

文章编号: 1001-0920(2011)08-1209-05

基于广义松弛方法的快速均值偏移算法的研究

文志强, 朱艳辉, 李长云, 邓晓军

(湖南工业大学 计算机与通信学院, 湖南 株洲 412008)

摘要: 针对均值偏移算法收敛性慢的缺点, 通过对加速均值偏移方法的研究, 将优化算法中的松弛方法扩展为广义的松弛方法, 并给出确定对角矩阵参数的方法; 然后对收敛准则进行了修改. 在此基础上, 提出了基于广义松弛方法的均值偏移算法, 并通过大量的聚类实验和图像分割实验验证了所提出算法的有效性.

关键词: 广义松弛方法; 快速均值偏移; 聚类分析; 图像分割

中图分类号: TP391.4

文献标识码: A

Study on fast mean shift algorithm based on generalized overrelaxed methods

WEN Zhi-qiang, ZHU Yan-hui, LI Chang-yun, DENG Xiao-jun

(School of Computer and Communication, Hunan University of Technology, Zhuzhou 412008, China.

Correspondent: WEN Zhi-qiang, E-mail: zhqwen20001@163.com)

Abstract: For the shortcomings of slow convergence of mean shift algorithm, based on the study of accelerated mean shift method, the overrelaxed methods coming from optimization algorithm are extended to generalized overrelaxed methods, and the method for determining the parameters in the diagonal matrix is given. Then the convergence criteria is amended. For the above analysis, mean shift algorithm based on generalized overrelaxed methods is proposed. A large number of experiments on cluster analysis and image segmentation show the effectiveness of the proposed algorithm.

Key words: generalized overrelaxed method; fast mean shift; cluster analysis; image segmentation

1 引言

均值偏移(MS)算法是由 Fukunaga 和 Hostetler^[1]于1975年提出的, 直到1995年 Cheng^[2]的研究成果出现, 才引起了人们的兴趣, 从此掀起了一股研究和应用MS算法的热潮. 作为一种有效的统计迭代算法, 近年来, MS算法已广泛应用于计算机视觉与模式识别等领域, 例如目标跟踪、图像分割、模式识别与聚类分析、滤波以及信息融合等方面.

目前, 普遍认为MS算法的收敛速度较慢, 不利于进行实时处理, 因此MS算法的实时性研究是一个急需解决的问题. 就实时性的提高而言, 改善MS收敛速度是一个非常有效的方法. 如文献[3]使用类Newton方法提高MS的收敛速度, 但该方法使用梯度信息近似Hessian矩阵, 需要搜索步长, 增加开销. [4]提出了基于共轭梯度的快速均值偏移算法, 具有较快的收敛速度. [5]提出一种自适应带宽的动态

MS方法, 引入动态更新机制, 提高了收敛速度. [6]研究了4种高斯MS策略: 空域离散策略, 空间邻域策略, sparse EM策略和EM-Newton策略, 以提高MS算法的效率. [7-8]提出了一种基于松弛方法的快速MS方法, 在迭代过程中, 引入一个学习速率因子, 当满足递增条件时, 认为此次迭代有效, 且增大了学习速率因子; 否则, 认为此次迭代无效, 修改学习速率因子为1. 这种方法简单, 不会增加算法的时间复杂度.

另一种改善MS算法实时性能的方法是降低MS时间复杂度, 如改进MS开销较大的操作效率(例如搜索数据点的最近邻域), 或采用近似技术 locality-sensitive hashing (LSH) 来减少自适应MS的计算复杂性^[9]. 降低特征空间分析的复杂度也是改善MS算法实时性能的方法, 如文献[10]提出一种快速MS(FMS)算法, 在描述密度分布时将每个局部样本视为一个整体, 每个样本集中的样本均假设来自同一类别,

收稿日期: 2010-04-28; 修回日期: 2010-08-10.

基金项目: 国家自然科学基金项目(60773110); 湖南省自然科学基金项目(09JJ6087); 湖南省教育厅科研项目(10C0621).

