

# Mean-Shift 算法的原理及用于跟踪的实验

June 17, 2014

## 1 问题

本文讲解 Mean-Shift 算法的原理，及在目标跟踪中的应用。主要参考《Learning OpenCV》一书<sup>1</sup>中的讲解及 OpenCV 的源码。

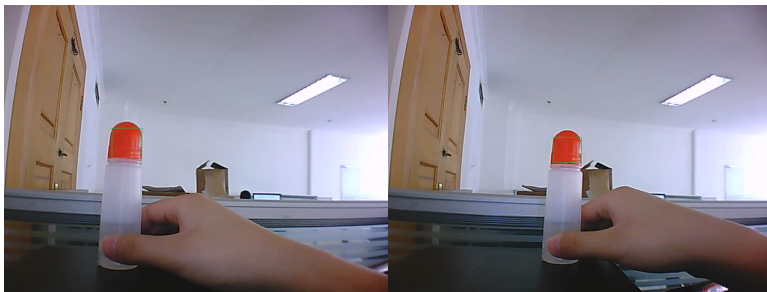
## 2 解答

### 2.1 Mean-Shift 的本质

Mean-Shift 是个算法，本质是个方法（工具）。用来求局部极值的方法（工具）。是众多数值分析方法中的一个。

### 2.2 为什么要求局部极值？

从目标跟踪这个问题的定义开始：



---

<sup>1</sup>位于第十章“运动与跟踪”第二节

- 追踪场景中胶水瓶的盖子
- 将图像表示成概率图，每个像素点都是一个概率值，代表该点是目标的可能性的<sup>2</sup>大小
- 初始选定目标所在位置，中心位于矩形的对角线中心（几何中心）

问题：下一帧中，中心的位置？

如果能找到概率最大(可能性最大)的位置，就认为目标是所在位置<sup>2</sup>。

## 2.3 颜色概率模型

上面说的颜色概率模型，生成过程如下：

1. 生成目标的直方图。
2. 根据直方图各区间内像素的多少,为该区间分配一个概率 ( $p_i = \frac{\text{当前区间像素数}}{\text{所有像素数}}$ )。
3. 遍历原图，根据像素点所在区间，为其分配概率。这一过程称为 back-projection。
4. 新建图像，存储每一点的概率，得到概率图。

具体可参考 opencv 文档 + 代码。-calcBackProject 接口。

## 2.4 直观的理解

确定下一个中心的位置，实际是个投票的过程。扫描窗口中的每个点都要发言。故事是这样的：每个点都认为，下一个中心应该向自己靠拢，自己才是世界的中心。听谁的？——谁的钱多(权值大)听谁的。

权值是什么？就是该点的概率。

如果你是红色，你很可能是目标，所以，中心向你那儿移动多点；

如果你是紫色，你有可能是目标，所以，中心向你那儿移动少点；

如果你是黑色，你不可能是目标，所以，中心不会向你那儿移动。

<sup>2</sup>注意：此处不是单个最大值点，否则直接遍历即可。谁是最大值，一个点说了不算，得所有点投票。

## 2.5 直观上加点数学，质心

上述的过程，实际就是求质心的过程。求中心的公式，就是求质心的公式：

$$\bar{x} = \sum x_i * \frac{P_i}{\sum P_i} \quad (1)$$

$$\bar{y} = \sum y_i * \frac{P_i}{\sum P_i} \quad (2)$$

以  $x$  为例，其中：

- $x_i$  表示每一点的横坐标
- $P_i$  表示该点是目标的概率
- $\sum P_i$  是所有点概率之和，显然， $P_i / \sum P_i$  表示该点在所有点中的影响力
- $x_i * (P_i / \sum P_i)$ ，即表示往自己  $(x_i)$  偏移的多少

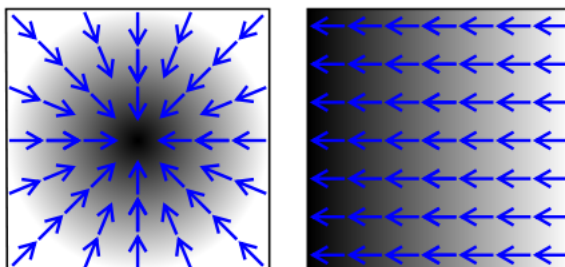
最终的结果， $\bar{x}$ ，显然受  $P_i$  值较大的点影响较大。多么民主的过程！

## 2.6 直观上再加点数学，梯度

不知道怎么开口，也不知道怎么跟你说，但是求极值就是绕不开梯度（导数）。梯度的定义非常有意思：

- 方向 (direction)：朝着函数值变化最快的方向
- 大小 (magnitude)：等于该点切线的斜率

参考维基百科的示意图<sup>3</sup>：



<sup>3</sup><http://en.wikipedia.org/wiki/Gradient>

梯度的方向就是颜色加深最快的方向；梯度的大小等于  $\Delta y / \Delta x$ ，即该点竖直方向的变化比上水平方向的变化，也就是切线斜率。

参考维基百科的例子：

Consider a room in which the temperature is given by a scalar field,  $T$ , so at each point  $(x,y,z)$  the temperature is  $T(x,y,z)$ . (We will assume that the temperature does not change over time.) At each point in the room, the gradient of  $T$  at that point will show the direction the temperature rises most quickly. The magnitude of the gradient will determine how fast the temperature rises in that direction.

