

# 基于混沌粒子群的模糊 C-均值聚类算法

张春娜, 李轶然

(辽宁科技大学 软件学院, 辽宁 鞍山 114051)

**摘要:** 为了解决模糊 C-均值 (FCM) 聚类算法的固有缺陷, 提出基于混沌粒子群的模糊 C-均值聚类算法 (CPSO-FCM)。针对 FCM 对聚类初始值的敏感度问题, 辅以粒子群算法以避免随机选取的聚类数和聚类中心所导致的结果不一致。通过引入混沌序列, 在粒子的位置和速度上与原有粒子群优化算法所得计算值加以比较, 取优者。这样不仅能够提高算法全局搜索能力, 也可有助于粒子跳出局部最优。同时定义加速因子与逃逸算子对粒子移动速度加以优化, 以加速收敛。实验结果表明, CSPO-FCM 算法稳定性强, 收敛速度快, 且聚类的准确率高, 效果较好。

**关键词:** 聚类; 粒子群; 混沌序列; 模糊 C-均值; CPSO-FCM 算法

**中图分类号:** TP18 **文献标识码:** A **文章编号:** 1000-7024 (2013) 03-1039-05

## Research of fuzzy C-means clustering algorithm based on chaos particle swarm

ZHANG Chun-na, LI Yi-ran

(School of Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China)

**Abstract:** In order to solve inherent defect of the fuzzy C-means clustering algorithm (FCM), this paper presents a fuzzy C-means clustering algorithm (CPSO-FCM) which based on chaos particle swarm. Firstly, according to the FCM on cluster initial value sensitivity problem, supplemented by particle swarm algorithm to avoid inconsistent results caused by the random selection of cluster number and cluster center. Moreover, by introducing a chaotic sequence, the particle position and velocity with the original particle swarm optimization algorithm the calculation values can be compared, and the best. In this way, not only can improve the algorithm global search ability, but also contribute to the escape of local optimal solution of particle. At the same time, because of definition of acceleration factor and escape operator on particle movement speed optimization, the convergence can be accelerated. Experiments prove that CSPO-FCM algorithm has strong stability, fast convergence speed, high accuracy of clustering and better effect.

**Key words:** clustering; particle swarm; chaotic sequence; FCM (fuzzy C-means); CPSO-FCM algorithm

## 0 引言

聚类作为数据挖掘的方法已被广泛应用于图形图像处理、数据分析统计、模式识别等领域, 模糊 C 均值 (fuzzy c-means, FCM) 算法是聚类分析中一种经典算法<sup>[1]</sup>, 在 1971 年由 Bezdek 首先提出。算法的优势在于能够处理大规模数据集, 其不足之处体现在: 对初始值的选取较为敏感, 若不当, 极易造成聚类结果不一致; 以梯度下降法为基础的目标函数会使计算过程陷入局部最优。

Web 页面信息聚类是将文法或语义相似的内容在一种无监督的学习环境下划分为一类, 过程无需手工标识等预处理, 问题的解属于优化范畴, 目标函数期望的极值以迭代方式得到<sup>[2]</sup>。为了保证实验结果的稳定性并获得全局最

优解, 搜索过程不应过分依赖于初始值, 宜采用随机策略。上述一应问题常规的算法包括遗传算法、蚁群算法、粒子群算法等。

近年来, 基于粒子群的混合算法的相关研究较多, 文献 [3] 是将粒子群算法与 k-means 算法相结合, 执行快速的文档搜索, 有效避免局部最优, 但算法对初值较为敏感。文献 [4] 同样是一种混合策略, 文中引入了单纯搜索, 提出 K-NM-PSO 算法, 该方法提高了搜索最优解的速度。文献 [5] 利用混沌序列计算粒子的速度和位置, 并聚类得到新的速度更新公式, 提高了算法收敛速度和精度。文献 [6] 是基于模糊 C-均值的混合粒子群算法, 算法将 FCM 与 PSO 有效结合, 避免了对初值的依赖, 效果较明显。文献 [7] 则是 C-均值与量子粒子群的结合, 算法参数较少,

