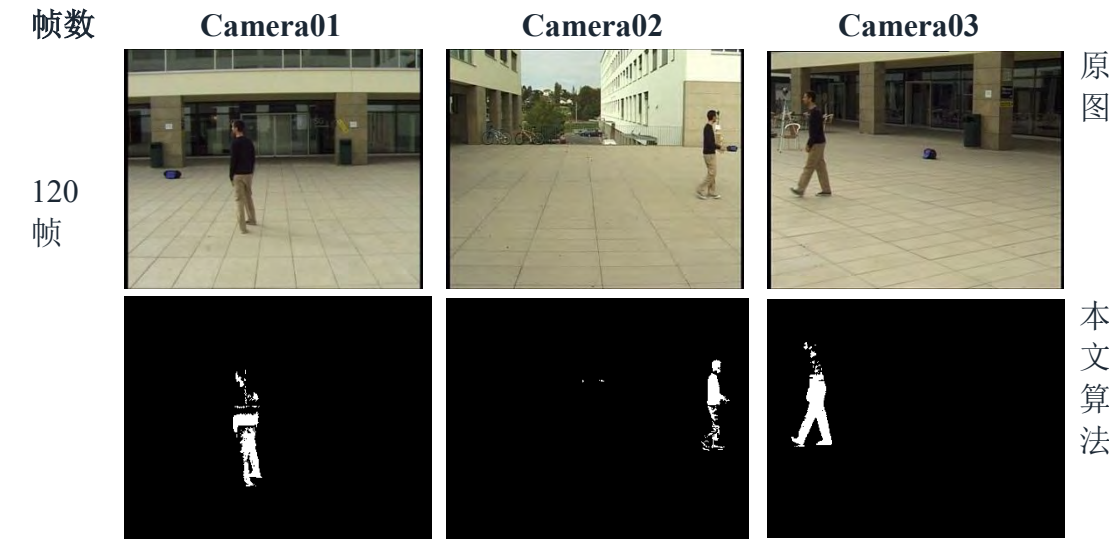
由于本题采用的数据集为洛桑联邦理工学院 CVLAB 实验室提供的公开数据集，该数据集中对每个相机的位置都有描述，因此我们可以通过相机间的几何约束来进行坐标校正，从而完成目标匹配的功能。

6.3 算法仿真与结果分析

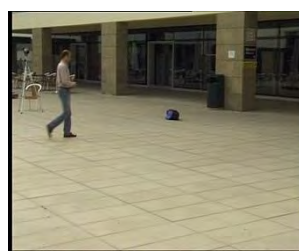
本文选取 Campus 和 Laboratory 两个场景分析多角度视频的前景目标提取问题。其中，Campus 场景为同一时间从三个不同角度拍摄同一地点的视频，Laboratory 场景为同一时间从四个不同角度拍摄同一地点的视频。

6.3.1 Campus 场景下多角度视频的前景目标提取

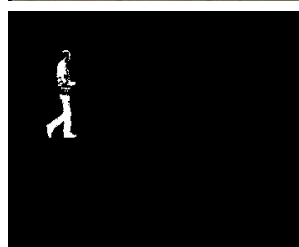
Campus 场景为同一时间从三个不同角度拍摄同一地点的视频，对于三个摄像机镜头我们分别命名为 Camera01、Camera02、Camera03。对于 Campus 场景我们选取以下关键帧：120 帧，第一个显著目标（男）出现在场景中；260 帧，第二个显著目标（男）走入场景；350 帧，第三个显著目标（男）出现在场景中；376 帧，第四个显著目标（女）出现在场景中但仅在 Camera02、Camera03 中有显示；490 帧，第四个显著目标出现在 Camera01 中；660 帧，第四个显著目标走出场景，第一到三个显著目标（男）聚集在场景中；900 帧，第五个显著目标（男）出现在场景中；999 帧，视频最后一帧，第五个显著目标即将走出场景，第一到三个显著目标（男）即将分散。



260  
帧



原图

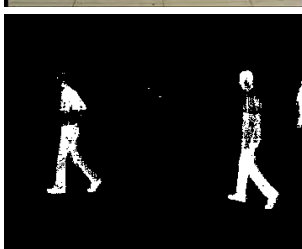
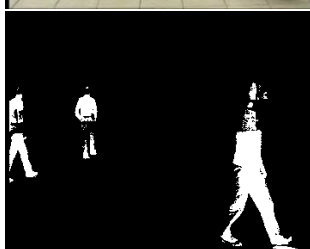


