

# 基于蚁群优化算法的彩色图像颜色聚类研究

胡新荣<sup>1,2</sup>, 李德华<sup>1</sup>, 王天珍<sup>3</sup>

<sup>1</sup>华中科技大学 图像识别与人工智能研究所, 湖北 武汉 430074)

<sup>2</sup>武汉科技学院 计算机科学系, 湖北 武汉 430074)

<sup>3</sup>武汉理工大学 图像处理研究所, 湖北 武汉 430070)

**摘 要:** 颜色聚类是计算机视觉和图形学中基本的处理问题, 提出了蚁群聚类算法在彩色图像颜色聚类中应用的新思路. 根据蚁群 pick-drop 机制的基本原理, 提出了在 RGB 彩色空间中, 改进的蚁群算法在彩色图像像素的颜色值聚类中的应用, 最后进行像素的颜色映射, 从而完成量化工作. 实验表明, 采用该算法进行色彩的量化更具鲁棒性, 颜色失真小, 同时实现简单.

**关键词:** 蚁群聚类算法; 颜色聚类; 颜色量化; pick-drop

中图分类号: TP317

文献标识码: A

文章编号: 1000-1220(2004)09-1641-03

## Research on Color Quantization of Color Images Based on Optimized Ant Colony Algorithm

HU Xin-rong<sup>1,2</sup>, LI De-hua<sup>1</sup>, WANG Tian-zhen<sup>3</sup>

<sup>1</sup>Institute for Pattern Recognition and Artificial Intelligence, HuaZhong University of Science&Technology, Wuhan 430074, China)

<sup>2</sup>Department of Computer Science&Technology, WuHan Institute of Science&Technology, Wuhan 430074, China)

<sup>3</sup>Institute of Image Processing, Wuhan University of Technology, Wuhan 430070, China)

**Abstract:** It is a primary process of computer vision and graphics to color clustering. A new approach using ant colony algorithm applied in color images is proposed. According to the theory of pick-drop, a promoted ant algorithm is used to cluster colors of images in the color space of RGB. It completes quantization of color mapping of every pixel. Experimental results show that the algorithm proposed in the papers has the advantages of excellent robustness, less distortion, and has simple realized.

**Key words:** ant colony algorithm; color clustering; color quantization; pick-drop

### 1 引言

长期以来人们主要致力于灰度图像的研究并取得了很好的效果, 但彩色图像能比灰度图像提供更多的信息. 同时, 人眼能感受到的自然界中的真实物体也全都是彩色的, 因此, 彩色图像的处理受到更多的关注. 但是, 由于彩色图像所包含的数据量非常大, 做复杂处理如彩色图像的识别、检索、编码和分割等处理时会带来时间上的消耗, 因此对彩色图像颜色量化的研究是十分有意义的.

颜色聚类也称为颜色量化, 色彩量化或色彩聚类, 它是指在一些无法显示全彩色的硬件设备上显示全彩色图像时, 计算机需要选择合适的色彩描述集合并将图像的各像素点的颜色用集合中的色彩表示, 这就是“颜色量化”<sup>[1]</sup>. 传统的聚类分析在分类前没有一致类别属性的样本做参考, 而且往往也不知道样本究竟应该分多少类, 只给出若干样本, 要求根据他们的性质进行归类.

传统的聚类方法可以分为层次聚类法和分割聚类法两

种. 层次聚类法输出的是一棵层次化的分类树, 而分割聚类法输出的是互不相交的聚类集. 分割聚类法又包括试探性算法和迭代算法两大类. 试探性算法通常按某种就近原则聚类, 这种聚类算法计算简单, 但因为缺乏误差调节机制, 聚类效果往往难以预料. 迭代算法则从一个初始分割开始, 反复迭代直至某个控制函数达到最优. 比较典型的迭代算法有 C—均值算法、ISODATA 算法等等, 这些算法能够很好的解决试探性算法的误差问题, 但需要确定一些经验参数和阈值, 如聚类的个数以及初始聚类中心的设定等. 另一种迭代算法即自适应算法, 该算法不需要经验参数和阈值的设定, 从初始分割开始, 经过多次聚类分裂和合并的迭代, 自适应的达到用户满意的聚类效果.