作者简介: 文志强(1973—), 男, 副教授, 博士, 从事图像处理、目标检测与跟踪等研究; 朱艳辉(1968—), 女, 教授, 从事文本分类、机器学习等研究.

只需为每个样本集标记一个样本中心,以改善MS算法的实时性.其他加速的方法多集中于函数展开技术,如快速的高斯变换应用于相似区域颜色分布的高效模型^[11],改进的高斯变换加速MS矢量计算^[12]等.

本文针对MS算法收敛慢的问题,对松弛方法进行扩展,提出一种基于广义松弛方法的快速MS算法.实验结果表明了该方法的有效性.

2 MS算法简介

密度估计(又称Parzen窗估计)是一种很流行的密度估计方法.给定在 d 维空间 R^d 的 n 个样本数据 $\mathbf{x}_i, i = 1, 2, \dots, n$,多维变量的核密度估计可写成

$$f_{\mathbf{H},k}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n |\mathbf{H}|^{-1/2} k\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{\mathbf{H}}\right\|^2\right).$$

其中: $k(x)$ 为核函数且满足下列条件:1) $k(x)$ 为非负;2) $k(x)$ 非单调递增,即如果 $a < b$,则 $k(a) \geq k(b)$;3) $k(x)$ 为连续函数且有界,即

$$\int_0^{\infty} k(t)dt < \infty.$$

\mathbf{H} 为 $d \times d$ 的带宽矩阵.用完整的参数表示 \mathbf{H} 会增加估计的复杂性.实际中, \mathbf{H} 可以为对角矩阵 $\mathbf{H} = \text{diag}[h_1^2, \dots, h_d^2]$ 或为 $h^2 \mathbf{I}$,这里 \mathbf{I} 为 $d \times d$ 单位矩阵.为简单起见,本文使用后一种 \mathbf{H} ,则归一化后的核密度估计为

$$f_{h,k}(\mathbf{x}) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^n k\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right), \quad (1)$$

其中 $c_{k,d}$ 为归一化常数.当 $k(x)(x > 0)$ 可微时,对式(1)求梯度 Δ ,然后令 $\Delta = 0$,便可获得MS迭代公式为

$$\mathbf{y}_{t+1} = m(\mathbf{y}_t) = \frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{y}_t - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{y}_t - \mathbf{x}_i}{h}\right\|^2\right)}, \quad (2)$$

其中 $g(x) = -k'(x)$.MS迭代公式沿梯度方向迭代,使每个待处理的点“漂移”到分布密度函数的局部极大值点处,其步长随迭代过程自适应地变化.在低密度区,迭代步长较长;在局部极大值附近,迭代步长较短.

MS算法是严格单调递增收敛的.设 \mathbf{y}_* 为MS算法的收敛点,则有 $\mathbf{y}_* = m(\mathbf{y}_*)$.设 λ_{\max} 为矩阵 $\Delta m(\mathbf{y}_*)$ 的最大特征值,若 λ_{\max} 越接近于1,则MS算法的收敛速度越慢.下面给出MS算法.

算法1 MS算法

Step 1: 初始化 \mathbf{y}_0 ,结束条件 ε 为小数, $t = 0$.

Step 2: 用式(2)计算 \mathbf{y}_{t+1} 的值.

Step 3: 判断是否满足 $\|\mathbf{y}_{t+1} - \mathbf{y}_t\| \leq \varepsilon$.如果满足,则结束;否则,用 \mathbf{y}_{t+1} 替代 \mathbf{y}_t 并转Step 2.

3 快速MS算法

3.1 广义超松弛方法

对文献[8]使用的松弛方法进行扩展,在松弛方法中引入一个矩阵 \mathbf{B} ,则广义超松弛方法如下:

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \mathbf{B}(\bar{\mathbf{y}}_{t+1} - \mathbf{y}_t), \quad (3)$$

其中

$$\bar{\mathbf{y}}_{t+1} = m(\mathbf{y}_t) = \frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{y}_t - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{y}_t - \mathbf{x}_i}{h}\right\|^2\right)}, \quad (4)$$

