

# Mean-Shift 目标跟踪算法的带宽选取策略

胡淑芳<sup>1,2</sup>, 袁宝峰<sup>3</sup>, 屈源<sup>4</sup>

(1. 重庆大学计算机学院, 重庆 400044; 2. 重庆理工大学设备处, 重庆 400050;  
3. 重庆通信学院信号与信息处理实验室, 重庆 400035; 4. 重庆警备区, 重庆 400010)

**摘要:** 分析不同带宽策略对 Mean-Shift 目标跟踪算法的影响, 提出一种新的带宽策略。构造一个与目标窗口内接椭圆相似的椭圆, 利用其长轴和短轴确定带宽, 将此带宽策略纳入 Mean-Shift 算法中进行目标跟踪。实验结果表明, 该策略具有良好的跟踪效果, 能够有效降低时间复杂度。

**关键词:** 带宽; Mean-Shift 算法; 目标跟踪

## Bandwidth Choosing Strategy of Mean-Shift Target Tracking Algorithm

HU Shu-fang<sup>1,2</sup>, YUAN Bao-feng<sup>3</sup>, QU Yuan<sup>4</sup>

(1. College of Computer, Chongqing University, Chongqing 400044, China;  
2. Equipment Department, Chongqing University of Technology, Chongqing 400050, China;  
3. Signal and Information Processing Lab, Chongqing Communication Institute, Chongqing 400035, China;  
4. Chongqing Garrison Command, Chongqing 400010, China)

**【Abstract】** The influence of different band strategies are analyzed. A new bandwidth strategy is put forward. Through structuring an ellipse being similar with target window's inscribes ellipse, it uses major axis and minor axis to identify bandwidth. This bandwidth strategy is integrated into Mean-Shift target tracking algorithm. Experimental results show this strategy has good effect on tracking target and reduces the time complexity.

**【Key words】** bandwidth; Mean-Shift algorithm; target tracking

DOI: 10.3969/j.issn.1000-3428.2011.02.063

### 1 概述

Mean-Shift 跟踪算法是一种基于模板匹配的目标跟踪算法<sup>[1-3]</sup>, 它的核心思想是将特征的核密度估计漂移到局部极大值点。由于其计算简单、对目标旋转、部分遮挡具有一定鲁棒性, 近年来成为目标跟踪领域的研究热点。Mean-Shift 目标跟踪算法将颜色或纹理<sup>[4]</sup>作为目标特征, 利用核函数构建目标模板的核密度估计, 在目标候选区域建立候选目标的核密度估计, 通过 Bhattacharyya 系数的最大化来确定下一时刻目标的位置。在目标模板区域, 标定矩形内接椭圆内的像素点会参与到核密度估计的计算中, 在目标候选区域, 通常是在一个以上一帧目标中心为圆心的圆形区域内计算待匹配区域的核密度估计。此时该圆形区域半径的大小即决定了参与计算的像素点数以及目标尺度的大小, 该半径即是算法的窗口带宽。对窗口带宽的研究主要针对自适应变化<sup>[5-7]</sup>问题, 很少有文献谈及带宽的取值策略。但在实际应用中, 带宽的取值直接关系到参与计算的像素的数量和背景噪声混入的程度, 故对此方面的研究很有必要。

### 2 核函数带宽取值的讨论

#### 2.1 传统带宽策略

在候选目标区域, 目标核密度估计为:

$$\hat{p}_u(y) = C_h \sum_{i=1}^N k\left(\left\|\frac{y-x_i}{h}\right\|^2\right) \delta(b(x_i)-u) \quad (1)$$

其中,  $C_h = \frac{1}{\sum_{i=1}^N k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)}$ 。在式(1)中,  $h$  即为窗口带宽, 它

