

一个基于链表的快速色彩量化算法

余庆军^{1,2} 余报楚³ 唐祯安¹

(大连理工大学电子与信息工程学院 大连 116023)¹ (大连东软信息学院数字艺术系 大连 116023)²

(大连理工大学工业装备与结构分析国家重点实验室 大连 116023)³

摘要 为了提高色彩量化算法的质量和速度,作者考虑在设计调色板的过程中充分利用分裂算法的快速性和聚类算法的准确性,试图改进像素匹配过程,然后在其基础上提出一个新的实时色彩量化算法。该算法将图像中的所有颜色按照其出现次数的大小排列成一个数据结构链表。整个色彩量化过程可视为关于该链表的一系列操作。实验结果表明,该算法能够获得期望的结果。

关键词 色彩量化,逐对聚类,数据结构,链表

中图分类号 TP751.1 **文献标识码** A

Fast Color Quantization Algorithm Based on Linked List

YU Qing-jun^{1,2} YU Bao-chu³ TANG Zhen-an¹

(School of Electronics and Information Engineering, Dalian University of Technology, Dalian 116023, China)¹

(Digital Art Department, Dalian Neusoft Institute of Information, Dalian 116023, China)²

(State Key Laboratory of Structural Analysis for Industrial Equipment, Dalian University of Technology, Dalian 116023, China)³

Abstract In order to improve the quality and speed of color quantization algorithm, we took the rapidity of splitting algorithm and the accuracy of clustering algorithm into account during the design process of color palette, and tried to improve the process of pixel matching, and then proposed a new real-time color quantization algorithm. In this algorithm, all colors in an image were connected together via a data structure link according to its appearance frequency. The whole process of color quantization could be viewed as a series of operations about that data structure link. Experimental results show that the proposed algorithm is able to achieve the expected results.

Keywords Color image quantization, Pairwise clustering, Data structure, Linked list

1 引言

色彩量化算法通常由调色板的设计与像素的匹配这两个主要部分组成。其最终目的是在原始彩色图像的基础上构造一个合适的调色板,使得依据该调色板所获得的重建图像的失真度最小化。

关于调色板的设计,目前学术界已经提出了很多算法^[1-8]。这些算法大致可分为两大类:一类是分裂算法,中值切割算法与均值切割算法等是其中最常用的几个算法^[1,2]。另一类是聚类算法^[3,4]。一般来说,分裂算法执行速度快,但是无法保证获得最优解。至于聚类算法,尽管它能够获得一个局部最优解,但是其计算复杂程度高,而且输出结果严重依赖初始条件。更为关键的是,现有色彩量化算法都将调色板设计与像素匹配视为两个独立的过程,因而均难以保证色彩量化的实时性^[1,8]。

本文试图整合现有分裂算法和逐对聚类算法的优势,优化像素匹配过程,并在其基础上提出一个新的快速色彩量化

算法。本文第2节提出一个快速的逐对聚类色彩量化算法;第3节对所提出算法进行算法分析与仿真;最后给出一些结论。

2 所提出的算法

为了提高色彩量化的实时性,我们需要设法减少运算量、运算复杂度以及数据量。为了提高结果图像的质量,我们则需要设法获得最优的调色板。显然,现有的分裂算法和聚类算法在这两方面各有优势。鉴于此,本文提出的算法试图充分利用分裂算法的快速性和聚类算法的最优性优势。

为了便于描述,不妨使用红绿蓝(RGB)三基色来表示图像中的颜色值,并将颜色的数据结构定义如下:

```
typedef struct _stColor {
    int hist; // 给定颜色的出现次数
    int R; // 给定颜色的R值
    int G; // 给定颜色的G值
    int B; // 给定颜色的B值
    int num; // 给定颜色在数据链上的索引号
```

到稿日期:2010-03-29 返修日期:2010-06-30 本文受辽宁省科技计划重大项目(2008217004),中国博士后科学基金(200704111071)资助。

余庆军(1973—),男,博士后,高级工程师,副教授,主要研究方向为数字媒体技术和游戏设计与开发, E-mail: yuqingjun@neusoft.edu.cn;

余报楚(1974—),男,博士后,副教授,主要研究方向为计算力学、结构工程等。

```

struct _stColor * past;// 指针,指向与给定颜色相邻的前一个颜色
struct _stColor * next;// 指针,指向与给定颜色相邻的后一个颜色
struct _stColor * down;// 指针,指向挂接在给定颜色下方的一个颜色

```

```

} stColor;