收稿日期: 2012-05-08; 修订日期: 2012-07-23

作者简介: 张春娜 (1979-), 女, 辽宁鞍山人, 硕士, 讲师, 研究方向为分布式计算; 李轶然 (1978-), 男, 辽宁鞍山人, 硕士, 讲师, CCF 会员, 研究方向为数据挖掘。E-mail: zcn1979@yahoo.com.cn

全局搜索能力也较强。

以上文献考察的重点涉及初始值敏感度、全局搜索能力、局部逃脱能力、收敛速度、遍历性等，由于切合点多不宜囊括，因此也是本文需特别考虑的问题。

粒子群优化算法 (particle swarm optimization)，简称 PSO，是一种新近发展起来的进化算法，思想源于鸟群、鱼群等群体觅食行为<sup>[8-9]</sup>。算法假定空间中每个粒子都可作为一个解，最优解产生于迭代过程中。每次迭代中粒子由自身速度决定其位置的变化，粒子群则紧跟当前最优粒子，并进行自身的搜索，最终找到个体和全局的极值。算法简单明确，无需过多人工干预，收敛速度快。基于此，文本将粒子群优化算法引入 C-均值聚类，以对 Web 页面信息实施归纳。

### 1 混沌粒子群算法

#### 1.1 混沌搜索

空间散列点属于非线性范畴，混沌现象亦如此，其具有两大特点：遍历性与规律性，即能够有规律的遍历到所有给定区域内的状态，且不重复。粒子搜索若采用混沌策略显然要优于简单的随机搜索<sup>[10]</sup>。搜索步骤如下：

(1) 定义初始区域，设 N 维初态向量  $R_0 = (R_{01}, R_{02}, \dots, R_{0N})$ ， $R_0$  中各个值之间相邻，且差异很小。

(2) 利用 logistics 方程对初始向量  $R_0$  进行计算，产生混沌序列  $m_1, m_2, \dots, m_n$ 。这里，经过数次迭代后，系统将完全处于混沌状态。向量层可表达为

$$m_{i+1} = m_i(1 - m_i)\lambda \tag{1}$$

式中： $\lambda$ ——迭代控制参数。

(3) 设空间粒子  $X_i$ ，利用式 (1) 得到  $X_i$  更好的位置，记  $X_i'$

$$X_i' = r \cdot rnd \cdot m_j + X_i \tag{2}$$

式中： $r$ ——粒子  $X_i$  活动半径， $rnd \in [-1, 1]$ ， $j \in [0, n]$ 。

#### 1.2 带混沌的粒子群算法

PSO 算法中粒子在寻找个体和全局的极值运动中具有某种随机性，即当粒子搜索过程中，若到达局部最优解时，其它粒子会紧随其后，纷纷落入该区域；而粒子自身的运动又可能无法使其脱离局部最优的束缚，因此很容易陷入局部最优。混沌的引入，使得粒子的运动具有一定规律，且兼具遍历特性。同时，混沌机制还可反作用于 PSO 以使粒子快速逃离局部最优，取得更好的极值搜索效果，算法流程如下：

(1) 粒子群参数初始化：包括粒子群的空间度，即群体大小；粒子的初始速度与位置；搜索迭代次数、混沌序列的长度以及其它必要的参数。

(2) 适应值计算：计算当前粒子的适应值，若个体当前值好于前值，则更新粒子位置；依次统计全部粒子当前值，若个体极值中最佳解好于前全局最优解，则替换之。

(3) 混沌优化：设当前最优解为  $P_b = (P_{b1}, P_{b2}, \dots, P_{bk})$ ，记作  $P_{bi}$ ， $i \in [1, k]$ ，将其映射到式 (1)，得  $m_i = (P_{bi} - a_i)/(b_i - a_i)$ ，而后利用 logistics 方程迭代出混沌变量序列，并逆映射到解空间求得极值解  $P_b' = (P_{b1}', P_{b2}', \dots, P_{bk}')$ 。最后，遍历解空间每个解，计算其适应值，取得可替换解  $P_b''$ 。