并记为MS(\mathbf{B}).当 $\mathbf{B} = \mathbf{I}$ 时,即为经典MS算法^[13],记为MS(\mathbf{I});当 $\mathbf{B} = \beta \mathbf{I}$ 时,则为经典的超松弛方法,文献[7-8]使用该经典超松弛方法于MS中以减少迭代次数.本文主要研究 \mathbf{B} 为稍复杂矩阵的情形.设 $\Phi(\mathbf{y}_t) = \mathbf{y}_t + \mathbf{B}(m(\mathbf{y}_t) - \mathbf{y}_t)$,则有 $\mathbf{y}_{t+1} = \Phi(\mathbf{y}_t)$.如果 $m(\mathbf{y})$ 在收敛点 \mathbf{y}_* 附近可微,则有

$$\Delta \Phi(\mathbf{y}_t) = \mathbf{I} - \mathbf{B}(\mathbf{I} - \Delta m(\mathbf{y}_t)).$$

当 \mathbf{y}_t 在收敛点 \mathbf{y}_* 的邻域内且 $\Delta m(\mathbf{y}_t)$ 的特征值在1附近时, $\mathbf{B} = \mathbf{I}$ 时的MS算法收敛速度非常慢,这时应增大矩阵 \mathbf{B} 的特征值,以加快MS算法的收敛速度.为简便起见,本文仅讨论一种简单情形: \mathbf{B} 为对角矩阵,并设对角矩阵参数分别为 b_1, b_2, \dots, b_d .下面对MS(\mathbf{B})算法的收敛性进行分析.

对 $\mathbf{y} = \Phi(\mathbf{y})$ 在 \mathbf{y}_* 处使用Taylor公式展开,得

$$\Phi(\mathbf{y}_t) = \Phi(\mathbf{y}_*) + \Delta \Phi(\mathbf{y}_*)(\mathbf{y}_t - \mathbf{y}_*) + \dots$$

如果 \mathbf{y}_t 在 \mathbf{y}_* 的邻域内,则有

$$\mathbf{y}_{t+1} - \mathbf{y}_* \approx \Delta \Phi(\mathbf{y}_*)(\mathbf{y}_t - \mathbf{y}_*).$$

于是

$$\|\mathbf{y}_{t+1} - \mathbf{y}_*\| \approx \|\Delta \Phi(\mathbf{y}_*)(\mathbf{y}_t - \mathbf{y}_*)\| \leq \rho \|\mathbf{y}_t - \mathbf{y}_*\|,$$

其中 ρ 为最大的 $\Delta \Phi(\mathbf{y}_*)$ 谱半径,即

$$\rho = \max\{|1 - b_1(1 - \lambda_1)|, \dots, |1 - b_n(1 - \lambda_d)|\}.$$

这里 $\lambda_1, \lambda_2, \dots, \lambda_d$ 为 $\Delta \Phi(\mathbf{y}_*)$ 的特征值.

为了保证MS(\mathbf{B})算法收敛,需满足

$$\rho = \max\{|1 - b_1(1 - \lambda_1)|, \dots, |1 - b_n(1 - \lambda_d)|\} < 1, \quad \text{即}$$

$$|1 - b_j(1 - \lambda_j)| < 1, 1 \leq j \leq d.$$

于是得 $0 < b_j < 2$.另外,在 \mathbf{y}_* 的邻域内, t 足够大且 $0 < b_j < 2$ 时,有

$$r =$$

$$\frac{\|\mathbf{y}_{t+1} - \mathbf{y}_*\|}{\|\mathbf{y}_t - \mathbf{y}_*\|} = \frac{\|\Delta \Phi(\mathbf{y}_t)(\mathbf{y}_t - \mathbf{y}_*)\|}{\|\mathbf{y}_t - \mathbf{y}_*\|} \leq \frac{\rho \|\mathbf{y}_t - \mathbf{y}_*\|}{\|\mathbf{y}_t - \mathbf{y}_*\|},$$

则 $r = \rho < 1$.因此,当 $0 < b_j < 2 (1 \leq j \leq d)$ 时,对于 \mathbf{y} 在 \mathbf{y}_* 的邻域内,MS(\mathbf{B})算法以线性速度收敛到核密度函数 $f_{h,k}(\mathbf{y})$ 函数的局部最优值.