的取值决定了参与核密度估计计算的像素点数, 同时也反映了目标的尺度。

传统的带宽策略有 3 种: (1) 以上一帧窗口的半短边为带宽; (2) 以上一帧窗口的半长边为带宽; (3) 以上一帧窗口的半对角线为带宽。下面以一个汽车视频序列为例, 说明不同带宽策略对跟踪效果和时间复杂度的影响, 如图 1 和表 1 所示。



图 1 不同的带宽策略

表 1 不同带宽策略的比较

带宽取值	总像素点 /(个)	背景噪声 像素点/(个)	目标像素点 /(个)	目标像素点 比例/(%)
$h=17$ (以半短边 为带宽)	1 003	147	856	68.7
$h=25$ (以半长边 为带宽)	2 054	850	1 204	96.6
$h=30$ (以半对角线 为带宽)	2 578	1 332	1 246	100

以半短边为带宽的带宽策略含目标像素点比例过小, 不

**基金项目:** 国家“863”计划基金资助项目(2007AA01Z423); 重庆市自然科学基金资助项目(CSTC,2007BB2134)

**作者简介:** 胡淑芳(1981-), 女, 硕士研究生, 主研方向: 机器学习, 模式识别, 图像处理; 袁宝峰、屈源, 硕士研究生

**收稿日期:** 2010-05-20 **E-mail:** hushufang@cqu.edu.cn

予考虑；以半长边和半对角线为带宽的策略大大提高了所含目标像素点的比例，但同时引入了大量的背景噪声。

目标初始窗口为  $51 \times 35$ ，目标尺度越来越小，在仿真实验中使用  $\pm 10\%$  [1] 的自适应带宽技术。可以看出，2种带宽策略均能较好地跟踪目标，但由于以半长边为带宽的策略丢失了部分目标像素，因此图2的跟踪效果略逊于图3。表2反映出2种不同带宽策略下的时间复杂度。



图2 以半长边为带宽时的跟踪效果



图3 以半对角线为带宽时的跟踪效果

表2 不同带宽情况下的跟踪时间复杂度

带宽取值	总帧数	总时间/s	平均时间/(s·f <sup>-1</sup> )
$h=25$ (以半长边为带宽)	60	84.5	1.41
$h=30$ (以半对角线为带宽)	60	95.9	1.59

2.2 新的带宽策略

设上一帧的目标窗口长和宽分别为  $h_x$  和  $h_y$ ，则上一帧图像的目标窗口的内接椭圆可以用式(2)表示：

$$\frac{x^2}{h_x^2} + \frac{y^2}{h_y^2} = 1 \tag{2}$$

如图4所示， $OT$ 为上一帧目标窗口的长边  $h_y$ ， $OQ$ 为前一帧目标窗口的短边  $h_x$ ，矩形  $ABCD$ 为上一帧图像的目标窗口，椭圆  $OTQ$ 为其内接椭圆，以  $O$ 为圆心，作一个与椭圆  $OTQ$ 相似的椭圆  $OPH$ ，其长轴为  $OP$ ，短轴为  $OH$ 。

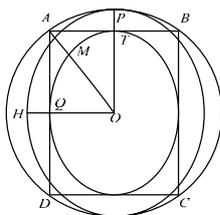


图4 相似椭圆示意图

因为椭圆  $OTQ$  与椭圆  $OPH$  相似，所以有：

$$\frac{OP}{OH} = \frac{OT}{OQ} = \frac{h_y}{h_x} \tag{3}$$

而  $OP = OA = \sqrt{h_x^2 + h_y^2}$ ，所以  $OH = h_x \sqrt{h_x^2 + h_y^2} / h_y$ ，故目标候选区域为一长轴  $h'_y$  为  $\sqrt{h_x^2 + h_y^2}$ 、短轴  $h'_x$  为  $h_x \sqrt{h_x^2 + h_y^2} / h_y$  椭圆，则目标候选区域的核密度估计为：

$$\hat{\rho}_u(y) = C_h \sum_{i=1}^N k(\|x_i^*\|^2) \delta(b(x_i) - u) \tag{4}$$

其中， $C_h = \frac{1}{\sum_{i=1}^N k(\|x_i^*\|^2)}$ ； $x_i^*$ 为利用目标候选区域的长轴  $h'_y$  和短轴  $h'_x$  对  $X$  坐标和  $Y$  坐标分别进行归一化后的位置坐标。