```

本文提出的算法流程图如图 1 所示。其具体步骤可描述如下。

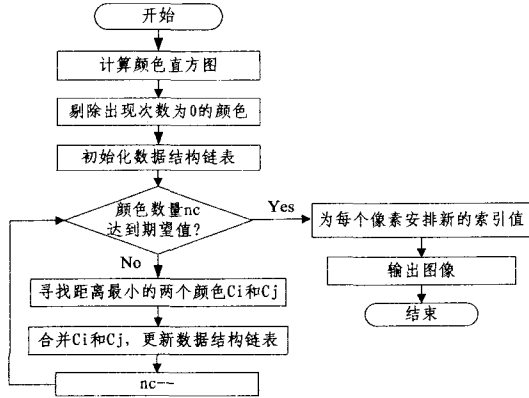


图 1 所提出算法的流程图

步骤 1 计算颜色直方图,即统计图像中每个颜色的出现次数。

步骤 2 剔除出现次数为 0 的颜色。为方便起见,我们不妨令 $C = \{C_1, C_2, \dots, C_N\}$ 为给定图像中出现次数大于 0 的所有颜色的集合,其中 N 为颜色总数, C_1 代表出现次数最多的颜色, C_N 代表出现次数最少的颜色。同时,令 $H(C_i)$ 表示颜色 C_i 的出现次数,其中 $i = 1, \dots, N$ 。值得注意的是,为了追求计算速度,甚至可以考虑剔除出现次数在某个特定数字以下的颜色。

步骤 3 构造由集合 C 中的所有颜色组成的数据结构链表,如图 2 所示。同时,按照如下方式对每个颜色结构进行初始化赋值:

$$\begin{aligned}
C_i \rightarrow hist &= H(C_i) \\
C_i \rightarrow num &= i \\
C_i \rightarrow past &= C_{i-1} \\
C_i \rightarrow next &= C_{i+1} \\
C_i \rightarrow down &= 0
\end{aligned}$$

其中, $1 < i < N - 1$ 。如果 $i = 1$, $C_i \rightarrow past = \text{Null}$; 如果 $i = N$, $C_i \rightarrow next = \text{Null}$ 。这里 Null 代表空值。

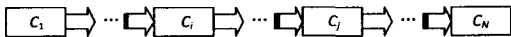


图 2 初始化数据结构链表

步骤 4 判断数据结构链表中的颜色数量是否减少到期望值。如果没有,则进入步骤 5; 否则进入步骤 8。

步骤 5 寻找数据结构链表上距离最小的两个颜色。不失一般性,我们不妨假设最小距离所对应的两个颜色为第 i 个和第 j 个颜色,即 C_i 和 C_j , 而且 $i < j$ 。这里,我们将颜色 C_i 和 C_j 之间的距离的平方 d_{ij}^2 定义为

$$d_{ij}^2 = (C_i \rightarrow R - C_j \rightarrow R)^2 + (C_i \rightarrow G - C_j \rightarrow G)^2 + (C_i \rightarrow B - C_j \rightarrow B)^2 \quad (1)$$

当然,我们还可以采用加权的方法计算颜色之间的距离以获得最佳的视觉效果。

步骤 6 采用下述方法对颜色 C_i 和 C_j 进行加权合并:

(1) 根据式(2)计算合并后的颜色:

$$C_{mp} = \frac{H(C_i)C_j + H(C_j)C_i}{H(C_i) + H(C_j)} \quad (2)$$

式中, $H(C_i)$, $H(C_j)$ 分别表示颜色 C_i 和 C_j 的出现次数。

(2) 根据式(3)计算合并后所得到的新颜色的出现次数 $H(C_{mp})$ 为:

$$H(C_{mp}) = H(C_i) + H(C_j) \quad (3)$$

(3) 将颜色 C_{mp} 赋值给颜色 C_i , 即

$$C_i \leftarrow C_{mp} \quad (4)$$

(4) 更新颜色 C_i 的出现次数, 即

$$H(C_i) \leftarrow H(C_{mp}) \quad (5)$$

(5) 将颜色 C_j 从数据链表上删除, 并将其挂接到颜色 C_i 的下面, 如图 3(a) 所示。也就是需要按照下述方式更新数据结构链表:

$$C_j \rightarrow past = C_j \rightarrow next \quad (6)$$

$$C_i \rightarrow down = C_j \quad (7)$$

式中, $C_j \rightarrow past$ 表示颜色 C_j 的前一个颜色, $C_j \rightarrow next$ 表示颜色 C_j 的后一个颜色, $C_i \rightarrow down$ 表示挂接在颜色 C_i 下的一个颜色, $C_j \rightarrow num$ 表示颜色 C_j 在数据链表上的索引号。

值得注意的是, 如果碰到颜色 C_j 下带有挂接颜色, 即 $C_j \rightarrow down$ 不等于空的情况时, 则直接将 C_j 下的所有颜色挂接到颜色 C_i 的下面, 如图 3(b) 所示。这里, 我们假设颜色 C_j 下挂接有颜色 C_k 。容易证明, 合并颜色 C_i 和 C_j 所得到的新颜色为颜色 C_i , C_j 和 C_k 所构成的聚类的中心。