本文算法

350  
帧

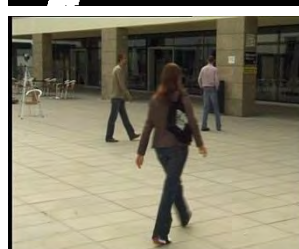


原图

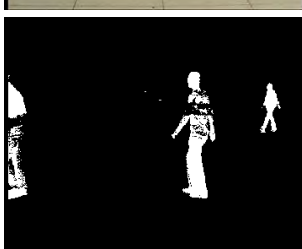


本文算法

376  
帧

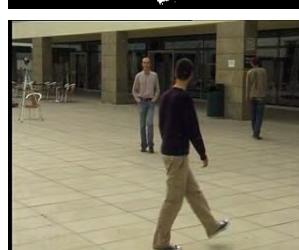


原图



本文算法

490  
帧



原图

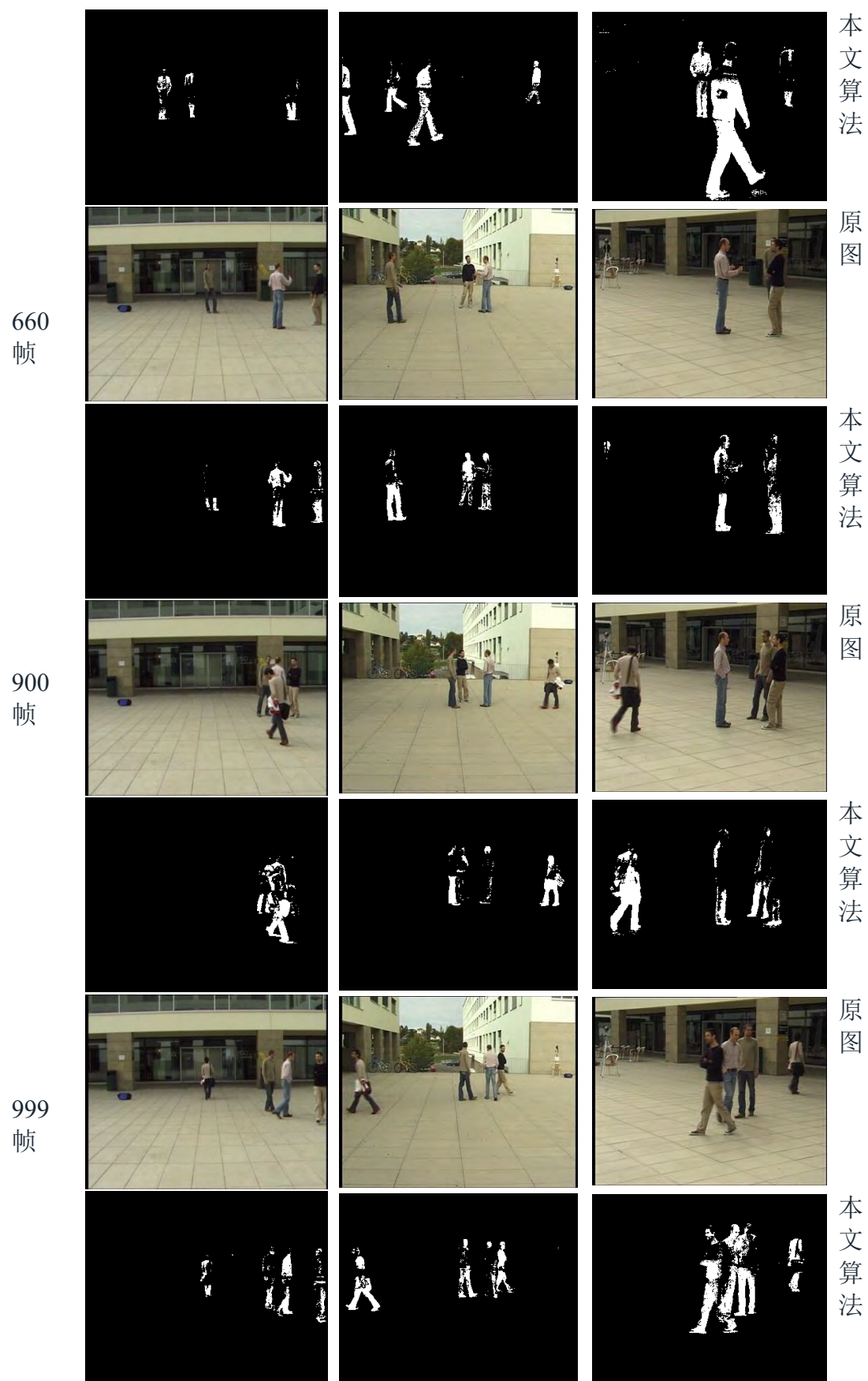


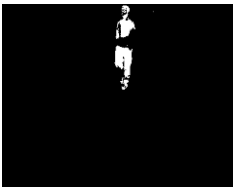






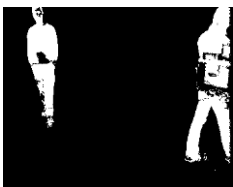














图 6.1 Campus 场景下多角度视频的前景目标提取



6.3.2 Laboratory 场景下多角度视频的前景目标提取

Laboratory 场景为同一时间从四个不同角度拍摄同一地点的视频，对于四个摄像机镜头我们分别命名为 Camera01、Camera02、Camera03、Camera04。对于 Laboratory 场景我们选取以下关键帧：180 帧，第一个显著目标（男）出现在场景中；480 帧，第二个显著目标（男）出现在场景中；736 帧，第三个显著目标（男）出现在场景中； 999 帧，视频最后一帧，三个显著目标（男）聚集。

帧数	Camera01	Camera02	Camera03	Camera04	
180 帧					原图
					本文算法
480 帧					原图
					本文算法
736 帧					原图
					本文算法
999 帧					原图

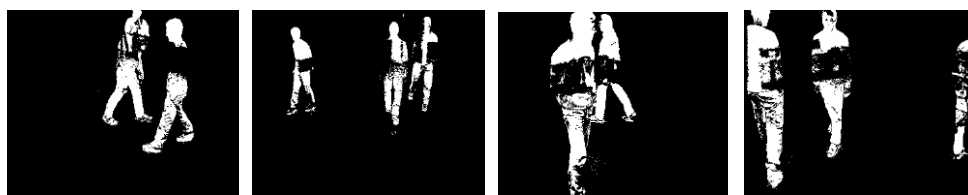


图 6.2 Laboratory 场景下多角度视频的前景目标提取

## 7. 问题六：异常事件的发现

### 7.1 问题分析

对于问题六，利用获得的前景信息，检测视频中有人群聚集、人群逃散、群体规律性变化、物体爆炸、建筑物倒塌等异常事件的发生。在解决该类问题中，

各种有效的方法被提出。其中一种流行的方法是基于轨迹建模的，它跟踪每个场景中的每个目标并且对目标的运动轨迹进行建模，但这种方法对于高密度场景非常困难。另外一种流行的方法是不基于跟踪的运动表示方法，最典型的是光流法或者其他形式的基于时空梯度信息的方法。

