

目录

一 算法方案介绍	2
1 技术背景.....	2
2 方案原理.....	3
3 实施细节.....	4
3.1 低分辨率运行	4
3.2 ASMS 引导.....	4
3.3 加权背景重建	4
3.4 背景差分	5
3.5 基于灰度直方图的自适应阈值分割算法.....	5
3.6 质心回归算法	7
二 测试数据及验证	8
1 评估方法.....	8
2 测试步骤.....	8
3 测试数据分析与对比	8
三 总结.....	9
1 速度	9
2 鲁棒性	9
3 算法优点.....	10
4 算法缺点.....	11

一 算法方案介绍

1 技术背景

视觉跟踪和目标检测是计算机视觉领域内较早开始的研究方向。经过几十年的积累，这两个方向已经取得了显著的发展。然而，很多方法只是在相对较好地程度上解决了一些关键问题。并且仍旧有不少一般性的关键问题未得到有效的解决。国内外很多研究机构都在致力于研究和发展这两个方向。近些年这两个方向持续发展，涌现了很多比较优秀的方法。

经典的目标视觉跟踪方法，大家比较公认的是两大类：生成模型方法和判别模型方法。**生成类方法**：在当前帧对目标区域，下一帧寻找与模型最相似的区域就是预测位置，比较著名的有卡尔曼滤波，粒子滤波，mean-shift 等；**判别类方法**：CV 中的经典套路图像特征结合机器学习的方法，当前帧以目标区域为正样本，背景区域为负样本，机器学习方法训练分类器，下一帧用训练好的分类器找最优区域。

随着目标跟踪技术的发展，目标跟踪领域出现了相关滤波类方法和深度学习方法，这两种方法大有超越经典方法之势。**相关滤波类方法**的算法有 DAT、ASMS、Staple、C-cot、DSST、SAMF 等。其中 DAT 和 ASMS 都采用纯颜色直方图特征；Staple 采用了 HOG 和颜色直方图结合的特征；DSST 在 KCF 的基础上即 HOG，增加了尺度自适应的滤波方式；C-cot 采用了空域正则化和 SRDCFdecon 的自适应样本权重方法；SAMF 基于 KCF，特征是 HOG+CN。**深度学习类方法**有 MDNet、TCNN，速度方面比较突出的如 80FPS 的 SiamFC 和 100FPS 的 GOTURN。GOTURN，效果比较差，但优势是跑的很快 100FPS。需要注意的是深度学习类方法都是在 GPU 上运行。

本次大赛要求算法能够在静止的室内场景中跟踪摄像头中的指定单个人体目标，并且对算法的速度及鲁棒性都有一定的要求。由于比赛使用的 ARM 单板机上没有 GPU，因此深度学习类方法在 ARM 上的运行速度相对较慢。因此我们小组决定采用非深度学习的方法。

我们在实际场地上分别测试了基于 HOG 特征和颜色直方图特征的算法，发现基于颜色直方图的算法表现会相对较好一些。原因是比赛环境比较复杂，颜色梯度变化较大，HOG 特征受到较大环境干扰，容易出现跟踪失败的情况；基于颜色直方图特征可以较好

地完成跟踪任务，但也存在一定的问题，算法对不同颜色衣服的人跟踪效果差距较大，即鲁棒性不够好。

针对以上的问题及比赛要求，我们小组提出了**基于 ASMS 引导的背景差质心回归算法**，用基于颜色直方图特征的 ASMS 算法进行引导，完成环境的重建后利用背景差转 binar 图并采用相关去噪方法去除环境中的噪声点，在计算人体质心之后利用回归算法得到相应的跟踪边框。该算法的运行速度不逊于其他相关滤波类方法，且有较强的鲁棒性。

2 方案原理

针对比赛所提出的对算法的性能及鲁棒性的要求，我们设计了一种基于 ASMS 引导的背景差质心回归算法。初期该算法运用 ASMS 算法进行引导，从而完成对跟踪背景的加权构建。在完成背景模型重建后，使用背景差分的方法得到伪前景，并使用了我们自行设计的基于灰度直方图的自适应阈值分割算法，进一步完成对前景的提取。最后，使用质心回归算法，计算前景的图像质心，并在质心周围对检测框分别在 X 与 Y 方向上进行迭代回归，最终得到准确的目标检测框位置。

