

# 基于高斯混合模型联合 CamShift 的运动图像检测跟踪方法

张婧懿<sup>1</sup>, 隋思逸<sup>2</sup>

(1. 黑龙江中医药大学, 哈尔滨 150040; 2. 黑龙江中医药大学附属医院, 哈尔滨 150040)

**摘要:** CamShift 算法是典型的运动图像跟踪方法, 但是单纯 CamShift 准确率低, 容易丢失目标; 将高斯混合模型运用到 CamShift 算法中进行目标跟踪操作, 提高准确性; 首先采用高斯混合模型标示出目标局部区域, 并将其作为 CamShift 的初始搜索窗, 提高效率; 随后对目标进行跟踪时将 CamShift 算法的窗中心同差分法计算出的目标区域中心作对比, 确定后续帧搜索窗, 避免目标跟踪丢失; 最后, 实验证明了该方法可以对目标进行有效跟踪, 且在目标颜色同背景色差异小的情况下依然具有非常高的准确率。

**关键词:** 高斯混合模型; CamShift 算法; 运动检测

## Moving Image Detection and Tracking Method Based on Gauss Mixed Model Combined With CamShift

Zhang Jingyi<sup>1</sup>, Sui Siyi<sup>2</sup>

(1. Heilongjiang university of traditional Chinese medicine, Harbin of China, 150040;

2. The first affiliated hospital of heilongjiang university of traditional Chinese medicine, Harbin of China, 150040)

**Abstract:** The CamShift algorithm is a typical method of moving image tracking, but the accuracy of CamShift is low and it is easy to lose the target. The Gauss hybrid model is applied to the CamShift algorithm to carry out the target tracking operation to improve the accuracy. First the Gauss mixture model marked the local area, and the CamShift as the initial search window, improve efficiency; then track the target when the target area center CamShift algorithm with window center difference method to calculate the comparison, determine the subsequent search window frame, avoid the loss of target tracking. Finally, experiments show that this method can track the target effectively, and still has very high accuracy in the case that the color difference between the target and the background is small.

**Keywords:** Gauss mixed model; CamShift algorithm; Motion detection

## 0 引言

新时期, 移动互联网技术与人工智能技术飞速发展, 为生产生活提供了巨大的便利。作为人工智能的重要分支, 运动目标跟踪方法将图像处理、智能计算以及模式识别等众多技术相融合, 在众多领域应用广泛, 如: 安全监控、目标探测、车辆跟踪等。CamShift 算法是典型的特征跟踪算法, 依据目标物的颜色特征进行识别跟踪, 其运算复杂度低, 实时获取跟踪结果, 且性能稳定。不过, 一旦目标物的颜色与背景色、目标物颜色相似, 或者存在遮挡的情况下, 就会出现目标丢失、准确性大大降低的问题<sup>[1-4]</sup>。

徐琨等人在 2009 年时提出了基于 CamShift 的自适应颜色空间目标跟踪算法, 依据当前量和类间平均距离动态地选取当前的颜色空间, 克服了只适用于单个颜色空间的缺陷<sup>[5]</sup>。倪麒等人在 2010 年将高斯混合模型同 CamShift 算法结合, 通过背景差分提取出运动物体所在的局部区域避免了手动选取的缺陷<sup>[6]</sup>。王冉等人则在 2012 年提出将运动预测与 Camshift 算法相结合以提高对于运动目标的跟踪准确率<sup>[7]</sup>。本文将提出一种

将高斯混合模型引入 CamShift 算法中的优化方法, 以解决在目标与背景差异较小情况下, 目标丢失率升高的问题, 并通过实验证明该方法的可行性与准确性。

## 1 CamShift 动态图像跟踪

### 1.1 颜色概率分布提取

本实验中, 为统计图像颜色概率分布情况, 我们先将图像由 RGB 颜色空间转换为 HSV 颜色空间。因为 HSV 色度空间在色度、饱和度以及亮度三个分量上三者之间相互独立并且可以很好的反映出图像的灰度信息以及色彩信息, 且 HSV 的 H (色度) 分量可以比较好的反应出目标色彩信息, 可将 H 分量作为统计目标颜色直方图的考量依据。