3 新的带宽策略纳入到跟踪算法

将新的带宽策略纳入到 Mean-Shift 跟踪算法框架中。目标模板的核密度估计  $\hat{q}_u$  为：

$$\hat{q}_u = C \sum_{i=1}^N k(\|x_i^*\|^2) \delta[b(x_i^*) - u] \tag{5}$$

其中， $C = \frac{1}{\sum_{i=1}^N k(\|x_i^*\|^2)}$ 。

(1)初始化第1帧目标所在位置，计算核密度估计  $\hat{\rho}_u(y)$  和 Bhattacharyya 相似度系数。

$$\rho(\hat{q}_u, \hat{\rho}_u(y_0)) = \sum_{u=1}^m \sqrt{\hat{q}_u \hat{\rho}_u(y_0)} \tag{6}$$

(2)按照(7)式计算目标候选区域像素点权值  $\omega_i$ ：

$$\omega_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{\rho}_u(y)}} \delta(b(x_i) - u) \tag{7}$$

(3)参照前面的带宽策略计算当前帧的目标候选区域的长轴  $h'_y$  和短轴  $h'_x$ 。

(4)按照式(8)计算目标在下一帧图像中的形心位置：

$$y_1 = \frac{\sum_{i=1}^N x_i \omega_i g(\|x_i^*\|^2)}{\sum_{i=1}^N \omega_i g(\|x_i^*\|^2)} \tag{8}$$

其中， $g(x) = -k'(x)$ ； $x_i^*$ 为利用目标候选区域的长轴  $h'_y$  和短轴  $h'_x$  对  $X$  坐标和  $Y$  坐标分别进行归一化后的位置坐标。

(5)按照(4)式计算新的目标候选区域核密度估计  $\hat{\rho}_u(y)$  和目标模板的 Bhattacharyya 相似度系数：

$$\rho(\hat{q}_u, \hat{\rho}_u(y_1)) = \sum_{u=1}^m \sqrt{\hat{q}_u \hat{\rho}_u(y_1)}$$

(6)当  $\rho(\hat{q}_u, \hat{\rho}_u(y_1)) < \rho(\hat{q}_u, \hat{\rho}_u(y_0))$ ，则  $y_1 = \frac{1}{2}(y_0 + y_1)$ ，然后再计算  $\rho(\hat{q}_u, \hat{\rho}_u(y_1))$ 。

(7)如果  $\|y - y_1\| < \epsilon$ ，则停止算法，否则  $y_0 = y_1$ ，转到(2)。

根据文献[1]，第(6)步在实际应用中对目标位置的修正作用不大，同时为减小时间复杂度，舍掉第(6)步。

4 实验仿真

为了验证所提出的带宽策略的有效性，分别采用文献[1]的半对角线带宽策略和本文的带宽策略对2个视频序列进行跟踪实验。本文实验仿真平台为 VC++6.0，计算机配置为 Pentium 4 CPU 2.40 GHz，内存为 256 MB。

对于汽车视频，传统带宽策略的跟踪结果见图3，其余结果如图5~图7所示。实验结果表明，本文带宽策略在保证跟踪效果的基础上，有效地降低了时间复杂度。



图5 本文带宽策略的跟踪结果(汽车)

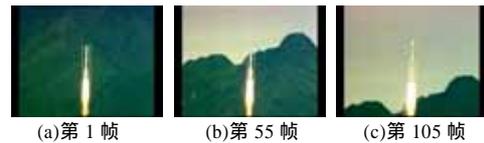


图6 传统带宽策略的跟踪结果(火箭)



图7 本文带宽策略的跟踪结果(火箭)