

# 关于超像素分割方法的总结

朱亚菲

2015 年 1 月

## 目录

1 引言	1
2 基于图论的超像素分割方法	2
2.1 Graph-based segmentation	2
3 基于梯度上升的超像素分割方法	2
3.1 Mean shift 算法	2
3.1.1 概念	2
3.1.2 原理	4
3.1.3 应用	6
3.2 Turbopixel	6
3.3 SLIC 算法	6
4 实验效果分析	8

## 1. 引言

图像分割是指按照一定的相似性准则将图像划分为具有特殊语义的不同区域，其研究最早可以追溯至 20 世纪 60 年代，已历经几十年的发展，图像分割作为计算机视觉领域的基本问题，是图像理解的重要组成部分。与此同时，它在图像处理、模式识别和人工智能等多个领域也扮演了关键的角色。

目前对图像的处理大多以像素为单位，用二维矩阵来表示一幅图像，并未考虑像素之间的空间组织关系，这使得算法处理效率过低。2003 年，Ren 等人 [10] 最早提出了超像素这一概念，所谓超像素，是指具有相似纹理、颜色、亮度等特征的相邻像素构成的图像块。它利用像素之间特征的相似程度将像素分组，可以获取图像的冗余信息，在很大程度上降低了后续图像处理任务的复杂度。从本质上看，可以将超像素分割看成是聚类问题。

超像素生成算法大致可分为基于图论的方法、基于梯度上升的方法两类 [7]。

## 2. 基于图论的超像素分割方法

基于图论<sup>1</sup>的超像素分割方法将

### 2.1 Graph-based segmentation

Felzenszwalb 等人 [8] <http://cs.brown.edu/pff/segment/>

## 3. 基于梯度上升的超像素分割方法

梯度上升方法是从一个粗糙的像素的初始聚类开始，通过不断的迭代来优化聚类簇，直至满足收敛准则以形成超像素。以下几种方法都采用了聚类的基本思想，但各自具体方法不同，也有不同的优缺点。

### 3.1 Mean shift 算法

#### 3.1.1 概念

1. 看了百度文库上的这篇文章 [1] 后的几点收获：

1) Mean shift 有两种含义，早期刚提出时 Mean shift 是一个名词，指代的是偏移的均值向量。而我们现在所说的 Mean shift 通常是一个动词，指的是一个迭代的步骤，即先算出当前点的偏移均值，移动该点到其偏移均值，然后以此为新的起始点，继续移动，直到满足一定的条件结束。

2) 几个难懂而又关键的词语：Mean shift 向量、核函数、核密度估计、带宽矩阵

2. 经学习后对上述词语的理解：

**核函数**：由网站 [4] 了解到：在机器学习领域，核函数是低维空间到高维空间的一个映射，目的是解决以下两种问题：

1) 分类问题中解决线性不可分的情况

参考网址 [5]，详见图 1

2) 回归问题中为了更好地拟合

<sup>1</sup>图论的产生源于 18 世纪的著名古典数学问题之——哥尼斯堡七桥问题，问题提出后，很多人对此很感兴趣，纷纷进行试验，但在相当长的时间里，始终未能解决。1736 年 29 岁的欧拉向圣彼得堡科学院递交了《哥尼斯堡的七座桥》的论文，在解决问题的同时，开创了数学的一个新的分支——图论与几何拓扑。