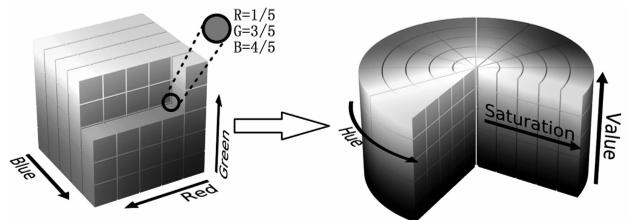


图 1 RGB 颜色空间转换 HSV 颜色空间

具体转换过程中, 先把原图中的像素点替换为对应的颜色直方图以作后续统计, 最终将其归一化为  $[0 \ 255]$ , 最终获取到图像颜色概率分布图<sup>[8-12]</sup>。具体步骤如下:

收稿日期: 2018-03-09; 修回日期: 2018-03-30。

**作者简介:** 张婧懿(1984-), 女, 辽宁昌图人, 硕士, 供职于黑龙江中医药大学, 主要从事中医药信息工程方向的研究。

**通讯作者:** 隋思逸(1990-), 男, 黑龙江哈尔滨人, 主要从事计算机图像处理与计算机控制技术方向的研究。

- 1) 根据查找轮廓, 读入一帧图片, 预览;
- 2) 将图片中所有颜色换成 HSV 色度空间;
- 3) 设定识别颜色的取值范围, 分成 R、G、B 三种色彩;
- 4) 在图片中查找相关颜色, 转换成二值图, 并将所得的二值图像四边都分别增加一个像素, 设定为查找轮廓;
- 5) 重复上面步骤, 直到所有颜色都识别完毕。



图 2 Camshift 图像跟踪原理

### 1.2 CamShift 目标搜索

在跟踪搜索领域, CamShift 算法是一种常见的根据目标特征进行目标跟踪的算法, 该算法以目标图像颜色的直方图为依据, 只要通过在概率分布中不断迭代, 就可以最终寻找到极值并进行目标定位。如图 2 为 CamShift 图像跟踪原理, CamShift 算法思想主要包含如下几点:

- 1) 目标局部发生位移, 影响目标颜色概率分布;
- 2) 实时计算、统计整个目标不同颜色的概率分布情况, 分析目标变化情况;
- 3) 将历史 frame 的计算值, 作为实时计算的 frame 的 Search Window 的初始值;
- 4) 对搜索框中的颜色分布进行计算和分析, 实时、灵活调整搜索框位置和大小。

### 1.3 搜索质心位置

本设计中, 首先分析图像颜色概率分布确定目标搜索窗大小, 假定其为  $s$ ; 计算出所采用的搜索窗的质心坐标, 其可以采用以上求得的概率分布图的零阶矩和一阶矩求出来, 其零阶矩如下:

$$M_{00} = \sum_x \sum_y I(x, y) \quad (1)$$

而  $x$  以及  $y$  的一阶矩如下所示:

$$\begin{cases} M_{10} = \sum_x \sum_y xI(x, y) \\ M_{01} = \sum_x \sum_y yI(x, y) \end{cases} \quad (2)$$

通过以上方法求得的搜索窗质心为:

$$x_c = \frac{M_{10}}{M_{00}}, y_c = \frac{M_{01}}{M_{00}}$$

得出搜索框质心位置, 这其中,  $I(x, y)$  是像素点  $(x, y)$  的值, 其横坐标和纵坐标在搜索窗内。

### 1.4 搜索窗调整

搜索窗的大小需要根据需要进行适当调整, 其大小根据下式重新确定:

$$s = 2\sqrt{\frac{M_{00}}{256}} \quad (3)$$

重新确定的搜索窗的大小为:

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}} \quad (4)$$

$$\omega = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \quad (5)$$

这其中:

$$a = \frac{M_{20}}{M_{11}} - x_c^2, b = 2(\frac{M_{11}}{M_{00}} - x_c y_c), c = \frac{M_{02}}{M_{11}} - y_c^2$$

$l$  是新确定的搜索窗的长, 而  $\omega$  是新的搜索窗的宽度。

## 2 高斯混合模型 CamShift 算法

### 2.1 高斯混合图像模型

高斯模型用高斯概率密度函数精确量化图像, 将图像分解为若干个基于高斯概率密度函数形成的模型。为提高模型的精确性, 本文在模型中引入权值均值, 建立背景图像并实时更新, 后续处理中结合权值、权值均值和背景图像等因素对像素点进行前景和背景的分类。所以:

$$D_k(x, y) = |f_k(x, y) - B_k(x, y)| \quad (6)$$

并且根据如下公式进行阈值判断:

$$R_k(x, y) = \begin{cases} 0 & \text{背景 } D_k(x, y) < \text{阈值} \\ 1 & \text{前景 } D_k(x, y) \geq \text{阈值} \end{cases} \quad (7)$$