以房间中的温度为例，是个标量场，用  $T$  来表示，任一点的温度是  $T(x,y,z)$ 。方便分析，假设房间里各点温度保持不变。某一点的梯度方向是什么？就是该点处，温度上升最快的方向。那什么是梯度的 Magtitude？就是在该方向上温度变化的有多快。

从梯度的角度考虑确定了：

- 往哪里走？——梯度的方向。
- 走多远？——梯度的大小。

两个问题，就能找到极值点了。

那如何求梯度呢？求导。对谁求导呢？对概率密度函数求导<sup>4</sup>。那怎么得到概率密度函数呢？——核密度估计。

## 2.7 如何求概率密度函数？

核密度估计是从有限样本，得到这些样本所服从的概率分布——通过估计概率密度函数来实现<sup>5</sup>。

## 2.8 公式推导

核密度估计在  $x_p$  点的值的估计 (以高斯核为例)：

$$P(x_p) = \frac{1}{N} * \frac{1}{\sqrt{2\pi\sigma^2}} * \sum_{i=1}^N e^{-\frac{(x_p - x_i)^2}{2 * \sigma^2 * h^2}} * P(x_i) \quad (3)$$

<sup>4</sup>图像被表示成了概率图，所以梯度的方向指向概率上升最快的方向。此时，中心就是样本分布最集中的地方，如二维高斯分布的中心。这个中心，在跟踪中，就认为是目标。

<sup>5</sup>[http://xueyayang.github.io/pdf\\_posts/核密度估计的概念与实验.pdf](http://xueyayang.github.io/pdf_posts/核密度估计的概念与实验.pdf)

[http://xueyayang.github.io/pdf\\_posts/2 维核密度估计实验.pdf](http://xueyayang.github.io/pdf_posts/2 维核密度估计实验.pdf)

注意其中的  $P(x_i)$ ，在脚注 4 的实验里是没有的。它表示样本点的概率。其意义是明显的：样本点的来自目标的可能性越大，对估计结果的影响越大；如果样本点来自目标的概率为 0（噪声），则不该对估计结果产生影响。脚注 4 中的实验之所以没有这一项，是因为所有的样本点都是有效的。抽象一些，核用  $k(\cdot)$  来代替，写成更一般的形式：

$$P(x_p) = ck(\|\frac{(x_p - x_i)^2}{h}\|) \quad (4)$$

求导：

$$\nabla P(x_p) = c' * \sum k'_i(\|\frac{(x_p - x_i)^2}{h}\|) * \sum 2 * (x_p - x_i) \quad (5)$$

如果令：

$$g(x) = -k'(x) \quad (6)$$

代入上式得：

$$\begin{aligned} \nabla P(x_p) &= c' * \left\{ \sum g_i(\|\frac{(x_p - x_i)^2}{h}\|) * \sum x_i - \sum g_i(\|\frac{(x_p - x_i)^2}{h}\|) * \sum x_p \right\} \\ &= c' * \sum g_i(\|\frac{(x_p - x_i)^2}{h}\|) * \left[ \frac{x_i * g_i(\|\frac{(x_p - x_i)^2}{h}\|)}{g_i(\|\frac{(x_p - x_i)^2}{h}\|)} - x_p \right] \end{aligned}$$

令上式等于 0，解出：

$$x'_p = \frac{x_i * g_i(\|\frac{(x_p - x_i)^2}{h}\|)}{g_i(\|\frac{(x_p - x_i)^2}{h}\|)} \quad (7)$$

$x'_p$  就是新的中心位置。

可以看到，如果  $g_i(\cdot)$  的值始终为 1（方波情况），上式就蜕化成最简单（符合直觉）的形式：求质心。

### 2.8.1 核密度估计的物理意义：

观察公式发现，相对最简单的“质心”形式，选用高斯核，本质是将样本与原中心的距离这一因素考虑进去。

对于高斯核来讲，这一因素表现为，距离越远，样本的发言权越小。

对于矩形核来讲，这一因素表现为，距离远近对发言权的大小无影响。

回到实际情况，目标的颜色分布一般来讲，更接近矩形分布（颜色均匀）。并不随着距离而衰减，所以，对于胶水瓶盖子，高斯模型并无效果提升。

## 2.9 源码

所有源码在:

[https://github.com/xueyayang/v4l2\\_demo/tree/master/mean-shift](https://github.com/xueyayang/v4l2_demo/tree/master/mean-shift)

## 3 总结

### 3.1 核的选择

对于最简单的颜色模型，选用矩形核，即，求质心的形式足够了。

### 3.2 时间开销

选用高斯核，一次 shift 的时间大约是矩形核的 30 倍左右。开销集中在

- 高斯公式的计算
- 带宽  $h$  的计算
- 方差的计算