本文采用灰度矩阵与光流法相结合的方法，检测人群异常事件，如果完全利用运动信息，容易在人群低密度的情况下产生误判，所以结合人群密度分析，能够更好进行检测。

### 7.2 基于灰度矩阵和光流法的人群异常检测

#### 7.2.1 灰度共生矩阵

因为纹理是灰度分布在空间位置上反复出现而形成的，所以在图像空间中距离为某个值的 2 个像素之间会存在一定的灰度关系，而灰度共生矩阵是对图像上保持某距离的 2 个像素分别具有某灰度的情况进行统计得到的。在一张图像中随机地选择一个像素点  $(x, y)$  和一个水平距离  $a$ ，垂直距离  $b$  的点  $(x+a, y+b)$ ，形成一个点对。对图像中所有点都两两组合成一个点对，然后取得所有点对的灰度值。灰度共生矩阵实质上代表的是 2 个距离为  $(a, b)$  的像素点的联合直方图， $a$  和  $b$  取不同的值就可以求得图像沿着某一个方向，相距某一个距离的灰度共生矩阵。一般情况下，方向取  $0^\circ$ 、 $45^\circ$ 、 $90^\circ$  和  $135^\circ$  总共 4 个方向。如下图所示，假设左图有 8 个灰度级，设距离为 1，方向取水平方向，从图中统计出像素灰度值 1 变化到灰度值 1 的像素对总计 1 次，而从灰度值 1 变化到灰度值 2 的像素对总计为 2 次。

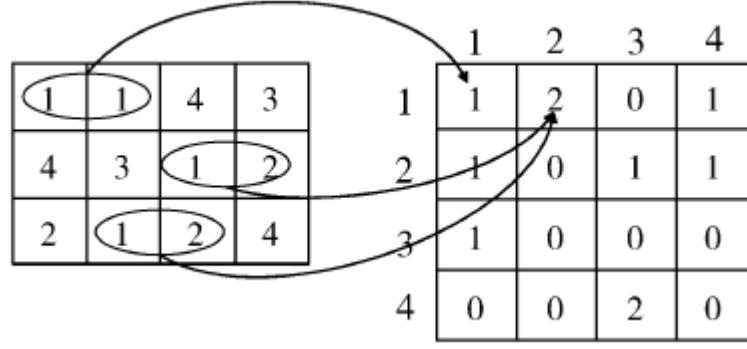


图 7.1 灰度共生矩阵

### 7.2.2 基于光流法的人群运动估计

假设一系列连续的图像。 $F(x,y,t)$ 是时刻  $t$  上坐标点为  $(x,y)$  的灰度值。时域中的动态图像用关于位置 and 时间的函数来表示，那么就是一个泰勒公式：

$$f(x + dx, y + dy, t + dt) = f(x, y, t) + f_x dx + f_y dy + f_t dt + O(\delta^2) \quad (7.1)$$

其中  $f_x$ 、 $f_y$ 、 $f_t$  表示  $f$  的偏导数。假设  $(x,y)$  的直接领域在时间间隔  $\Delta t$  内被平移了某个微小的距离  $(dx,dy)$ ，即有  $dx,dy,dt$  使得：

$$f(x + dx, y + dy, t + dt) = f(x, y, t) \quad (7.2)$$

如果  $dx$ 、 $dy$ 、 $dt$  非常小，那么式 ( ) 中的高阶项可以忽略不计，则有：

$$-f_t = f_x \frac{dx}{dt} + f_y \frac{dy}{dt} \quad (7.3)$$

上述公式就是光流方程。令  $\mu = \frac{dx}{dt}$ ， $v = \frac{dy}{dt}$ ，则光流方程可表示为：

$$f_x \mu + f_y v = -f_t \quad (7.4)$$

其中， $(\mu, v)$  是速度矢量。

光流法由于需要计算光流场，运算复杂度较高，Gunnar 提出基于多项式展开的运动场估计，起初该算法用于解决摄像机拍摄抖动的问题。本文采用此算法来估计相邻两帧图像之间的光流场。假设  $S = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  是前一帧图像的点集，通过光流场计算，可以得到当前帧上与点集相对应的点集  $Q = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ ，则对于某个点来说，它的一帧的运动位移为：

$$|V_i| = \sqrt{(x_i - X_i)^2 + (y_i - Y_i)^2} \quad (7.5)$$

其运动方向为：

$$O(V_i) = \arctan \left( \frac{|y_i - Y_i|}{|x_i - X_i|} \right) \quad (7.6)$$

通过对这些点集的运动矢量进行统计，统计得到的主要运动趋势应该就是人群的主要运动趋势。本文在统计分析时利用基于方向直方图的方法，方向直方图是按每  $45^\circ$  建立一个直方图条 (bin)，那么  $360^\circ$  方向就总共有 8 个直方图条。直方图定义如下：

$$h(i) = \{k_i, 0 < i \leq n\} \quad (7.7)$$

其中， $n$  为直方图刻度数， $k_i$  代表每一帧第  $i$  刻度所覆盖的角度中运动矢量的数目。根据方向直方图，将每一个单位的数目除以运动矢量的总数，就可以得到每一帧方向的各个方向的概率分布，其定义为：

$$p(i) = \frac{h(i)}{m}, 0 < i \leq n, i \in N \quad (7.8)$$

其中， $m$  为一帧中运动矢量的总数。有了运动矢量方向的概率分布，可以采用方向熵来表征运动矢量方向的混乱程度，从而估计出当前人群运动的混乱程度。本文定义方向熵为：

$$H(x) = -\sum_{i=1}^n p(x_i) \log p(x_i) \quad (7.9)$$

其中， $p$  为事件发生的概率分布，如果熵值较小，则认为场景中人群运动方向趋于一致，那么此时发生聚集的可能性很小；如果熵值较大，表面人群运动混乱，那么发生聚集的可能性很大。