在彩色图像的颜色量化过程中, 基于上述的传统聚类算法, 由于需要依次为图像的每一个像素的颜色值进行计算和比较, 因此时间复杂度很高. 针对传统算法的不足, 笔者提出利用蚁群聚类算法来完成量化工作.

## 2 蚁群聚类算法

### 2.1 蚁群算法的提出

在生物界,蚂蚁是一种头脑简单、视力也很不好的小东西,然而蚂蚁群体通过相互协调、分工、合作来完成筑巢、觅食、迁徙、清扫蚁穴等复杂行为,它们都属于复杂系统研究的范围,即相对简单的个体在没有一个集中控制的情况下,通过相互作用产生复杂的群体行为。那么数以百万计的蚂蚁如何组成一个群落?它们是如何相互协调、分工、合作来完成这些任务呢?像蚁巢这样复杂结构的信息又是如何存储在这群蚂蚁当中呢?各个领域的专家已经对这些问题有了长期、深入的研究,提出了一些解释并建立了一些算法模型。这些算法就是蚁群算法的雏形。

蚁群算法是模拟自然界蚂蚁寻找食物过程中的一种分布式、启发式搜索算法。最早是由 Colomi 等人提出的,主要用于求解复杂的组合优化问题,如旅行商问题(Traveling Salesman Problem, TSP)、加工调度问题(Job Shopping Scheduling Problem, JSSP)、图着色问题(Graph Coloring Problem, GCP)等。下面通过蚂蚁 pick up-drop 行为的最短路径策略来简要介绍蚂蚁聚类算法的基本原理。

### 2.2 蚁群聚类算法的基本思想

生物学家在研究中发现自然界中的一个奇特现象,即蚂蚁能够将蚂蚁巢穴中的尸体聚集成几堆。如果一个地方已经有一些尸体的聚集,那么它将吸引蚂蚁将其余的尸体放在这里,越聚越多,最终形成几个较大的尸体的聚集堆。Deneubourg 等人对上述现象提出了解释,并提出了描述蚂蚁行为基本模型(Basic Model, BM),这种模型主要是基于对于单只蚂蚁拾起、放下物体的行为方式进行建模。一只随机移动的无负载蚂蚁在遇到一个物体时,周围与这个物体相同的物体越少,则拾起这个物体的概率越大;一只随机移动的有负载蚂蚁如果周围的与所背负物体相同的物体越多,则放下这个物体的概率越大。这样可以保证不破坏大堆的物体,并且能够收集小堆的物体。

受社会性昆虫群的聚集行为的启发,研究者们提出了许多聚类算法的尝试,本文中提出的蚁群聚类算法在彩色图像颜色量化中的应用就是受到蚂蚁 pick up-drop 行为的启发而提出的。实验表明,这种方法可以将相同种类的物体聚集在一起,如彩色图像的分割、数据挖掘、超大规模集成电路的设计等等。

如果用每个采样的样本表示蚂蚁巢穴的空间结构,并且各样本之间不需要集中控制或所处环境的空间描述,则通过对不同的样本数据对象进行测量, Lumer 和 Faieta 将 Deneubourg 等人的 BM 推广应用到数据分析。其主要思想是<sup>[2]</sup>:

(1) 将待聚类数据初始随机散放在一个二维平面上,然后在这个平面上每个坐标点处产生一些虚拟的“蚂蚁”,即作为聚类中心的数据对象,将这些数据视作具有不同属性的蚂蚁;

(2) 平面上的单个蚂蚁的行为和上面 BM 中所描述的单个蚂蚁行为相似。不同之处在于,它们不是观察当前所背负的

目标数据与其周围的数据是否相同,而是判断是否相似;

假设待进行聚类分析的数据集合  $O = \{O_i | O_i = (o_{i1}, o_{i2}, \dots, o_{im})\}$  表示蚂蚁要寻找的食物源的集合,则相似性判据规则为:

$$f(O_i, O_j) = \sum_{O_j \in \text{area}(O_i, r)} (1 - d(O_i, O_j)) \quad (1)$$

式中  $d(O_i, O_j)$  表示数据对象  $O_i$  和  $O_j$  的加权欧氏距离,  $d$

$$(O_i, O_j) = \sqrt{\sum_{k=1}^m P_k (O_{ik} - O_{jk})^2};$$

$P_k$  表示自适应加权参数,其值可根据在聚类过程中的作用大小进行动态调整;