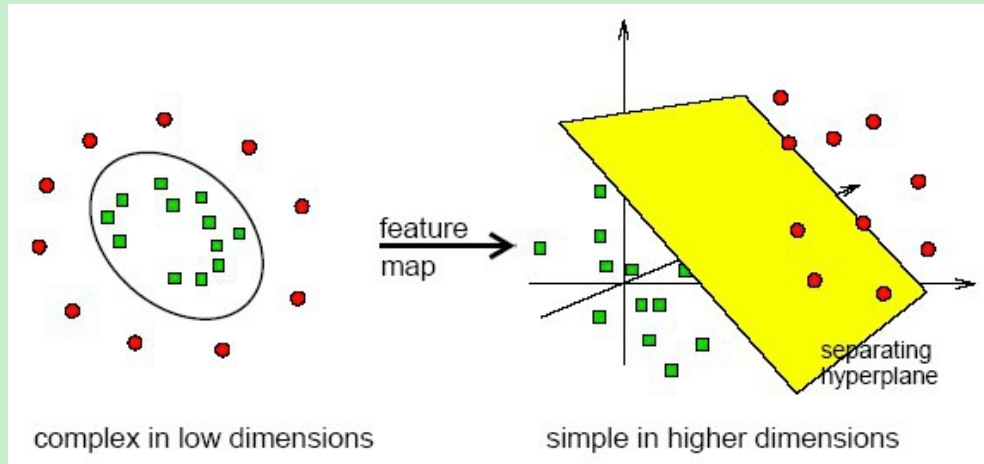


图 1: 将特征映射到高维分类变得简单

例如如下的“线性回归”问题，特征是房子的面积  $x$ ，这里的  $x$  是实数，结果  $y$  是房子的价格。假设我们从样本点的分布中观察到  $x$  和  $y$  符合 3 次曲线，那么我们希望使用  $x$  的三次多项式来逼近这些样本点，首先需要将特征  $x$  扩展到三维  $(x, x^2, x^3)$ ，然后寻找特征和结果之间的模型。

**核密度估计：**由给定样本点集合求解随机变量的分布密度函数问题是概率统计学的基本问题之一。解决这一问题的方法包括参数估计和非参数估计。其中非参数估计又有直方图估计、核密度估计等。

核密度估计方法是由 Rosenblatt[] 和 Parzen[] 提出的，该方法不利用有关数据分布的先验知识，对数据分布不附加任何假定，是一种从数据样本本身出发研究数据分布特征的方法。

核密度估计的目的：设  $x_1, x_2, \dots, x_n$  是从具有未知密度函数  $f(x)$  的总体中抽出的独立同分布样本，要依据这些样本对每一  $x$  去估计  $f(x)$  的值。

核密度估计的公式为：

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (1)$$

其中  $K(\cdot)$  称为**核函数**，满足对称性及  $\int K(x)dx = 1$ 。

$h$  称为带宽，一般  $h$  越大，估计的密度函数就越光滑，但偏差可能较大，选择的原则是使得均方误差最小。在实际中，带宽的选择很大程度上取决于主观判断：如果认为真实的概率密度分布曲线是比较平坦的，那么就选择较大的带宽；相反，如果认为真实的概率分布曲线是比较陡峭的，那么就选择较小的带宽。

核密度估计原理：利用数据点  $x_i$  到  $x$  的距离来决定  $x_i$  在估计点  $x$  的密度时所起的作用，若某个数在观察值中出现了，我们可以认为这个数的概率密度比较大，和这个数比较近的数的概率密度也会

比较大，而那些离这个数远的数的概率密度会比较小。基于这种想法，对点  $x$  来说，若  $x - x_i$  较小，说明  $x$  与观测值  $x_i$  相距较近，因而  $x$  处的概率密度也会比较大，若  $x - x_i$  较大，则对应  $x$  处的概率密度也会比较小，可以用  $\frac{1}{h}K\left(\frac{x - x_i}{h}\right)$  来拟合我们想象中的那个远小近大的概率密度，其中  $K$  函数内部的  $h$  分母用于调整 KDE 曲线的宽幅，而  $K$  函数外部的  $h$  分母则用于保证曲线下方的面积符合 KDE 的规则 (KDE 曲线下方面积和为 1)。针对每一个观察中出现的数拟合出多个概率密度分布函数之后，取平均。

### 3.1.2 原理

1975 年 Fukunage 在论文 [9] 中首先提出 Mean shift 的概念，这篇论文主要是估计样本数据的多元概率密度梯度，基本思想是：某点的概率密度梯度可以用其周围一个很小的区域内的样本观测值来估计。对样本中的每一点，需要估计出其周围  $k$  个点的梯度向量， $k$  是固定的，这些梯度向量就叫做“ $k$  最近邻 Mean shift 簇”，对这些 Mean shift 向量求加权平均则得到该点的密度梯度。