### 7.2.3 算法设计及流程

本文将人群密度估计以及人群运动估计相结合来检测人群聚集事件，如果仅采用人群密度，那么诸如行军、游行、人行道路口行人等有秩序的运动也会被误判为聚集行为，如果仅靠运动估计，那么在人群稀疏场景下，运动矢量仍有可能比较混乱，此时判定为人群异常事件则与实际情况不符。本文将 2 种方法相结合，取长补短，只有人群密度达到一定程度且人群运动混乱时才会判定为发生了聚集事件。下图为人群聚集事件的算法流程图。

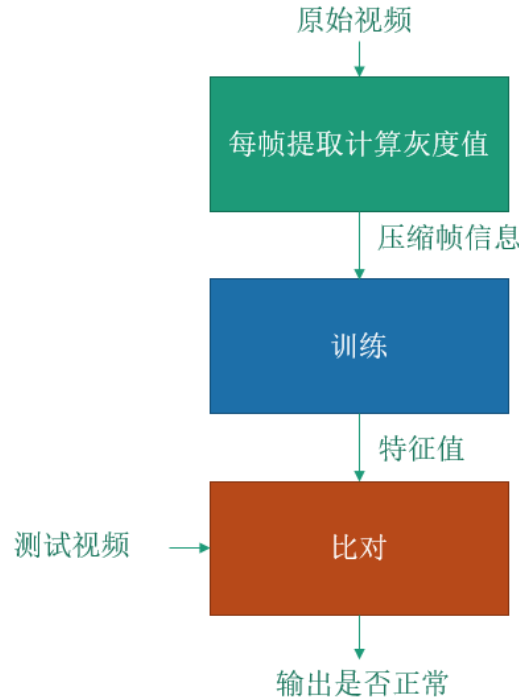


图 7.2 人群聚集事件的算法流程图

可以事先知道监控场景的大小，因此可以通过先验知识来确定下列公式中的阈值取值，定义阈值判定公式如下所示。

$$\text{isGather} = \begin{cases} 1, & y = \text{dense and } n > \theta \\ 0, & y = \text{sparse or } n \leq \theta \end{cases} \quad (7.10)$$



其中,  $y$  是人群密度估计的结果,  $\theta$  是当前帧运动方向熵的阈值, 当人群密度估计结果为密集, 且方向熵大于一定阈值的时候则判定为聚集, 当密度结果为稀疏, 或者方向熵小于一定阈值的时候则不认为发生了聚集事件。

## 7.3 实验结果及分析

### 7.3.1 人群异常时的异动事件检测

对人群的异动事件检测, 我们主要对以下三个场景, 即: 操场、走廊以及广场进行检测。通过上述算法, 实验结果如下:

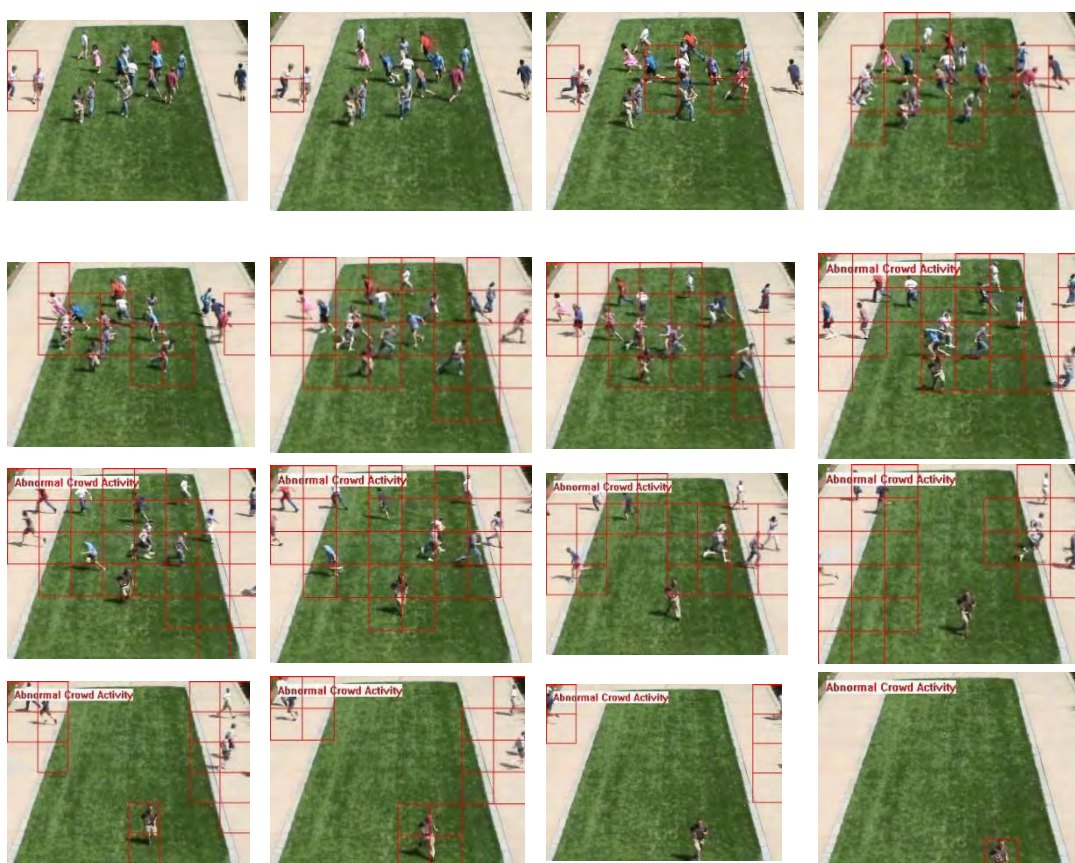
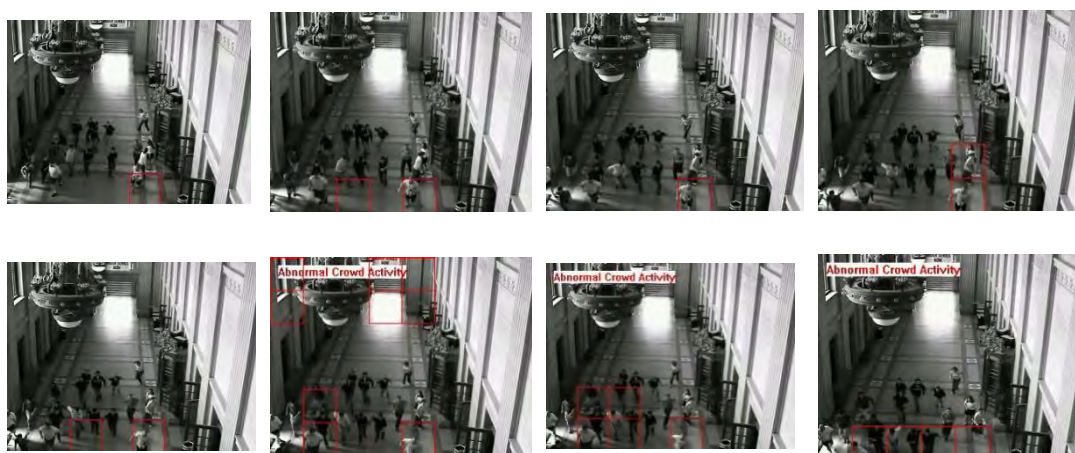