$\text{area}(O_i, r)$  表示数据对象  $O_i$  的聚类领域半径  $r$ ;

$f(O_i, O_j)$  表示数据对象  $O_i$  和  $O_j$  的相似度,选择合适的加权因子  $P_k$ , 使  $0 \leq f(O_i, O_j) \leq 1$ , 当  $f(O_i, O_j) = 0$  时,  $O_i$  和  $O_j$  的相似度最小,当  $f(O_i, O_j) = 1$  时,  $O_i$  和  $O_j$  的相似度最大,即二者完全相同;

由上述公式可以看出,当数据对象  $O_i$  和  $O_j$  越相似,则它们之间的加权欧氏距离  $d(O_i, O_j)$  越小,相似度  $f(O_i, O_j)$  也就越大。

(3) 利用概率转移函数计算蚂蚁拾取数据对象(pick up)概率和放下数据对象(drop)概率;

概率转移函数的基本模型<sup>[2]</sup>:

$$\text{拾取数据对象(pick up)概率: } P_p = \left( \frac{1}{1 + f_p/f} \right)^2$$

$$\text{放下数据对象(drop)概率: } P_d = \left( \frac{1}{1 + f_d/f} \right)^2$$

在上式中,  $f_p, f_d$  分别表示被拾取数据对象和被放下数据对象与当前数据对象的相似度阈值。

在本文的研究工作中,采用了简化的概率转移模型:

$$\text{拾取数据对象(pick up)概率: } P_p = 1 - f(O_i, O_j) \quad (2)$$

$$\text{放下数据对象(drop)概率: } P_d = f(O_i, O_j) \quad (3)$$

对于无负载的蚂蚁,只需计算其拾取数据对象(pick up)概率  $P_p$ ; 对于有负载蚂蚁,则只需计算其放下数据对象(drop)概率  $P_d$ 。

(4) 根据计算出来的概率来完成单个负载蚂蚁的聚集行为,最终能够将相似的数据聚为一类,形成一些可见的聚类中心。

用  $P_0$  表示概率阈值,  $O_j$  是否归并到  $O_i$  由下式给出:

对无负载蚂蚁,若  $P_p < P_0$ , 蚂蚁不拾取数据对象;否则,蚂蚁拾取数据对象,成为有负载蚂蚁。

对有负载蚂蚁才能进行聚类,如果  $P_d > P_0$ , 蚂蚁放下数据对象,同时将蚂蚁的属性赋值给数据对象,此时蚂蚁的状态又成为无负载;否则蚂蚁就负载继续移动。

令  $C_i = \{O_j | f(O_i, O_j) \geq f, j = 1, 2, \dots, I\}$ ,  $C_i$  表示所有归并到  $O_i$  邻域的数据集合,可以根据下式求出理想的聚类中心(之所以称为理想是因为该中心可能在样本集中不存在):

$$C_i = \frac{1}{I} \sum_{j=1}^I O_j, \text{ 其中: } O_j \in C_i$$

### 3 基于蚁群聚类算法的颜色聚类

在基于蚁群算法的聚类分析中,可以将数据视为具有不同属性的蚂蚁,聚类中心看作是蚂蚁所要寻找的“蚂蚁堆”。所以,数据聚类过程就看作是蚂蚁寻找不同蚂蚁堆的过程。

本实验中,无负载蚂蚁的数据结构:

```
typedef struct Ant_unloaded
{
    int x;           //蚂蚁的 X 坐标
    int y;           //蚂蚁的 Y 坐标
};
```

有负载蚂蚁的数据结构表示为:

```
typedef struct Ant_loaded
{
    int x;           //蚂蚁的 X 坐标
    int y;           //蚂蚁的 Y 坐标
    BYTE red;        //数据对象的颜色:红色分量
    BYTE green;      //数据对象的颜色:绿色分量
    BYTE blue;       //数据对象的颜色:蓝色分量
};
```

数据对象的数据结构表示为:

```
typedef struct data_object
{
    BYTE red;        //数据对象的颜色:红色分量
    BYTE green;      //数据对象的颜色:绿色分量
    BYTE blue;       //数据对象的颜色:蓝色分量
};
```