3.2 迭代停止准则

对于实际应用而言, 初始 y_0 并不在 y_* 附近, 因此 MS 算法并不一定收敛. 设定一个小数 ε , 当 $\|y_{t+1} - y_t\| < \varepsilon$ 时, 停止迭代. 当算法不收敛时, 一般的处理方式是设定一个最大迭代次数, 当超过迭代次数时结束当前迭代. 这种方法非常费时.

通过分析 MS 算法迭代过程可以发现, 数值 $d_t = \|y_{t+1} - y_t\|$ 的变化是曲线, 如图 1 所示.

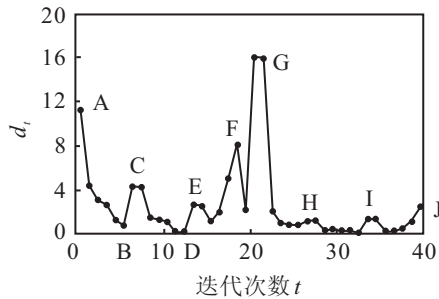


图 1 迭代收敛时 d_t 值随 t 变化过程

在图 1 中, 从 A 点开始, d_t 值随迭代次数逐渐减小, 且趋近于点 B; 然后又增大至 C 点, 直到迭代次数达到规定的次数, 整个过程是不收敛的 (这里 ε 取 0.1, 最大迭代次数设为 40). 从整个迭代过程看, 从 A 点开始迭代到 B 点再到 C 点, 其实 MS 算法在 B 点迭代时便已收敛到局部极值点; 然后又从 C 点开始迭代直到 D 点. 实质上, MS 迭代过程是从一个局部极值点跳跃到附近的另一个极值点, 这样一直跳跃下去. 因此这种迭代只会增加迭代次数和计算时间, 甚至会造成误差. 为了避免这种耗时的迭代, 重新设计了算法的停止准则如下:

```

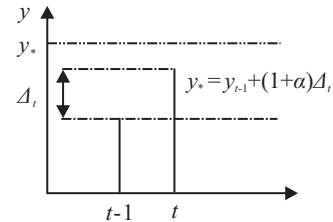
if (t = 0 && d_t < ε), 则停止迭代;
else {
    if ((d_t - d_{t-1} > 0) || (d_t < ε) || (t = N)), 则停止迭代;
}
    
```

其中: t 为迭代次数, N 为最大迭代次数. 通过该规则能够避免局部极值收敛跳跃的情形, 大大减少迭代次数.

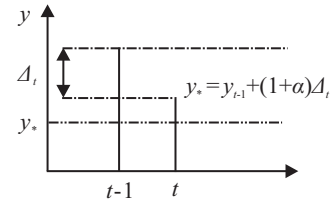
3.3 对角矩阵参数修改的方法

在 MS 算法的迭代过程中会产生序列 y_t, y_t 的第 j 个分量为 y_{tj} , 可简单表示为 y_t . y_* 表示第 j 个分量收敛的值, 而对于固定第 j 个分量的 y_t 相当于一个时间序列, 这种时间序列具有 3 种形式, 如图 2 所示.

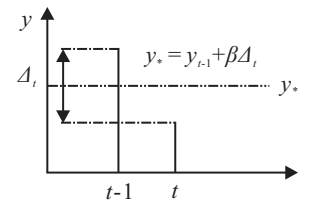
图 2(a) 和图 2(b) 表示序列递增或递减迭代过程, 称为平稳收敛; 图 2(c) 表示在收敛点附近跳动, 称为震荡收敛, 其中 y_* 为预测的收敛点. 对于所有的收敛数列 y_t , 或具有震荡收敛, 或是这两者的结合. 对于平稳收敛数列 y_t 而言, 存在 t , 使得 $\Delta_t \cdot \Delta_{t-1} \geq 0$



(a) 平稳收敛数列 (序列递增)



(b) 平稳收敛数列 (序列递减)