(4) 搜索群体中每个粒子，若优化后的解优于当前解，则将其当前位置替换为  $P_b''$ 。

(5) 查看当前状态是否满足搜索条件，若满足，则当前解为最优解；否则，返回继续计算适应值。

## 2 模糊 C-均值与粒子速度优化

### 2.1 模糊 C-均值算法分析

模糊 C-均值聚类算法 (FCM) 本质上是一种数据分类法<sup>[11]</sup>，依据拉格朗日最小二乘法原理，采用迭代法反复计算采样数据的聚类中心和分类矩阵中各元素的值，使其隶属度满足条件，簇内方差最小。

设 Q 维空间  $S^Q$  中有限集  $X = \{x_1, x_2, \dots, x_n \mid x_i \in S^Q\}$ ，利用模糊矩阵  $u$  将向量划分为  $m$  类， $m \in (1, n)$ ，初始化簇的中心为  $Z = \{Z_1, Z_2, \dots, Z_m\}$ ，定义目标函数如下

$$J(u, Z) = \sum_j \sum_i u_{ji}^r D_{x_i, z_j}^2 \tag{3}$$

式中： $u_{ji}$ ——隶属度，即第  $i$  个文本属于第  $j$  个类的隶属度； $D_{x_i, z_j}^2$ ——第  $i$  个文本到第  $j$  聚类中心的距离； $r$ ——权重因子，是影响模糊矩阵的模糊度，文中设定  $r \in [1, 3]$ 。

另，式 (3) 中模糊矩阵元素  $u_{ji}^r$  应满足

$$\sum_{j=1}^m u_{ji}^r = 1, \sum_{i=1}^n u_{ji}^r > 1 \tag{4}$$

其中， $u_{ji} \in [0, 1]$ ， $0 \leq i \leq n$ ，且有  $0 \leq \sum_i u_{ji}^r \leq n$ 。

可知，极限情况下，当  $J(u, Z)$  取得极小值， $u_{ji}^r$ 、 $D_{x_i, z_j}^2$  可得最优值。这里，依据拉格朗日最小二乘法，调整  $J(u, Z)$  中的  $u, Z$  可参考下式

$$u_{ji} = \frac{1}{\sum_{k=1}^m \left(\frac{D_{x_i, z_k}}{D_{x_i, z_j}}\right)^{2(r-1)}} \tag{5}$$

$$Z_j = \frac{\sum_{i=1}^n (u_{ji})^r x_i}{\sum_{i=1}^n (u_{ji})^r} \tag{6}$$

这里，若  $\|x_i - Z_j\| = 0$ ，则  $u_{ji} = 1$ 。

式 (5)、式 (6) 不断迭代，并调整  $u_{ji}$ 、 $Z_j$  的值；当  $J(u, Z)$  收敛到一定程度，即满足约定，则得到聚类最终结果。算法步骤如下：

(1) 初始化训练样本集，随机产生  $m$  个聚类中心，记作  $Z^1 = \{Z_{11}, Z_{12}, \dots, Z_{1m}\}$ ；

(2) 根据式 (5) 以及给定的初始聚类中心  $Z^1$  计算隶

属性  $u^k$  ;

(3) 重新计算聚类中心  $Z^k$  ;

(4) 设定迭代限定条件, 若  $|\max(u^{k+1}) - \max(u^k)| < \delta$ , 则终止; 否则, 迭代算子  $k$  递增, 并返回 (2)。