设  $X_1, X_2, \dots, X_N$  是一组由  $N$  个独立同分布的  $n$  维随机向量组成的集合。在  $X$  点处密度梯度表示如下：

$$\hat{\nabla}_x p_N(X) = (Nh^{n+2})^{-1} 2c \sum_{X_i \in S_h(X)} (X_i - X) = \left( \frac{k}{Nv_h(X)} \right) \frac{n+2}{h^2} \left( \frac{1}{k} \sum_{X_i \in S_h(X)} (X_i - X) \right) \quad (2)$$

其中，

$$v_h(X) \equiv \int_{S_h(X)} dY = \frac{h^n \pi^{n/2}}{\Gamma(n+2/2)} \quad (3)$$

$$S_h(X) \equiv \{Y : (Y - X)^T(Y - X) \leq h^2\} \quad (4)$$

Mean shift 向量就是从密度函数梯度的非参数估计中推导获得的，取 (2) 式的最后一项就是就是 Mean shift 向量

$$M_h(X) \equiv \frac{1}{k} \sum_{X_i \in S_h(X)} (X_i - X) \quad (5)$$

显然，如果梯度为零，对应  $S_h(X)$  区域内密度较均匀， $X$  处的偏移的均值向量也将为零；如果  $X$  处的密度梯度是一个指向概率密度函数增长最快的方向的非零梯度，那么从平均意义上说， $X$  周围更

多的观测点都将沿其方向存在，相应地，偏移的均值向量 Mean shift 也应该指向该方向，并且其长度与梯度大小成比例。

举个例子，见 [3]

在  $d$  维空间中，任选一个点，然后以这个点为圆心， $h$  为半径作一个高维球。落在这个球内的所有点和圆心都会产生一个向量，向量是以圆心为起点落在球内的点位终点，然后把这所有向量都相加，相加的结果就是 Mean shift 向量。如图2所示，其中黄色箭头就是 Mean shift 向量。

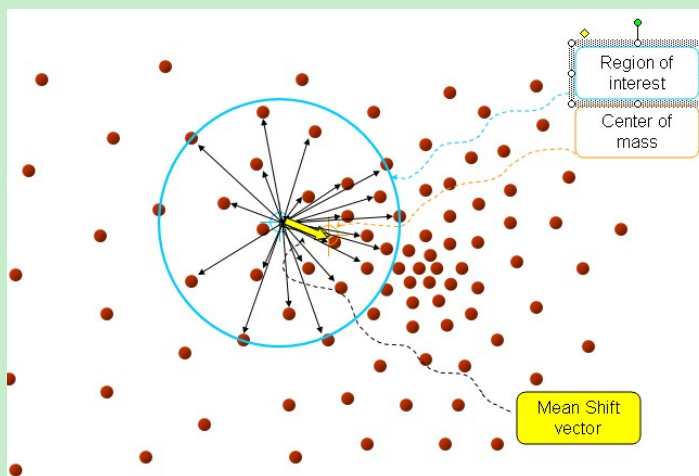


图 2: 初始状态

再以 Mean shift 向量的终点为圆心，再做一个高维的球。如图3所示，重复以上步骤，就可得到一个 Mean shift 向量，如此重复下去，可以收敛到概率密度最大的地方，也就是最稠密的地方。

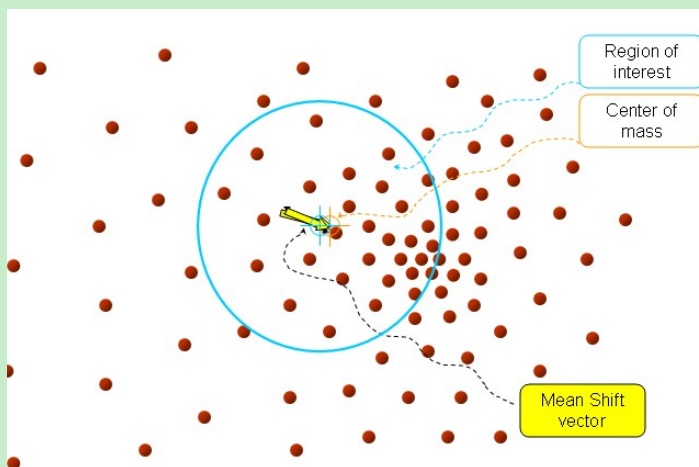


图 3: 迭代过程

最终结果如图4:

怎么确定  $X$  周围  $k$  个数据点所在的区域?

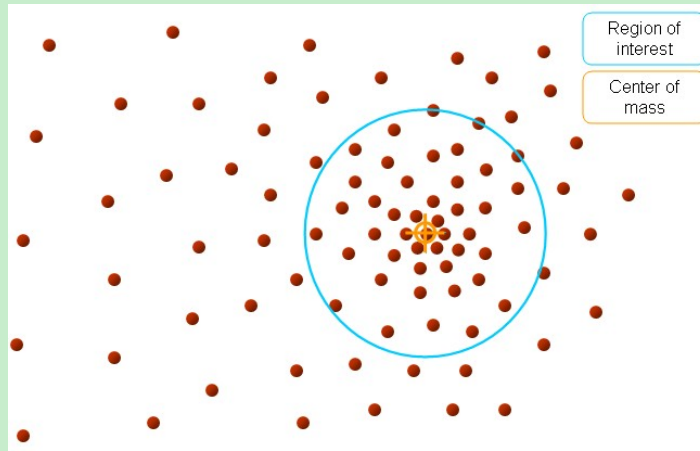


图 4: 最终状态

### 3.1.3 应用

#### 1. 聚类

Mean shift 算法是基于概率密度的方法，不需要任何先验条件，数据集中的每一点都可以作为初始点，分别执行 Mean shift 算法，收敛到同一个点算作一类，它可对任何维度、任何分布的采样点进行聚类。

#### 2. 图像分割

Mean shift 算法是一种统计聚类方法，在图像分割时，首先将图像像素转换成特征空间的采样点，例如对一幅彩色图像，考虑到图像的空间信息和色彩信息，特征空间可由 2 维的位置空间和 3 维的色度空间组成，图像像素转换成特征空间中的一个 5 维采样点，然后对采样点进行 Mean shift 聚类，特征空间中的聚类对应于图像空间的分割。

#### 3. 目标跟踪

利用 Mean shift 算法进行物体跟踪的实质是求解最优化的 Bhattacharyya 系数函数 [?]。该函数表示的是目标对象和候选对象的相似度，通过泰勒展开后可转化为隐含估计的概率密度函数，因此用 Mean shift 算法求解。

## 3.2 Turbopixel

## 3.3 SLIC 算法

SLIC 是 Simple Linear Iterative Clustering 的缩写，最早由 Achanta [6] 于 2010 年 6 月在洛桑联邦理工学院的一次技术报告中提出。2012 年经整理、完善后发表在 TPAMI 上 [7]。其中“Linear”体现在算法复杂度  $O(N)$  与图像中像素个数  $N$  是线性关系，“Clustering”是采用 k-均值聚类。