图 7.3 操场场景下的人群异动事件检测



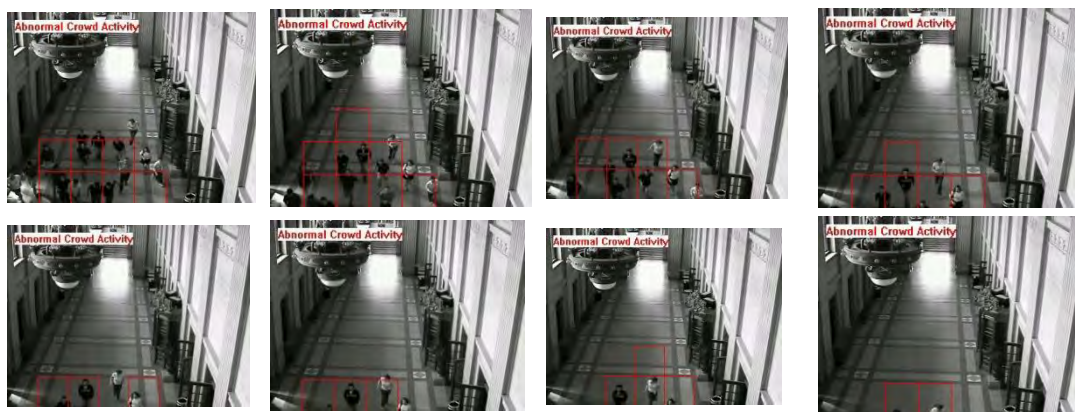


图 7.4 走廊场景下的人群异动事件检测

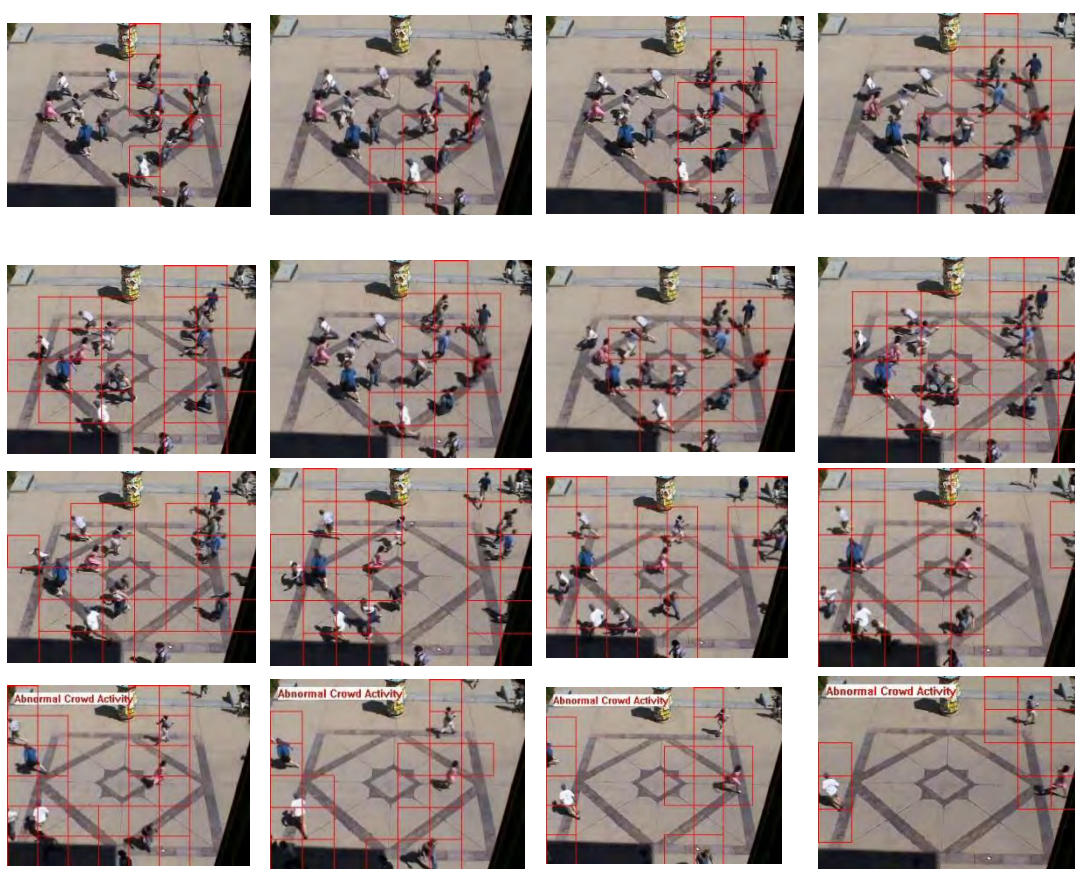


图 7.5 广场场景下的人群异动事件检测

### 7.3.2 发生物体爆炸时的异动事件检测

对物体爆炸异动事件的检测，通过上述算法，实验结果如下：





图 7.6 发生物体爆炸时的异动事件检测

### 7.3.3 发生建筑物倒塌时的异动事件检测

对建筑物倒塌异动事件的检测，我们主要对以下场景进行了检测。通过上述算法，实验结果如下：

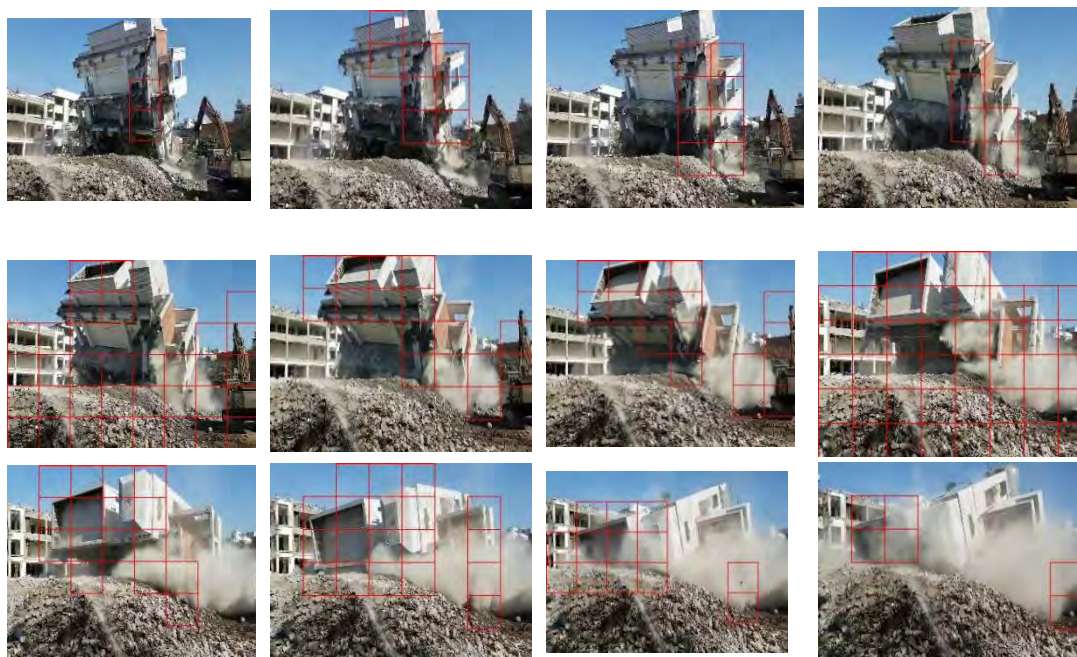




图 7.7 发生建筑物倒塌时的异动事件检测

## 8. 问题总结与评价

### 8.1 问题总结

本文通过不同的模型依次解决了题目中所提出的静态背景下前景目标提取、动态背景下前景目标提取、晃动镜头下前景目标提取、多镜头下前景目标提取以及人群聚散、爆炸、建筑物倒塌等异常事件发现等问题，总结如下：

对于问题一，本文采用了混合高斯模型对含有静态背景的监控视频进行前景提取。本文从背景模型初始化、背景减除、背景更新三个方面描述了模型的建立过程和参数选取过程，并运用该模型完成了多个场景下的前景提取(图 2.1-2.4)，选取了多个关键帧和其它算法进行对比。在此基础上，本文提出了一系列评价指标，通过比较不同算法的召回率、准确率和 F 值等指标（表 2.1-2.4），发现混合高斯模型有较好的效果。

对于问题二，本文采用了 ViBe 模型对含有动态背景的监控视频进行前景提取，并详细阐述了模型建立过程。本文运用 ViBe 模型对含有水面波动的场景进行了前景提取，并和其它算法在关键帧（图 3.3）和各项指标上进行了详细对比（表 3.1）。

对于问题三，本文首先运用像移补偿技术对视频进行了稳像处理，然后再进行前景提取。本文主要从运动模型与运动估计、背景匹配、图像补偿多个方面介绍了像移补偿模型的建立过程，最后对比了采用稳像技术和不采用稳像技术在前景提取上的差异，并发现采用稳像技术的前景提取技术在各个指标上有显著提升（表 4.1-4.4）。