各个部分的设计思路如下：

1. ASMS 算法的鲁棒性较强，运行速度快，能够为背景重建提供稳定的引导。

2. 使用了加权背景重建主要考虑了一下两点：考虑到暗光环境下的背景噪声较多，在重建背景图像的过程中，使用了加权重建的方法来滤除高斯白噪声，使背景图像更加稳定；加权重建具有更好的抗干扰性，抑制了当跟踪目标框选过小时，目标框外图像对背景的影响。

3. 背景差分的方法是一种全局快速搜索方法，有很好的鲁棒性。相比相邻帧差分法，能够提供更为稳定的前景信息。

4. 相比于人工设定分割阈值，基于灰度直方图的自适应阈值分割算法提升了在不同光照条件下对前景信息的提取能力，使提取到的前景信息更为完整。

5. 质心回归算法用于跟踪目标的检测框的生成。通过质心计算的方法得到前景的检测框中心，抑制了来自环境噪声的少量伪前景图像的干扰，提升算法鲁棒性。

最后，以检测框中心分别对检测框的长宽以一定的标准进行迭代，得到最终检测结果。

3 实施细节

3.1 低分辨率运行

在经过算法测试与权衡后，我们采用 160*120 分辨率的图像（即为原图像四分之一的大小）进行目标跟踪，一方面这是由于我们的算法有较高的鲁棒性，另一方面这一分辨率能很好的平衡速度与准确度。

此外，由于采用了双线性插值的方法得到低分辨率图像，能够同时能够实现一定的去噪效果。

3.2 ASMS 引导

算法的第一步是使用 ASMS 算法进行地图构建引导。通过在第一帧人为标注的检测框确定引导区域，如图 1(a)的红色框所示。为确保重建的稳定性，对检测框进行适当的放缩，避免跟踪目标落入背景区域。图 1(b)与图 1(c)则为引导区域与背景区域。

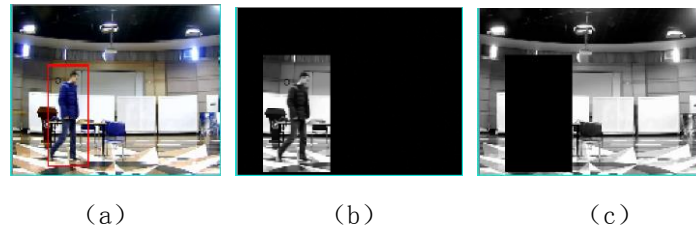


图 1 ASMS 引导算法示意

3.3 加权背景重建

该部分主要由三个变量构成： W ， Z ， B 。 W ， Z ， B 分别表示图像像素的权重图，零权重图与背景图，三张图分别与图片像素点一一对应。 $W(x,y)$ 代表了像素点 (x,y) 处的背景是由 $W(x,y)$ 个背景像素点加权计算得到的； $Z(x,y)$ 为 0 代表了像素点 (x,y) 已经有至少一个背景像素点， $Z(x,y)$ 为 1 则代表这个点还未被背景像素填充，如图 1(c)中的黑色矩形区域。

背景更新的计算公式如下：

$$Z_n = Z_{new} \& Z_{n-1}$$

$$B_n = \frac{B_{n-1}.*W_{n-1} + Img.*W_{new}}{W_{n-1} + Z_{n-1}}$$

$$W_n = W_{n-1} + W_{new}$$

Img : 摄像头获取的第 n 帧图像;

Z_{new} : 第 n 帧图像中的零权重图;

W_{new} : 第 n 帧图像中的权重图;

Z_n 、 Z_{n-1} : 由 n 、 $n-1$ 帧零权重图进行与操作得到的零权重图;

B_n 、 B_{n-1} : 由 n 、 $n-1$ 帧图像得到的背景图;

W_n 、 W_{n-1} : 由 n 、 $n-1$ 帧权重图累加得到的权重图;