基于蚁群聚类算法的颜色聚类算法的具体步骤如下:

(1) 根据经验或具体需求选择若干个代表点,将这些代表点作为初始的聚类中心;

选择代表点的方法很多<sup>[3]</sup>,在本文的研究工作中,根据问题的具体特点和性质,用经验的方法确定类别数,从数据对象集合中找出直观上看起来比较适合的代表点作为初始的聚类中心。

(2) 初始化一些特定的参数:自适应加权参数  $P$ ,蚂蚁初始数目  $num$ ,聚类领域半径  $r$  等;

(3) 将待颜色聚类的二维图像置于二维平面上,图像中的每个坐标对即像素代表一个虚拟的“蚂蚁”,假定“蚂蚁”的初始状态是有负载的;

(4) 蚂蚁数据集开始进入聚类循环;

(5) 按照公式(1)计算数据对象  $O_i$  与聚类半径  $r$  邻域中的其他数据对象的相似度:

$$f(O_i, O_j) = \sum_{O_j \in area(O_i, r)} (1 - d(O_i, O_j))$$

式中,数据对象  $O_i = (R_i, G_i, B_i)$  代表每个像素的红、绿、蓝颜色分量;

(6) 如果蚂蚁是有负载的,按照公式(3)计算拾取概率  $P_d$ ;如果蚂蚁是无负载的,按照公式(2)计算放下概率  $P_p$ ;

(7) 根据计算出来的概率来完成单个负载蚂蚁的聚集行为,最终能够将相似的数据聚为一类,形成一些可见的聚类中心。

(8) 如果数据集合中所有的蚂蚁完成了拾取数据和放下

数据的动作,则继续执行下列步骤,否则选择下一个没有拾取数据和放下数据动作的蚂蚁转至步骤(4)继续执行;

(9) 调整自适应加权参数  $P$  和其他参数如聚类中心等;

(10) 如果数据聚类未完成,转至步骤(4)继续循环;否则,输出聚类结果,算法执行完毕。

### 4 实验结果分析

通过计算机的实验研究来考察算法性能,下面以计算机生成的图 1 为例说明。图 1 是一幅  $300 \times 200$  大小、32 位真彩色图像,为了视觉上观察方便起见,在分类之前实现对其进行颗粒化处理。图 1 中共包含有 80 多种颜色,可以采用本算法将其颜色分成 12 类,效果如图 2 所示。



图 1 原图像



图 2 颜色量化后的图像

Fig. 1 Original image Fig. 2 Color quantized image

通过上述两幅图像可以看出,虽然进行了颜色归并处理,但处理后的图像主要颜色与原图像主要颜色很接近,同时处理前后物体的形状也很接近,色块效应较好,可见本算法中像素的聚类效果是比较理想的。由于算法中是将每类中拥有像素点较多的颜色作为该类的主要颜色,这样就避免了一些传统算法中可能将一些噪声点的颜色值保留到量化后的图像中从而引起失真的缺点,同时算法实现的复杂度降低,鲁棒性较好。实验表明,对于具有特定要求的场合,本文提出的颜色聚类算法是比较有效的。

### References:

- 1 What is color quantization [EB/OL]. <http://www.faqs.org/faqs/jpeg-faq/part1/section-8.html>.
- 2 <http://www.amss.ac.cn/iss/conferences/sino-japan2001/ShiZhongZhi-ppt.pdf>
- 3 Yang Xin-bin, Sun Jing-hao, Huang Dao. Research on fault diagnosis based on outlier mining[J]. Computer Engineering and Applications, 2002, 9: 12-13.
- 4 Wu Yin-qian, Hu Xin-rong, Yang Cui-ping, et al. A new approach to color quantization of images based on neural networks[J]. Journal of Wuhan Automotive Polytechnic University, 2000, 2: 33-36.

### 附中文参考文献:

- 1 杨欣斌,孙京浩,黄道. 基于蚁群聚类算法的离群挖掘方法[J]. 计算机工程与应用, 2002(9): 12-13.
- 1 吴颖谦,胡新荣,杨翠萍等. 一种新的基于神经网络彩色图像颜色量化方法[J]. 武汉汽车工业大学学报, 2000(2): 33-36.