优化算法伪代码如下:

```
public bool fn _CluCal (int i _MCount, double d _t,
double d _CZ [], int i _Count, ObjText ob _XText)
{
fn _InitZ (i _MCount, d _CZ [i _Count]);
//初始化聚类中心
ObjAttchedD ob _AD;
Ob _AD=fn _AttD (ob _x, d _CZ [i _Count]);
//计算隶属度
d _CZ [+ + i _Count] = fn _CluD (Ob _AD, ob _
XText); //引入隶属度重新计算聚类中心
return (d _CZ [i _Count+1] - d _CZ [i _Count]) <
d _t; //算法终止条件判断
}
```

### 2.2 群体内粒子速度的优化

聚类的效果主要是考察收敛的满足度, 群体中每一个粒子都有一定的自身速度, 且时刻在调整, 其轨迹来自于自身的飞行经验, 并受到其他粒子的影响。因此, 算法的分析不应将个体视为孤立存在的点, 而是将这些点当作互相联系、互相依托的点来分析。

定义 1 加速因子

通常, 随着种群的移动, 若速度下降过快, 群体很容易陷入局部最优。实验测得, 此现象在 FCM 中尤为明显, 往往是算法没有结束, 就快速的收敛。究其原因, 算法在迭代过程中, 速度会逐渐下降, 且最终会降为 0; 速度降到某一极值时, 搜索能力也会随之下降, 导致不足以跳出局部最优的束缚。这里, 设定阈值  $\epsilon$ , 若速度小于此值, 则调节当前速度, 即  $v_i < \epsilon$ , 则  $v_i = \lambda v_0$ 。  $v_0$  为初始速度, 即粒子最大速度;  $\lambda$  为加速因子, 设定  $0 < \lambda < 1$ 。

定义 2 逃离算子

物种繁衍有种特性称为分群, 即当某一种群聚集大量个体, 若空间相对狭小, 部分群体会分离出去, 另建新的群落。粒子位置的考量亦是基于此, 参考加速因子, 若速度小于阈值  $\epsilon$ , 文中设定一个逃离算子  $\gamma$ , 其选定某一位置作为当前位置来替代粒子的最优位置, 并以当前速度继续迭代, 即  $v_i < \epsilon$ , 则  $p_g = p_i$ 。这里, 需特别提出替代位置的选定宜采用随机策略, 这样既可以保证原有群体的稳定性, 又不失群体的多样性。

现有  $Q$  维空间, 设第  $i$  个粒子, 可表示为  $q_i = (q_{1Q}, q_{2Q}, \dots, q_{iQ})$ , 其适应值可表示为  $p_i = (p_{1Q}, p_{2Q}, \dots, p_{iQ})$ , 速度记作  $v_i = (v_{1Q}, v_{2Q}, \dots, v_{iQ})$ , 对于每次迭代第  $k (1 \leq k \leq Q)$  维方程如下所示

$$v_{ik}^n = \lambda v_{ik}^{n-1} + \alpha_1 \beta_1^{n-1} (p_{ik}^{n-1} - q_{ik}^{n-1}) + \alpha_2 \beta_2^{n-1} (p_{gk}^{n-1} - q_{ik}^{n-1}) \tag{7}$$

$$q_{ik}^n = q_{ik}^{n-1} + v_{ik}^n \tag{8}$$

其中,  $\lambda$  为加速因子, 取值应适中, 不宜过大或过小, 应在迭代过程中适时调整。  $\alpha_1, \alpha_2, \beta_1, \beta_2$  为修正因子,  $\alpha_1, \alpha_2$  设定同值,  $\in [0, 3]$ ;  $\beta_1, \beta_2$  为随机数,  $\in (0, 1)$ 。