这其中,  $f_k$  是当前的图像帧, 而  $D_k$  则是差分得出的图像,  $R_k$  是二值化后的图像,  $B_k$  则是图像的背景。

高斯混合模型的核心是能否构建一个合理的背景模型并且进行背景模型的更新操作。一般构建一个合适的背景模型是通过对一个时间段内的背景图像的累积获取的: 第一步是获取一段不存在运动物体的视频序列, 将这一背景序列的点  $(x, y)$  的像素值的和表示为  $S(x, y)$ , 而用  $S_q(x, y)$  代表这一背景序列中像素点  $(x, y)$  的平方和, 两者分别如下表示:

$$S(x, y) = I_1(x, y) + I_2(x, y) + I_3(x, y) + \dots + I_k(x, y) \quad (8)$$

$$S_q(x, y) = I_1^2(x, y) + I_2^2(x, y) + I_3^2(x, y) + \dots + I_k^2(x, y) \quad (9)$$

针对于高斯分布的背景模型的均值和方差如下计算:

$$m(x, y) = \frac{S(x, y)}{N} \quad (10)$$

$$\sigma(x, y) = \sqrt{\frac{S_q(x, y)}{N} - m^2(x, y)} \quad (11)$$

其中,  $N$  是整个背景序列中背景图的数量。

在整个环境中, 随着时间的变化, 背景也会发生改变。所以为了避免系统产生较大的误差, 需要对背景进行相应的更新, 这里采用自适应的背景模型更新方法, 其公式如下表示:

$$\begin{cases} m_{k+1}(x, y) = (1 - \alpha) \times m_k(x, y) + \alpha \times I_k(x, y), R_k(x, y) \neq 0 \\ m_{k+1}(x, y) = m_k(x, y), R_k(x, y) = 0 \end{cases} \quad (12)$$

$$\begin{cases} \sigma_{k+1}^2(x, y) = (1 - \alpha) \times \sigma_k^2(x, y) + \alpha \times [m(x, y) - I_k(x, y)]^2, \\ R_k(x, y) \neq 0 \\ \sigma_{k+1}^2(x, y) = \sigma_k^2(x, y), R_k(x, y) = 0 \end{cases} \quad (13)$$

公式内的系数  $\alpha$  是更新速度, 在采用自适应的背景更新算法时需要将运动物体排除在背景模型的更行之外, 而且不能要求每一个图像帧都进行背景模型的更新<sup>[17-18]</sup>。

### 2.2 基于高斯混合模型的 CamShift 运动目标跟踪方法

采用高斯混合模型对一个移动的目标进行检测, 先对图像进行二值化处理, 再根据高斯混合模型计算得出的当前帧的背景图像。高斯混合模型可以得出每一帧的当前背景进而提取出目标区域中的移动目标部分, 进而实现运动目标的跟踪监测。高斯混合模型联合 CamShift 运动图像跟踪检测实施步骤如下:

- 1) 高斯混合模型标示出跟踪区域, 作为后续算法跟踪初始搜索窗;

2) 计算目标跟踪区域颜色概率分布数据;

3) 用 CamShift 算法计算获取下一步搜索窗具体位置、长宽等数据;

4) 计算出运动目标所在矩形区域的中心、纵横坐标、搜索窗等参数的绝对差值;

5) 判断步骤 4 中得到的差值是否超出设定的阈值: 若差值大于阈值, 则采用第一步中获取的矩形框作为下一个搜索窗, 否则的话将步骤 (的搜索窗作为下一个搜索窗, 并从第二步重新计算得出结果。

### 3 实验与分析

#### 3.1 代码实现

本次实验智能高清摄像机采用海康威视某型号 200 W 像素 CCD, 操作系统为 Linux 操作系统。算法部分在 Linux ubuntu 操作系统下 (采用 ubuntu 操作系统的原因在于该系统裁剪性好, 且对多种接口友好度高), 采用 OpenCV2.3 视觉库编程设计实现, 将实现好的软件合并到智能高清摄像机中。其部分代码如下:

```
import rospy
import cv2from cv2
import cv as cvfrom rbx1_vision.ros2opencv2
import ROS2OpenCV2from std_msgs.msg
import Stringfrom sensor_msgs.msg import Image
import numpy as np
class CamShiftNode(ROS2OpenCV2):
def __init__(self, node_name):
ROS2OpenCV2.__init__(self, node_name)
self.node_name = node_name
The minimum saturation of the tracked color in HSV space,
as well as the min and max value (the V in HSV) and a
threshold on the backprojection probability image.
self.smin = rospy.get_param("~smin", 85)
self.vmin = rospy.get_param("~vmin", 50)
self.vmax = rospy.get_param("~vmax", 254)
self.threshold = rospy.get_param("~threshold", 50)
```