(c) 震荡收敛数列

图 2 时间序列收敛方式示意图

($\Delta_t = y_t - y_{t-1}$), 且满足 $y_* = y_{t-1} + (1 + \alpha) \Delta_t$. 其中 α 为预设常数 ($\alpha < 1$), 如图 2(a), 图 2(b) 所示. 对于震荡收敛数列 y_t 而言, 存在 t , 使得 $\Delta_t \cdot \Delta_{t-1} < 0$ 且满足 $y_* = y_{t-1} + \beta \Delta_t$, 其中 β 为预设常数 ($\beta < 1$), 如图 2(c) 所示. 因此目标权值预测方法如下:

```

if (Delta_t * Delta_{t-1} >= 0), y_* = y_{t-1} + (1 + alpha)Delta_t;
else y_* = y_{t-1} + beta Delta_t.
    
```

通过上述预测方法, 不断地预测收敛点, 最终收敛到最终目标点. 根据上述思想, 给出修改对角矩阵参数方法如下: 给定初始参数 $a_{\max}, a_{\text{mid}}, a_{\min}$ 的值, 在实验中分别取 1.7, 1.5, 和 1.3.

```

if (t = 0)
    for (j = 0; j < d; j++)
        b_j = b_j * a_mid;
    else for (j = 0; j < d; j++)
        if (Delta_{tj} * Delta_{t-1j} >= 0), b_j = b_j * a_max;
        else b_j = b_j * a_min.
    
```

3.4 快速 MS 算法

根据前面的分析, 下面给出一个快速 MS (FMS) 算法.

算法 2 快速 MS 算法

Step 1: 初始化参数. 初始值 y_0 , 迭代次数 $t = 0$, ε 为小数, B 的对角参数 b_j 为 1 ($1 \leq j \leq d$).

Step 2: 利用式 (4) 计算矢量 \bar{y}_{t+1} .

Step 3: 利用式 (3) 进行修正, 得 \mathbf{y}_{t+1} .

Step 4: 如果 $f_{h,k}(\mathbf{y}_{t+1}) > f_{h,k}(\mathbf{y}_t)$, 则用 3.3 节的方法修改 \mathbf{B} 矩阵对角参数 b_j ; 否则 $\mathbf{y}_{t+1} = \bar{\mathbf{y}}_{t+1}$, 且 \mathbf{B} 矩阵的对角参数 b_j 均为 1.

Step 5: 使用 3.2 节的停止准则判断是否还需要迭代. 如果需要, 则 $t = t + 1$, 转 Step 2; 否则, 算法结束.

4 实验结果及分析

为了验证算法的有效性, 从 MS 的数据聚类 and 图像分割应用两个方面进行实验. 实验中 $\varepsilon = 0.01$, 最大迭代次数 $N = 40$.

4.1 数据聚类实验

将数据集中的每个数据点分别作为初始点, 使用 MS 算法获取其收敛点, 于是会自动形成许多聚类中心. 为了验证所提出的快速 MS 算法的有效性, 在聚类实验中使用 3 个合成数据集, 每个数据集各包含 150 个数据. 数据集及相关参数如下:

set #1: 150 个一维数据点都是均等机率从 3 个正态分布中抽取, 3 个正态分布分别是 $N(2.9, 1)$, $N(0.8, 1.0)$, $N(-2, 1.0)$.

set #2: 150 个二维数据点都是均等机率从 3 个正态分布中抽取, 3 个正态分布分别是

$$N\left(\begin{bmatrix} 2.0 \\ 2.2 \end{bmatrix}, \begin{bmatrix} 2.0 & 0 \\ 0 & 2.1 \end{bmatrix}\right), N\left(\begin{bmatrix} 4.2 \\ 8.2 \end{bmatrix}, \begin{bmatrix} 1.3 & 0 \\ 0 & 2.0 \end{bmatrix}\right), \\ N\left(\begin{bmatrix} 7.8 \\ 3.0 \end{bmatrix}, \begin{bmatrix} 2.0 & 0 \\ 0 & 1.9 \end{bmatrix}\right).$$

set #3: 150 个二维数据点都是均等机率从 3 个正态分布中抽取, 3 个正态分布分别是

$$N\left(\begin{bmatrix} -7 \\ 9.8 \end{bmatrix}, \begin{bmatrix} 5.6 & 2 \\ 1 & 5.3 \end{bmatrix}\right), N\left(\begin{bmatrix} 0.45 \\ 11.7 \end{bmatrix}, \begin{bmatrix} 8.4 & 0.9 \\ 0.5 & 9.8 \end{bmatrix}\right), \\ N\left(\begin{bmatrix} 11.8 \\ 14.9 \end{bmatrix}, \begin{bmatrix} 14.5 & 2.0 \\ 2.1 & 3.7 \end{bmatrix}\right).$$