通过对速度的修正, 粒子在  $Q$  维空间搜索过程自由度更大, 且自身不易陷入局部最优, 收敛速度适中, 迭代次数以限定条件为准。

## 3 带混沌的 C-均值模糊聚类算法

### 3.1 算法描述

Web 网页信息有个重要特征-类别多样性。给定的页面内容被归属于多个类, 聚类过程即是群体依据归属度划分为不同的类, 群体中同类个体距离相对较近, 而不同类个体距离相对较远。惯常使用的 Web 聚类往往不善于考察类间关系, FCM 算法是一种利用隶属度来描述群体类别的聚类方法, 算法基于梯度下降的搜索策略, 初始状态随机分割  $m$  个有限集, 采用迭代重定向将群集内对象按照一定规则移动, 从而改变集合的形态。这里需要提出两个问题: 随机的初始化策略和寻优策略, 二者是否存在一定风险, 答案是肯定的, 前者可导致聚类结果不一致, 后者会造成结果陷入局部最优。解决的方式可采用混合策略, 结合 PSO (粒子群优化算法), 再辅以混沌序列则可有效解决上述问题, 不仅聚类效果好且收敛速度也较快。

PSO-FCM 算法亦需要考虑两个问题: 首先, 如何将文本向量转化为粒子群个体; 再者, 建立什么样的适应度模型来描述粒子群聚类。

算法将文本向量映射到粒子群, 以聚类编码格式体现, 群体中一个粒子对应一个文本集的可能解, 其存在形态是离散的。参考 3.1 的样本空间, 抽象出文本共有特征  $C$  个, 则任意文本可表示为  $x_i = \{x_{i1}, x_{i2}, \dots, x_{iC}\}$ ; 有聚类中心  $m$  个, 定义为  $Z_i$ , 粒子的位置由其组成; 粒子的长度可定义为  $Q \times m$ , 速度可用  $v_i = \{v_{i1}, v_{i2}, \dots, v_{iC}\}$  表示。适应度数学模型如图 1 所示。

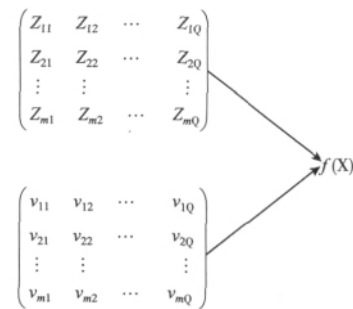


图 1 聚类编码格式

依据上文知，FCM 的最终目的是通过不断的修正隶属度与聚类中心以期获得适应度函数的稳定。极值的确定来自于适应度函数与各个聚类中心的距离，据此设定

$$f(x_i) = \frac{1}{J(u, Z)}$$

式中  $J(u, Z)$  取得最小值时，得最优解。

### 3.2 算法流程

基于混沌粒子群的 C-均值模糊聚类优化算法如下：

输入：网页文档提取的特征向量；

输出：分类文档，内含特定特征向量簇及孤立点簇。

步骤 1 规整特征向量成训练集。

步骤 2 初始化粒子群大小  $N$ ，设定迭代次数，起始  $k=0$ ，最大迭代次数为  $k_{max}$ ，设置群体内粒子的位置  $(x_i^k | x_i^1, x_i^2, \dots, x_i^N)$  和速度  $(v_i^k | v_i^1, v_i^2, \dots, v_i^N)$ ，混沌序列中粒子活动半径  $r$ ，修正因子  $\alpha_1, \alpha_2, \beta_1, \beta_2$ ，随机设定  $m$  个聚类中心  $Z^1$ 。

步骤 3 计算初始适应度，选取其中每个粒子最好的位置（最佳的适应度值），移动粒子，以此位置作为每个粒子的初始迭代位置，即  $P_{hi}^k = X_{hi}^k$ 。同时，适应度函数进行相应变化，即  $f(P_{hi}^k) = f(X_{hi}^k)$ 。