$.*$: 两矩阵之间对应位元素相乘。

3.4 背景差分

在得到完整的背景后，对当前帧与背景图像做差分，得到前景图像，如图 2(c)所示。

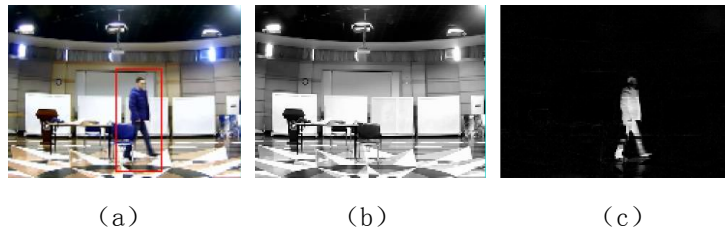


图 2 背景差分

3.5 基于灰度直方图的自适应阈值分割算法

考虑到跟踪环境的光照变化的情况下，背景差分得到的图像的亮度值有较大的差异（如图 3 所示），我们提出了基于灰度直方图的自适应阈值分割算法。

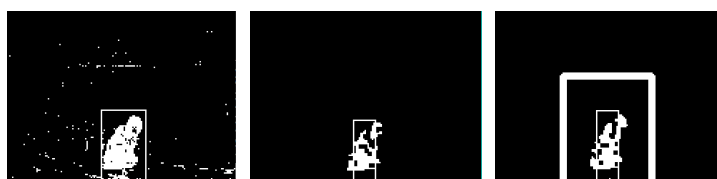


(a) 正常光照 (b) 暗光环境

图 3 背景差分

该算法的思路如下：

- 统计图像的 256bin 灰度直方图；
- 自适应阈值（大津法）；
- 对前景图像进行二值化分割；
- 使用(3,3)的腐蚀核对图像进行腐蚀去除背景噪声；
- 远波滤除，对以上一帧检测框为中心，对大于目标框两倍以外的噪声进行去除（该部分作为可选滤波，针对不同条件自行判断是否启用）。



(a) 自适应阈值化 (b) 腐蚀噪声 (c) 远波滤除

图 4 背景差分



(a) 正常光照 (b) 暗光环境

图 5 阈值化分割

根据测试结果分析得到，该算法具有很好的鲁棒性，能够在不同的光照条件下均较为准确地分割得到前景图像。

3.6 质心回归算法

在得到二值化前景图像后，对图像进行质心计算：

$$\text{Center} = \frac{\sum P(x,y)}{\text{num_white}}, \text{ if } I(x,y) == 255$$

$P(x,y)$: 像素点坐标；

$I(x,y)$: (x,y)处的亮度值；

Num_white: 前景像素点数量。

由于跟踪目标为前景图像的主要信息，经过质心计算能够有效的减少来自其他少量的噪声点的干扰，得到较为准确的检测框的中心位置。

然后，算法以质心点为中心，计算前景分割图像在 X 与 Y 方向的投影图：

$$X_{proj}(x) = \text{num of } P(x,y_i), I(x,y_i) == 255, i \propto 0 \sim \text{img.width}$$

$$Y_{proj}(y) = \text{num of } P(x_i,y), I(x_i,y) == 255, i \propto 0 \sim \text{img.height}$$

最后，在 X 方向与 Y 方向上按照框选比例进行迭代回归，框选比例如下：

$$\text{Ratio}_{\text{iter}} = \frac{\text{Inner_white}}{\text{num_white}}$$

Inner_white: 被检测框框选的像素点数量；

Num_white: 前景像素点数量。

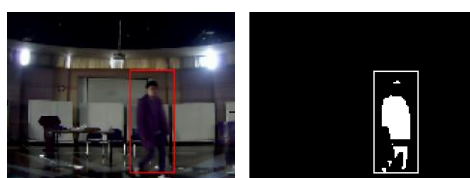


图 6 目标检测结果

二 测试数据及验证

1 评估方法

针对弯腰、关灯、遮挡下半身等具体测试场景，使用程序运行相应的视频文件，查看跟踪框的成功率及准确度，记录平均速度。

2 测试步骤

(1) 修改“main_demo.cpp”中的宏定义“#define USE_VIDEO_FILE”，重新编译程序；

(2) 选定具体测试场景，使用程序打开对应的视频文件；

(3) 观察跟踪框的成功率及准确度，记录平均速度。

3 测试数据分析与对比

	成功率	准确度	每帧平均用时 (ms)
普通	高	高	5
弯腰	高	高	5
关灯	高	较高	5
背景人群	高	较高	5
遮挡下半身	高	一般	5
白纸	高	一般	5