对每个数据集中的数据分别使用标准的 MS 算法、自适应松弛 MS (AMS) 算法^[8]和本文提出的 FMS 算法进行聚类实验. 实验中核函数 k 采用高斯函数, 参数 h 设为 2. 实验结果显示, 使用 3 种算法都能获得 3 个聚类中心, 而且 3 种算法的收敛轨迹没有很明显的差别, 但其迭代次数和时间开销却有所不同. 平均迭代次数和平均时间开销的比较如表 1 所示. 其中: d 为平均迭代次数, t 为数据聚类时的平均时间开销.

另外对 set #2 中的 80 个数据分别使用 AMS 和 FMS 算法进行聚类实验的迭代次数比较, 如图 3 所示.

表 1 算法聚类的平均迭代次数和平均时间开销

数据集	FMS		AMS		MS	
	d	t/ms	d	t/ms	d	t/ms
set #1	6.32	32	6.32	38	15.4	46
set #2	3.51	47	4.41	54	7.73	59
set #3	4.47	64	7.21	74	15.2	78

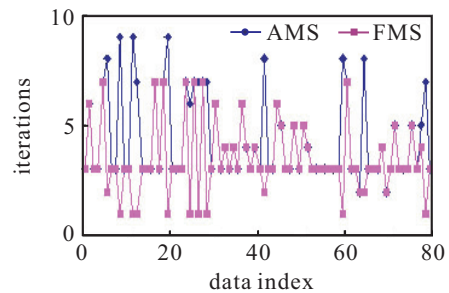


图 3 数据聚类实验中迭代次数的比较

从表 1 和图 3 结果可以看出, 当应用于数据聚类分析时, 相对于其他两种算法而言, 本文提出的 FMS 算法具有较少的迭代次数, 因而减少了算法的时间开销.

4.2 图像分割实验

为了进一步验证算法的有效性, 下面使用 MS, AMS 和 FMS 算法分别用于图像分割实验. 核函数采用截断高斯函数, 即

$$k(x) = \begin{cases} e^{-x}, & |x| \leq \lambda; \\ 0, & |x| > \lambda. \end{cases}$$

其中 λ 取 10. 实验使用的原始图像来自文献 [13], 每个像素表示为 $\mathbf{x} = (r, g, b)$, $0 \leq r, g, b \leq 255$. 对每个像素分别使用 MS, AMS 和 FMS 算法, 使其漂移到各自的局部极大值处. 实验结果显示均能形成分割图像, 但在进行图像分割时, 3 种算法各自的平均迭代次数有所不同, 因而其耗费的计算时间也不相同. 其平均迭代次数和时间开销的比较如表 2 所示. 其中: d 为平均迭代次数, t 为对单个像素进行聚类的平均时间开销.

表 2 图像分割的平均迭代次数和时间开销

编号	图像	FMS		AMS		MS	
		d	t/ms	d	t/ms	d	t/ms
1		6.32	32	6.32	38	15.4	46
2		3.51	47	4.41	54	7.73	59
3		4.47	64	7.21	74	15.2	78

