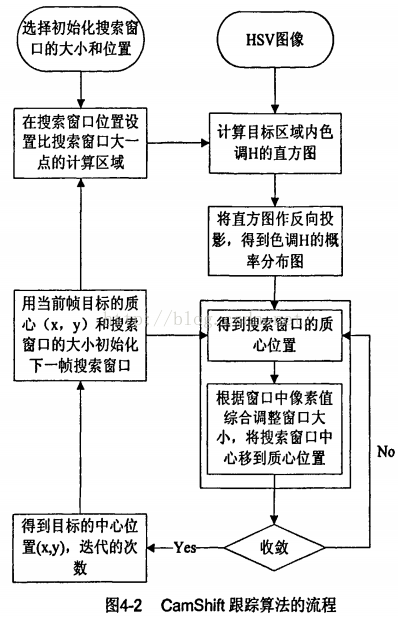
一、概述

    CamShift算法，全称是 Continuously AdaptiveMeanShift，顾名思义，它是对Mean Shift 算法的改进，能够自动调节搜索窗口大小来适应目标的大小，可以跟踪视频中尺寸变化的目标。它也是一种半自动跟踪算法，需要手动标定跟踪目标。基本思想是以视频图像中运动物体的颜色信息作为特征，对输入图像的每一帧分别作 Mean-Shift 运算，并将上一帧的目标中心和搜索窗口大小(核函数带宽)作为下一帧 Mean shift 算法的中心和搜索窗口大小的初始值，如此迭代下去，就可以实现对目标的跟踪。因为在每次搜索前将搜索窗口的位置和大小设置为运动目标当前中心的位置和大小，而运动目标通常在这区域附近，缩短了搜索时间；另外，在目标运动过程中，颜色变化不大，故该算法具有良好的鲁棒性。已被广泛应用到运动人体跟踪，人脸跟踪等领域。

二、算法流程



    具体步骤如下：

    步骤一：计算目标区域内的颜色直方图。  
    通常是将输入图像转换到HSV颜色空间，目标区域为初始设定的搜索窗口范围，分离出色调H分量做该区域的色调直方图计算。因为 RGB 颜色空间对光线条件的改变较为敏感，要减小该因素对跟踪效果的影响，CamShift 算法通常采用 HSV 色彩空间进行处理，当然也可以用其它颜色空间计算。这样即得到目标模板的颜色直方图。

    步骤二：根据获得的颜色直方图将原始输入图像转化成颜色概率分布图像，  
    该过程称为“反向投影"。所谓直方图反向投影，就是输入图像在已知目标颜色直方图的条件下的颜色概率密度分布图，包含了目标在当前帧中的相干信息。对于输入图像中的每一个像素，查询目标模型颜色直方图，对于目标区域内的像素，可得到该像素属于目标像素的概率，而对于非目标区域内的像素,该概率为0。

    步骤三：Mean Shift迭代过程。

        即右边大矩形框内的部分，它是 CamShift 算法的核心，目的在于找到目标中心在当前帧中的位置。首先在颜色概率分布图中选择搜索窗口的大小和初始位置，然后计算搜索窗口的质心位置。设像素点(i, j)位于搜索窗口内，I(i, j)是颜色直方图的反向投影图中该像素点对应的值，定义搜索窗口的零阶矩 M00 和一阶矩M10,M01如下：



   则搜索窗口的质心位置为:(M10/M00, M01/M00)。

    接着调整搜索窗口中心到质心。零阶矩反映了搜索窗口尺寸，依据它调整窗口大小，并将搜索窗口的中心移到质心，如果移动距离大于设定的阈值，则重新计算调整后的窗口质心，进行新一轮的窗口位置和尺寸调整。直到窗口中心与质心之间的移动距离小于阈值，或者迭代次数达到某一最大值，认为收敛条件满足，将搜索窗口位置和大小作为下一帧的目标位置输入，开始对下一帧图像进行新的目标搜索。

三、总结

    CamShift算法改进了 Mean-Shift 跟踪算法的第二个缺陷，在跟踪过程中能够依据目标的尺寸调节搜索窗口大小，对 有尺寸变化的目标可准确定位。但是，一方面 CamShfit 算法在计算目标模板直方图分布时，没有使用核函数进行加权处理，也就是说目标区域内的每个像素点在目标模型中有着相同的权重，故 CamShfit 算法的抗噪能力低于Mean-Shift跟踪算法。另一方面，CamShift 算法中没有定义候选目标，直接利用目标模板进行跟踪。除此以外，CamShift 算法采用 HSV 色彩空间的H分量建立目标直方图模型，仍然只是依据目标的色彩信息来进行跟踪，当目标与背景颜色接近或者被其他物体遮挡时，CamShift 会自动将其包括在内，导致跟踪窗口扩大，有时甚至会将跟踪窗口扩大到整个视频大小，导致目标定位的不准确，连续跟踪下去造成目标的丢失。