步骤 4 依式（1）、式（2）在当前迭代中引入混沌序列，计算其产生的粒子位置，并与当前位置进行比较，若优，则替换之。

步骤 5 位置替换过程中，在考虑加速因子和逃逸算子的前提下，速度依式（7）、式（8）加以更新。

步骤 6 迭代中不断比较粒子群全局适应度，选择历史记录中最优的全局适应度值，用此位置替换当前粒子所在的全局最优位置。

步骤 7 终止条件判定：①隶属度差值是否触及限定值  $\delta$ ；②迭代次数是否一定到达最大迭代限定数  $k_{max}$ 。

若满足上述任一条件，则得到全局最优解  $P_{gb}^k$ ；否则，递增迭代步长算子  $k$ 。

## 4 实验分析

### 4.1 数据采集

数据采集通过 google 中文搜索引擎达成，共选取三组数据来验证。

数据 1：wine 数据集是关于葡萄酒的口味测试数据，包括葡萄酒结构、酒精、酸以及丹宁；

数据 2：ISleep 数据集来自于对提高睡眠质量的调查结果，包括饮食习惯、情绪状态、睡眠环境、睡姿等；

数据 3：Smartphone 数据集是考察客户对智能手机技术指标的关注度，包括无线接入互联网能力、PDA 功能、系统扩展能力、人机交互性以及软件支持度。

对于从 Web 中采集的数据经过预处理后，文档的大小不等，经统计在 231B-29KB 之间，复杂度亦不尽相同，个别文档主题词较多。依此文档为基准，分别采用 k-means、

FCM、PSO-FCM、CPSO-FCM 这 4 个算法进行比较分析。

数据分析见表 1。

表 1 数据分析

数据集名称	个体个数	主题词数	聚类数	各类对象数
wine	142	4	4	46 : 45 : 51
ISleep	225	8	4	35 : 56 : 42 : 92
Smartphone	2326	5	4	520 : 368 : 855 : 583

### 4.2 结果分析

实验结果包括四类算法聚类准确率的最大、最小及平均值，见表 2。

表 2 聚类准确率比较

数据集名称	比较类型	k-means (%)	FCM (%)	PSO-FCM (%)	CPSO-FCM (%)
wine	max	87.5	90.2	90.1	90.2
	min	61.2	85.7	89.5	89.7
	average	76.8	89.1	89.7	90.1
ISleep	max	79.5	77.7	78.1	78.2
	min	68.4	75.2	76.9	77.8
	average	75.1	76.8	77.6	78.1
Smartphone	max	54.6	52.8	53.1	53.3
	min	48.1	51.1	52.5	52.9
	average	51.3	52.3	52.8	53.2

从表 2 是可知，4 种算法中 k-means 算法聚类准确度的最低、最高值与均值相差最大，原因是算法对初始值较敏感，导致聚类结果值出现跳跃，不够稳定。FCM、PSO-FCM 相较 k-means 更稳定，二者相差不大；由于混合后的 FCM 自身寻优能力增强的同时，稳定性也有所提高，聚类效果明显改善。CPSO-FCM 引入了混沌机制，执行结果更接近于最优解，稳定性也最好。

## 5 结束语

本中将 PSO 算法与 FCM 算法相结合进行搜索寻优，在解决 FCM 算法对初值敏感且容易陷入局部解的同时，亦优化了聚类结果。而在提高全局搜索能力方面，则引入混沌序列，使粒子移动能力增强，避免随机性。在算法后期，由于粒子速度的下降，又引入加速因子与逃离算子，重新调整速度，从而使粒子更容易摆脱局部束缚，加快收敛速度。实验结果证明，k-means、FCM、PSO-FCM、CPSO-FCM 这 4 种算法的比较中，优化后的 CPSO-FCM 算法无论是全局搜索能力，还是稳定性都优于其它算法。

### 参考文献：

[1] Li Chaoshun, Zhou Jianshong, Kou Pangao, et al. A novel chaotic particle swarm optimization based fuzzy clustering algo-