图 4 描述了表 2 中编号为 3 的图像分割中部分像

素分别使用MS, AMS和FMS算法进行分割的迭代次数比较。

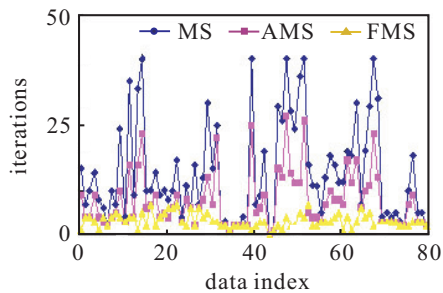


图4 图像分割实验中迭代次数的比较

从表2和图4的结果可以看出, 本文的FMS算法应用于图像分割时, 比其他两种算法具有更少的迭代次数, 从而减少了算法的时间开销。

5 结 论

将MS算法应用于实际中, 其实时性要求是必然的. 本文针对MS算法的加速方法进行了研究, 将松弛方法扩展为广义松弛方法, 并对相关参数的确定提出了可行的方法; 然后对算法迭代停止准则进行了修改; 最后给出了完整的基于广义松弛方法的MS算法. 该算法简单实用, 没有增加算法的时间复杂度, 减少迭代次数明显, 提高了计算效率. 大量基于MS算法的聚类分析和图像分割实验验证了本文算法的有效性.

虽然本文对MS算法进行了改进, 取得了减少迭代次数的效果, 但MS算法在相关领域的应用还需要深入研究. 比如MS目标跟踪的有效性, 背景像素对MS跟踪精度的影响和改善等.

参考文献(References)

- [1] Fukunaga K, Hostetler L D. The estimation of the gradient of a density function, with applications in pattern recognition[J]. IEEE Trans on Information Theory, 1975, 21(1): 32-40.
- [2] Cheng Y Z. Mean shift, mode seeking, and clustering[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 1995, 17(8): 790-799.
- [3] Yang C J, Duraiswami R, De Menthon D, et al. Mean-shift analysis using Quasi-Newton methods[C]. Proc of IEEE Int Conf on Image Processing. Barcelona, 2003, 3: 447-450.
- [4] 李艳灵, 沈轶. 基于共轭梯度法的快速Mean Shift图像分割[J]. 光电工程, 2009, 36(8): 94-99.
(Li Y L, Shen Y. Fast mean shift for image segmentation based on conjugate gradient[J]. Opto-Electronic Engineering, 2009, 36(8): 94-99.)
- [5] 周芳芳, 樊晓平, 叶榛. 基于自适应带宽的快速动态高斯核均值漂移算法[J]. 控制理论与应用, 2008, 25(4): 608-612.
(Zhou F F, Fan X P, Ye Z. Fast dynamic Gaussian mean-shift algorithm based on adaptive bandwidth[J]. Control Theory & Applications, 2008, 25(4): 608-612.)
- [6] Carreira-Perpinón M Á. Acceleration strategies for Gaussian mean shift image segmentation[C]. Proc of IEEE Conf on Computer Vision and Pattern Recognition. New York, 2006, 1: 1160-1167.
- [7] Shen C, Brooks M J, Hengel A. Fast global kernel density mode seeking with application to localisation and tracking[C]. Proc of the 10th IEEE Int Conf on Computer Vision. Beijing, 2005, 2: 1516-1523.
- [8] Shen C, Brooks M J. Adaptive over-relaxed mean shift[C]. Proc of the 8th Int Symposium on Signal Processing and Its Applications. Sydney, 2005, 2: 575-578.
- [9] Georgescu B, Shimshoni I, Meer P. Mean shift based clustering in high dimensions: A texture classification example[C]. Proc of ICCV. Berlin, 2003, 1: 456-463.
- [10] Zhang K, Tang M, Kwok J T. Applying neighborhood consistency for fast clustering and kernel density estimation[C]. Proc of 2005 IEEE Comp Society Conf on Computer Vision and Pattern Recognition. Washington DC, 2005, 2: 1001-1007.
- [11] Elgammal A, Duraiswami R, Davis L. Efficient nonparametric adaptive color modeling using fast Gauss transform[C]. Proc of Int Conf on Computer Vision and Pattern Recognition. Hawaii, 2001, 2: 563-570.
- [12] Yang C, Duraiswami R, Gumerov N, et al. Improved fast Gauss transform and efficient kernel density estimation[C]. Proc of the 9th IEEE Int Conf on Computer Vision. Berlin, 2003, 1: 664-671.
- [13] Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2002, 24(5): 603-619.