从上述测试可见，算法每帧平均处理时间大约为 5ms，满足实时性要求。

该算法具有一定的鲁棒性，对于普通、弯腰、关灯、背景人群的场景拥有高成功率以及较高准确度;对于遮挡下半身的场景，拥有高成功率以及可接受的准确度;白纸场景需等待背景重建完毕后，即可正确应对白纸场景。

三 总结

该算法与深度学习算法（Goturn）、梯度滤波类算法（KCF、DSST）、颜色滤波类算法（ASMS）的对比情况如下：

1 速度

选取 1 个普通场景的视频文件，记录不同算法运行的平均用时：

算法	该算法	Goturn	KCF	DSST	ASMS
每帧平均时间（ms）	5	350	16	45	43

2 鲁棒性

选取不同场景的视频文件，记录不同算法运行情况：

	该算法	Goturn	KCF	DSST	ASMS
普通	成功率高 准确率高	成功率高 准确率一般 （只框半身）	成功率高 准确率高	成功率高 准确率高	成功率高 准确率较高
弯腰	成功率高 准确率高	成功率高 准确率一般 （只框半身）	成功率高 准确率较高	成功率高 准确率较高	成功率高 准确率一般
关灯	成功率高 准确率较高	成功率高 准确率一般 （只框半身）	成功率较高 准确率高	成功率较高 准确率高	成功率高 准确率较高
背景人群	成功率高 准确率较高	成功率低 丢失目标	成功率低 丢失目标	成功率低 丢失目标	成功率高 准确率较高
遮挡下半身	成功率高 准确率一般	成功率低 丢失目标	成功率低 丢失目标	成功率低 丢失目标	成功率高 准确率一般

白纸	成功率高 准确率一般	成功率低 丢失目标	成功率低 丢失目标	成功率低 丢失目标	成功率低 丢失目标
----	---------------	--------------	--------------	--------------	--------------

3 算法优点

(1) 与传统背景减除法相比，采用了质心回归的方式，减少了背景噪点对跟踪的影响。如下图所示场景，背景出现噪点，但由于质心回归方法的应用，该噪点对跟踪的影响被移除。



图 8 该算法在出现噪点时的鲁棒性

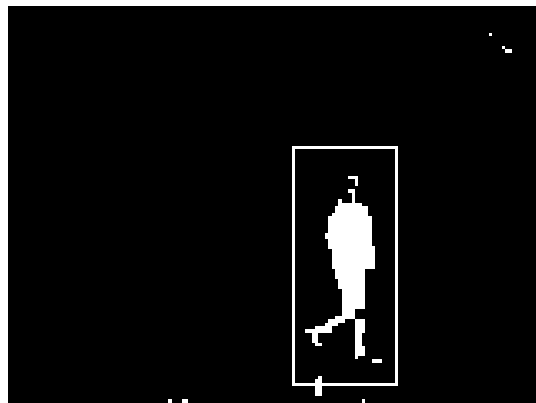


图 9 噪点的分布情况

(2) 与梯度滤波类方法相比，对复杂梯度背景的抗干扰能力更强，不容易在复杂梯度处丢失目标。如下图所示的人群背景的场景中，梯度滤波类方法在背景区域梯度与目标区域梯度相近时出现丢失目标。



图 10 该算法在复杂梯度下的鲁棒性



图 11 梯度滤波类算法在复杂梯度下的鲁棒性

(3) 与颜色滤波类方法相比，对目标颜色及背景颜色不敏感，跟踪框更准确。如下图所示场景中，目标颜色基本为白色，与背景及灯光的颜色相近，颜色滤波类方法的跟踪框易将背景框入。



图 12 该算法在复杂颜色下的鲁棒性



图 13 颜色滤波类算法在复杂颜色下的鲁棒性

(4) 背景差的方法对分辨率的要求小，即使在低分辨率环境下也能达到较好的效果，而其他算法在一定程度上受分辨率的影响。



图 14 该算法在低分辨率下的鲁棒性



图 15 梯度滤波类算法在低分辨率下的鲁棒性

(5) 考虑到暗光环境下的背景噪声较多，在重建背景图像的过程中，使用了加权重建的方法来滤除高斯白噪声，使背景图像更加稳定；加权重建具有更好的抗干扰性，抑制了当跟踪目标框选过小时，目标框外图像对背景的影响。

4 算法缺点

- (1) 需要一定的初始化过程等待算法重建背景。
- (2) 对背景噪声有一定的敏感，不能用于非固定背景的情况。