- rithm [J]. Neurocomputing, 2012, 83: 98-109.
- [2] HE Zhiping, XU Xuezhou, LI Ailing. Extracting topic information of Web page based on entropy [J]. Computer Engineering and Applications, 2007, 43 (4): 164-166 (in Chinese). [贺智平, 徐学洲, 李爱玲. 一种基于信息熵的 Web 页面主题信息抽取方法 [J]. 计算机工程与应用, 2007, 43 (4): 164-166.]
- [3] MAN Chuntao, ZHANG Rui, ZHANG Tao. Immune particle swarm optimization algorithm based on K-means clustering [J]. Control Engineering of China, 2010, 17 (6): 796-799 (in Chinese). [满春涛, 张锐, 张涛. 基于 K-均值聚类的免疫粒子群优化算法 [J]. 控制工程, 2010, 17 (6): 796-799.]
- [4] Kao Yi-Tung, Zahara E, Kao I-Wei. A hybridized approach to data clustering [J]. Expert Systems with Applications, 2007 (34): 1754-1762.
- [5] QIAN Weiyi, NING Bifeng. Clustering particle swarm optimization algorithm based on chaos [J]. Computer Engineering and Design, 2011, 32 (2): 685-688 (in Chinese). [钱伟懿, 宁必锋. 基于混沌的聚类粒子群优化算法 [J]. 计算机工程与设计, 2011, 32 (2): 685-688.]
- [6] Izakian H, braham A. Fuzzy C-means and fuzzy swarm for fuzzy clustering problem [J]. Expert Systems with Applications, 2011 (38): 1835-1838.
- [7] Wang Hao, Li Danyun, Chu Yayun. A new scalability of hybrid fuzzy C-means algorithm [C] //Artificial Intelligence and Computational Intelligence, 2010 (3): 55-58.
- [8] Schoeman I L, Engelbrecht A P. A novel particle swarm niching technique based on extensive vector operations [J]. Natural Computing, 2010, 9 (3): 683-701.
- [9] HUANG Xuan, ZHANG Jun, ZHAN Zhihui. Faster particle swarm optimization with random inertia weight [J]. Computer Engineering and Design, 2009, 30 (3): 647-650 (in Chinese). [黄轩, 张军, 詹志辉. 基于随机惯性权重的快速粒子群优化算法 [J]. 计算机工程与设计, 2009, 30 (3): 647-650.]
- [10] Araujo E, Coelho L S. Particle swarm approaches using Lozi map chaotic sequences to fuzzy modelling of an experimental thermal-vacuum system [J]. Applied Soft Computing, 2008 (8): 1354-1364.
- [11] WANG Hongchun, PENG Hong. Incremental clustering algorithm based on FCM [J]. Microelectronics and Computer, 2007, 24 (6): 156-161 (in Chinese). [王洪春, 彭宏. 基于模糊 C-均值的增量式聚类算法 [J]. 微电子学与计算机, 2007, 24 (6): 156-161.]
- (上接第 992 页)
- [10] Reeves WT. Particle systems—a technique for modeling a class of fuzzy objects [J]. Computer Graphic Forum, 1983, 17 (3): 359-376.
- [11] CHEN Xi, WANG Zhangye, PENG Qunsheng, et al. An integrated algorithm of real-time fluid simulation on GPU [J]. Journal of Computer-Aided Design & Computer Graphics, 2010, 22 (3): 396-405 (in Chinese). [陈曦, 王章野, 彭群生, 等. GPU 中的流体场景实时模拟算法 [J]. 计算机辅助设计与图形学学报, 2010, 22 (3): 396-405.]
- [12] LUO Yuetong, XUE Ye, SUN Jing. “Depth peeling” based transparency rendering and its application [C] //Proceedings of China Instrument’s 9th Youth Conference, 2007: 593-596 (in Chinese). [罗月童, 薛晔, 孙静. 基于“深度剥离”的透明渲染技术及其应用 [C] //中国仪器仪表学会第九届青年学术会议论文集, 2007: 593-596.]
- [13] WU Han, YANG Kejian. Kd-Tree based multi-dimensional indexing in the database application [J]. Computer Applications, 2007, 26 (9): 37-39 (in Chinese). [吴涵, 杨克俭. 基于 kd 树的多维索引在数据库中的运用 [J]. 计算机应用, 2007, 26 (9): 37-39.]