对于问题四，本文根据各个场景进行了不同分类，然后依据分类采用了前文已验证具有较好效果的模型，即对不含晃动的静态背景采用混合高斯模型、对不含晃动的动态背景采用 ViBe 模型。在获得前景提取照片后，分析对应帧的变化情况和密度，找到包含显著目标前景的关键帧。本文给出了所有场景的关键帧信息（表 5.1），并选取了部分帧进行验证。

对于问题五，本文分析了多摄像头下目标匹配和跟组等关键问题，并采用了坐标转换的方法来校对各个视频的前景提取。本文选用了公开数据中 Campus 和 Laboratory 两个场景，分析了各个视频的拍摄角度，然后选取了多个关键帧展示了算法的效果（图 6.1-6.2）。

对于问题六，本文分析了人群或其他异常检测的难点，并采用了灰度共生矩阵和光流法相结合的模型来发现异常事件。首先本文利用正常的视频作为输入，在上面进行模型训练，然后输入需要被检测的视频，如果包含异常事件，根据光流分析会给出异常事件出现位置。本文选取了人群突然扩散，爆炸，建筑物倒塌等多个场景进行测试，并给出了过个关键帧的详细分析（图 7.3-7.7）。



## 8.2 模型评价

本文主要使用了混合高斯模型、ViBe 模型、像移补偿技术、坐标转换法、灰度共生矩阵结合光流法五种模型，对这五种模型评价如下：

第一，混合高斯模型适用于解决不带晃动-静态背景的前景目标提取问题。相比于存在较多噪声点的帧间差分法和完整性较弱的单高斯模型，混合高斯模型出现的噪声点较少，前景目标提取较为完整，能够看出前景目标轮廓。从评价指标计算出结果来看，混合高斯模型在准确度（Precision）、特指度（Specificity）、F 值（F-Measure）等方面均显著优于其他模型。但从提取效果看，该模型仍存在部分误检的情况。

第二，ViBe 模型适用于解决不带晃动-动态背景的前景目标提取问题。在解决此类问题时，帧间差分法噪声点较多的问题仍然存在，而混合高斯模型对于动态背景的背景剔除效果一般。相比之下，ViBe 模型提取的前景目标背景中出现的噪声点很少，前景目标提取完整、轮廓清晰、空洞很少，背景剔除效果也较好，但仍存在部分误检的情况。从评价指标计算出结果来看，ViBe 模型在召回率（Recall）、准确度（Precision）、特指度（Specificity）、F 值（F-Measure）等方面也均显著优于其他模型。

第三，像移补偿技术适用于解决有晃动视频的前景目标提取问题。对于有晃动的视频进行前景提取时，混合高斯模型存在着噪声点较多、背景剔除不彻底等问题，ViBe 模型虽然噪声点相对较少，但在背景提出和完整性上不及经电子稳像算法处理后的效果。从评价指标计算出结果来看，电子稳像算法在准确度（Precision）、特指度（Specificity）、错误分类率（PWC）、F 值（F-Measure）等方面均显著优于其他模型。

第四，坐标转换法主要用于解决多摄像头情况下对前景目标的检测和追踪问题。由于不同摄像头对目标的坐标系不同，上述单一模型难以解决这一问题，坐标转换法对此有很好的弥补作用。

第五，灰度共生矩阵结合光流法主要用于解决异常事件检测的问题。经实验验证，对于此类问题，该算法有着较好的检测跟踪效果。

## 8.3 模型推广

本文所采用的模型能有效解决静态背景及动态背景下前景目标提取、晃动镜头下前景目标提取、多镜头下前景目标提取以及人群聚散、爆炸、建筑物倒塌等异常事件发现等问题，对于提升安保质量有很强的现实意义。本文算法经实验验证具有较好的效果，关键代码也将在附录里给出，具有很强的推广价值，

## 9 参考文献

- [1] 娄娜，运动目标的分割与跟踪，华中科技大学硕士论文，2006 年 10 月。
- [2] Hany Farid and Jeffrey B. Woodward, Video Stabilization and Enhancement, Computer Science, TR2007-605.
- [3] 徐俊红，李金，丛望，张建兵，基于静态背景下的视频多目标运动物体的提

- 取, 应用科技, 第 32 卷第 3 期: 31-33, 2005 年 3 月。
- [4] 桑海峰, 李伟, 陈仲, 自适应权值的静态背景提取与更新算法, 电子测量与仪器学报, 第 29 卷第 3 期: 358-367, 2015 年 3 月。
  - [5] 高韬, 刘正光, 张军, 岳士弘, 基于特征点的多运动目标跟踪, 电子与信息学报, 第 32 卷第 5 期: 1111-1115, 2010 年 5 月。
  - [6] 周文琳, 视频预处理系统中背景减除技术的研究与实现, 西安电子科技大学硕士论文, 2014 年 3 月。
  - [7] 刘潇然, 基于形状信息的运动目标提取方法研究, 浙江大学硕士论文, 2017 年 3 月。
  - [8] 刘舒, 基于自适应背景减除方法的运动目标检测, 哈尔滨工业大学, 2013 年 12 月。
  - [9] 吕杉, 基于 OpenCV 的视频运动检测中背景减除算法的研究与改进, 吉林大学硕士论文, 2011 年 4 月。
  - [10] 宁文鑫, 监控视频中基于背景减除的遗弃行李检测方法, 清华大学硕士论文, 2011 年 4 月。
  - [11] 林沁, 章历, 基于灰度共生矩阵和光流法的人群异动事件检测, 计算机与现代化, 总第 223 期, 2014 年第 3 期。
  - [12] 梁华, 多摄像机视频监控中运动目标检测与跟踪, 国防科学技术大学硕士论文, 2009 年 4 月。
  - [13] 郭达洁, 监控视频中的前景提取和目标检测跟踪算法研究, 浙江大学硕士论文, 2016 年 1 月 20 日。
  - [14] 戴金平, 智能视频监控中的运动目标检测与跟踪算法研究, 杭州电子科技大学硕士论文, 2009 年 12 月。
  - [15] 尚进, 监控视频中行人异常行为检测系统的设计与实现, 东南大学硕士论文, 2015 年 4 月。