SLIC 是一种使用简单和易于理解的方法，是对传统的  $k$ -均值超像素分割方法的改进。两者之间主要有以下两点区别：

1. SLIC 优化了距离计算的数目，传统的  $k$ -均值方法中每个聚类中心的搜索范围是整幅图像，要计算该聚类中心与图像中所有像素点之间的距离，而 SLIC 算法只在与预设的超像素尺寸成比例的一个区域内搜索，见图5。

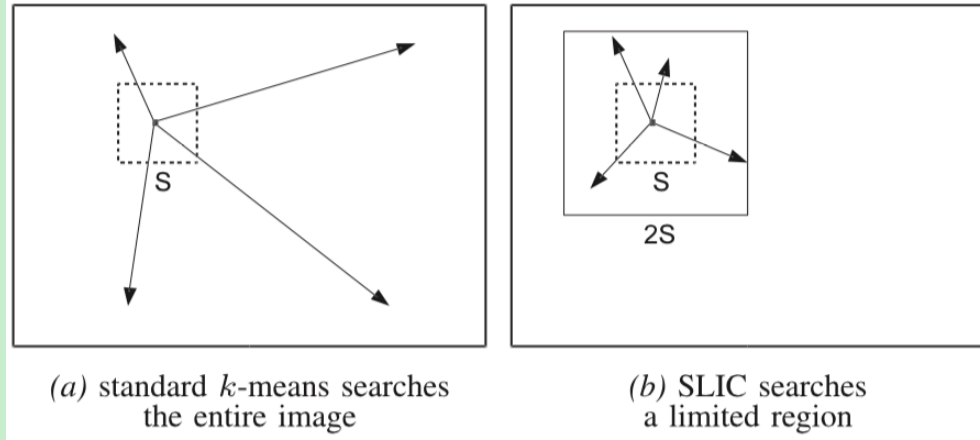


图 5: 传统的  $k$ -均值方法与 SLIC 算法搜索区域的区别

2. SLIC 算法在计算聚类中心与像素点之间的距离时同时结合了颜色和空间位置两方面的信息来表征其相似度，这样可以起到控制超像素尺寸和紧密度的作用。

默认情况下，该算法只有一个参数  $k$ ，即所需生成的近似同样大小的超像素的个数。具体步骤为：

a) 初始化聚类中心。假设一幅彩色图像有  $N$  个像素点，预分割为  $k$  个相同尺寸的超像素，那么首先要将该彩色图像转换为 CIELAB 颜色空间和 XY 坐标下的 5 维特征向量，然后将图像均分为  $k$  个网格，网格边长为  $S = \sqrt{N/k}$ ， $k$  个初始聚类中心  $C_i = [l_i, a_i, b_i, x_i, y_i]^T$  为每个网格内色阶值最低的像素。对图像中的每一个像素点，令其与最近的聚类中心的距离  $d(i)$  的初始值为  $\infty$ 。

b) 相似度衡量。对于每个聚类中心，逐一计算其  $2S \times 2S$  邻域内（如图5）各像素  $i$  与该类中心点的距离  $D$ ，若  $D < d(i)$ ，则  $i$  暂时归为该类的，并且将  $d(i)$  赋值为  $D$ ，其中

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \quad (6)$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (7)$$

$$D = \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)^2 m^2} \quad (8)$$

c) 确定新的聚类中心。当每个像素都被归类到其最近的聚类中心后，新的聚类中心为该类中所有像素的  $[l, a, b, x, y]^T$  向量的平均值。为了避免聚类中心落在边界上，以及对后续的聚类过程造成干扰，

需要将聚类中心在以  $3 \times 3$  的邻域内打乱，将聚类中心移到邻域内梯度最小的地方（图像灰度值变化最缓慢的地方）。计算残差  $E$ （新的聚类中心与之前聚类中心的  $L1$  范数）。

d) 重复 b) 和 c) 直至最后收敛 ( $E \leq threshold$ )。

## 4. 实验效果分析

### 参考文献

- [1] Mean shift 概述. <http://wenku.baidu.com/view/2307ca1cfad6195f312ba6f7.html>.
- [2] meanshift, 聚类算法. <http://blog.sciencenet.cn/blog-261330-706985.html>.
- [3] 核函数. <http://www.cnblogs.com/jerrylead/archive/2011/03/18/1988406.html>.
- [4] 核函数的作用. [http://www.360doc.com/content/14/0728/15/14106735\\_397653989.shtml](http://www.360doc.com/content/14/0728/15/14106735_397653989.shtml).
- [5] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. Technical report, 2010.
- [6] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012.
- [7] Mark Fashing and Carlo Tomasi. Mean shift is a bound optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):471–474, 2005.
- [8] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.
- [9] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, 1975.
- [10] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 10–17. IEEE, 2003.