#### 3.2 测试参数

测试设备: 海康威视某型号 200 W 像素 CCD 高清摄像机;

测试环境: 某地级市某狭窄双向行驶道路;

照度仪检测值: 88lux—125lux;

采集开始时间: 16 时 32 分;

视频帧率: 75 fps;

跟踪目标数: 4 个;

拍摄时, 摄像头位置较为固定, 可理想化为稳定状态;

设定: 跟踪目标质心为黄色、检测框为绿色、检测调整框为灰色。

我们对设计的目标跟踪方法进行实验验证, 验证可行性以及准确性。可行性的验证设计为使用该方法对一段录制好的视屏进行动态目标的跟踪提取, 以判断其是否可以完成运动目标的跟踪目的。准确性是验证跟踪目标不丢失, 且维持较长的跟踪时间。图 3 为 Ubuntu 系统下编程示例。

#### 3.3 验证结果分析

验证结果如图 4 所示, 其中正方形方框为检测框, 较粗且不规则线段为跟踪目标质心移动轨迹。

```
started roslaunch server http://zxwubuntu-Aspire-V3-572G:42711/
SUMMARY
=====
PARAMETERS
 * /camshift/smin: 85
 * /camshift/threshold: 50
 * /camshift/vmax: 254
 * /camshift/vmin: 50
 * /roscd: /opt/ros/indigo
 * /rosversion: 1.11.10
NODES
  /
    camshift (/rbx1_vision/camshift.py)
ROS_MASTER_URI=http://localhost:11311
core service [/roscout] found
process[camshift-1]: started with pid [4450]
[INFO] [walltime: 1461071378.622548] Starting node camshift
```

图 3 Ubuntu 系统下编程示例

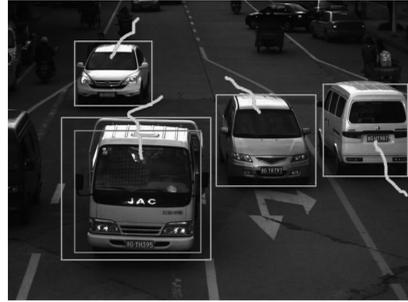


图 4 低照度环境下较大画面动态目标跟踪结果

#### 1) 可行性:

可以看出, 随着镜头画面中跟踪物的不断移动, 正方形搜索窗一直标示着运动的目标, 这证明了所提方法可以完成对运动目标进行跟踪的目的, 充分证明方法是可行且有效的。

且在跟踪过程中, 黄色质心点随着时间的推移构成了四条连续、未间断的线条轨迹, 且没有出现离散点。这说明在跟踪过程没有出现丢帧以及目标丢失的情况。

#### 2) 准确性:

本次测试实在黄昏时间段, 光照条件较差, 照度仅为 88lux—125lux (从光学角度来看, 该光照数值说明光照条件较差, 属于恶劣光照环境)。测试环境中, 不断出现行人、自行车、电瓶车等移动物体, 干扰性较大, 目前算法可以准确跟踪到画面中的移动物体。

四个跟踪目标中, 左上角白色小车与 JAC 小型卡车均整体位于检测框内, 左起第三辆银色小车的上半检测框侵入车辆图像上边缘内, 最右边银色面包车的检测框上边缘和左边缘均有一定程度的侵入。说明检测框跟踪效果较好, 没有出现目标的丢失, 且检测框的准确性较高 (没有出现 30% 以上的误差)。四个跟踪目标中仅有 JAC 牌小型卡车出现了检测调整框, 说明检测框调整及时且准确。

四个跟踪目标的质心轨迹均平滑、线性, 没有出现离散的质心点, 这说明跟踪的效果相比较于普通的跟踪算法来说, 有较大的性能优势。

#### 3.4 结果讨论

实验环节准确、充分证明了高斯混合模型联合 CamShift 算法的可行性, 于以往跟踪算法相比, 其优势如下:

1) 高斯混合模型的加入, 使得 CamShift 算法的性能大幅度提升, 主要体现在丢失率的降低和连续跟踪能力的增强;

2) RGB 颜色空间到 HSV 颜色空间的转换使图像采集数

(下转第 65 页)