## 10 附录

部分 Python 代码:

```
# -*- coding: utf-8 -*-
#分割视频的代码
import cv2
def spl():
    cap = cv2.VideoCapture('building2.mp4')
    start, end = 3300, 3600
    print "Here"
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter('5_test2.avi', fourcc, 30.0, (320, 240))
    cap.set(cv2.CAP_PROP_POS_FRAMES, start)
    while end-start>0:
        print "loop"
        ret, frame = cap.read()
        frame = cv2.resize(frame, (320, 240))
```

```

        out.write(frame)
        cv2.imshow('frame', frame)
        cv2.waitKey(1)
        start+=1
    cap.release()
    out.release()
    cv2.destroyAllWindows()

spl()

```

```

import cv2
import os
from os import listdir
from os.path import isfile, join

#calculate the Recall and Precision etc.
def calculate(path_gt, path_fg):
    files_gt = [f for f in listdir(path_gt) if isfile(join(path_gt, f))]
    files_fg = [f for f in listdir(path_fg) if isfile(join(path_fg, f))]

    TP = .0 # True positive pixels
    FP = .0 # False positive pixels
    TN = .0 # True negative pixels
    FN = .0 # False negative pixels
    Recall = .0 # TP / (TP + FN)
    Precision = .0 # TP / (TP + FP)
    Specificity = .0 # TN / (TN + FP)
    FPR = .0 # FP / (FP + TN)
    FNR = .0 # FN / (TP + FN)
    PWC = .0 # 100 * (FN + FP) / (TP + FN + FP + TN)
    Fscore = .0 # 2*(Precision*Recall)/(Precision+Recall)

    # TP - # of foreground pixels classified as foreground pixels.
    # FP - # of background pixels classified as foreground pixels.
    # TN - # of background pixels classified as background pixels.
    # FN - # of foreground pixels classified as background pixels.

    k = 1
    for file_gt, file_fg in zip(files_gt, files_fg):
        # print(k, file_gt, file_fg)
        if k == 1:
            k = k + 1
            continue
        img_gt = cv2.imread(path_gt + file_gt, cv2.IMREAD_GRAYSCALE)

```

```

img_fg = cv2.imread(path_fg + file_fg, cv2.IMREAD_GRAYSCALE)

rows, cols = img_gt.shape
for i in xrange(rows):
    for j in xrange(cols):
        pixel_gt = img_gt[i, j]
        pixel_fg = img_fg[i, j]
        if (pixel_gt == 255 and pixel_fg == 255):
            TP = TP + 1
        if (pixel_gt == 0 and pixel_fg == 255):
            FP = FP + 1
        if (pixel_gt == 0 and pixel_fg == 0):
            TN = TN + 1
        if (pixel_gt == 255 and pixel_fg == 0):
            FN = FN + 1

    k = k + 1
    # if k == 100:
    #     break
    # break

Recall = TP / (TP + FN)
Precision = TP / (TP + FP)
Specificity = TN / (TN + TP)
FPR = FP / (FP + TN)
FNR = FN / (TP + FN)
PWC = 100 * (FN + FP) / (TP + FN + FP + TN)
Fscore = 2 * Precision * Recall / (Precision + Recall)

# print path_gt,path_fg
print TP, FP, TN, FN, Recall, Precision, Specificity, FPR, FNR, PWC, Fscore
# print ''

# path_gt = 'l_result/airport/mask_in/'
# path_fg = 'l_result/airport/output_1/fg/'
# words = ['airport', 'hall', 'office', 'pedestrian', 'smoke']
# words = ['cars6', 'cars7', 'people1', 'people2']
words = ['corridor']
methods = ['fd', 'output_1', 'output_2']
for word in words:
    path_gt = 'l_result/' + word + '/mask_in/'
    for method in methods:

```

```
path_fg = 'l_result/' + word + '/' + method + '/fg/'
calculate(path_gt, path_fg)
```

```
import numpy as np
import motionInfluenceGenerator as mig
import createMegaBlocks as cmb

# traing the model

def train_from_video(vid):

    print "Training From ", vid
    frames, rows, cols = mig.getMotionInfluenceMap(vid)
    blockInfo = cmb.createMegaBlocks(frames, rows, cols)

    path, file = getTrainPath(vid)
    filename = path + "blockInfo_k5_" + file + ".npy"
    np.save(filename, blockInfo)
    if 'train' in file:
        codewords = cmb.kmeans(blockInfo)
        filename = path + "codewords_k5_" + file + ".npy"
        np.save(filename, codewords)
    # print codewords
    return

def getTrainPath(name):
    path = name[0:name.rindex('/') + 1]
    file = name[name.rindex('/') + 1:len(name)]
    file = file[0:file.rindex('.')]
    return path, file

if __name__ == '__main__':

    trainingSet = [
        "../Dataset/videos/scene4/4_train1.avi",
        "../Dataset/videos/scene4/4_test1.avi",
        "../Dataset/videos/scene5/5_train1.avi",
        "../Dataset/videos/scene5/5_train2.avi",
        "../Dataset/videos/scene5/5_test1.avi",
        "../Dataset/videos/scene5/5_test2.avi",
    ]

    for video in trainingSet:
```

```
train_from_video(video)  
print "Done"
```