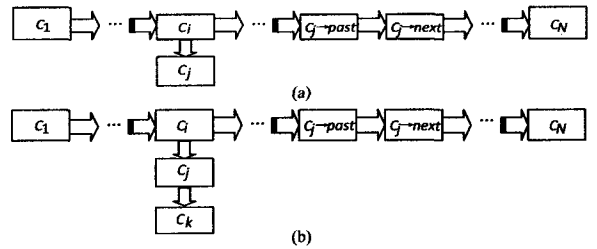


图 3 数据结构链表操作示意图

步骤 7 重复步骤 4, 直到获得期望的颜色数量为止。

步骤 8 将数据结构链表上的每个颜色视为调色板上的颜色, 并将其所有挂接颜色赋值为该颜色, 即可得到颜色减少的结果图像。

3 算法分析与仿真

3.1 算法分析

本文提出的算法将图像中每个颜色视为一个数据结构链表中的单个元素, 然后将整个颜色量化过程视为与该链表相关的一系列关于链表元素的删除、合并和挂接等操作过程。

在数据结构链表的设计过程中, 我们剔除了出现次数小于某个特定数值的颜色, 从而可在一定程度上减少数据量以及一些无用的运算量。同时, 鉴于分裂算法的特点, 我们对链表上的每个颜色按其出现次数的大小进行了排序, 从而为后续操作提供了极大的方便。

在颜色合并时, 将颜色的出现次数作为加权项, 容易得出, 颜色出现的次数越多, 合并后的颜色越与之接近。也就是说, 调色板上的颜色将与颜色出现次数多的更加接近。这也与常用的流行的分裂算法具有一定的相似性。

同时, 本文提出的算法在颜色合并与删减的过程中应用

了逐对聚类算法,因而继承了聚类算法的结果最优化优势。而且,在颜色合并时只需计算新的颜色与数据链表上其它颜色之间的距离,因为,先前已经计算过所有颜色之间的距离。

另外,本文提出的算法在调色板的设计过程中就已经完成了像素颜色的匹配,而且只需对链表上的颜色而无需对图像中所有像素颜色进行逐对聚类合并,因此可节省大量的运算工作量。

3.2 仿真实验

为了验证所提算法的有效性,我们针对 Lena, Mandrill, Girl, Sina46 等不同的图像做了一系列的实验,并且在均方差、运行时间这两个测量标准的基础上将其与 Luiz(1997)^[4]提出的逐对聚类算法以及 Cheng(2001)^[5]提出的快速、新颖的分裂算法进行了比较。限于篇幅,本文只提供了如表 1 以及图 4 所示的仿真结果。所有这些结果均是在 Linux 环境下获得的。

表 1 性能比较

| 颜色数 | 本文方案 | | Cheng ^[5] 方案 | | Luiz ^[4] 方案 | |
|-----|--------------|------|-------------------------|------|------------------------|------|
| | 时间(μ s) | MSE | 时间(μ s) | MSE | 时间(μ s) | MSE |
| 2 | 33720 | 2358 | 12656 | 2131 | 122841 | 3493 |
| 4 | 33776 | 1420 | 20778 | 865 | 124019 | 2452 |
| 8 | 33750 | 440 | 28660 | 552 | 127485 | 390 |
| 16 | 33749 | 233 | 36359 | 274 | 132661 | 197 |
| 32 | 33610 | 115 | 44779 | 143 | 142820 | 99 |
| 64 | 33165 | 53 | 54854 | 78 | 160908 | 42 |
| 128 | 29518 | 21 | 59365 | 40 | 187030 | 11 |

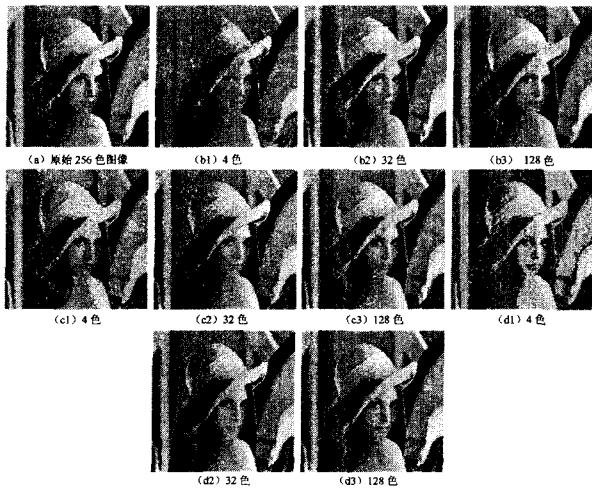


图 4 (a)原始图像,(b1-b3)本文方案仿真结果,(c1-c3)Cheng 方案仿真结果,(d1-d3)Luiz 方案仿真结果

根据实验结果对比,我们容易得出,在结果图像颜色数小于等于 8 时,本文方案与 Cheng 方案的速度差不多;但是当结果图像颜色数大于 8 时,本文方案则比它要快,而且所获得的图像质量要好。至于 Luiz 所提出的算法,本文方案与 Cheng 方案均比它快得多,但是当结果图像颜色数大于 8 时,该算法可获得三者之中质量最好的图像。由此,本文方案相对来说具有更快的运行速度,而且能够获得比较令人满意的结果图像。不过,需要说明的是,本文只是提供了一个色彩量化算法的框架,我们完全可以在其基础之上进行扩展如可以在颜色距离检测以及在颜色合并的过程中引入人类视觉特性,以使结果图片更加接近原始图像。

结束语 本文提出了一个基于数据结构链表操作的、综合考虑了传统分裂算法和逐对聚类算法优势的、快速的、可扩展的色彩量化算法。本算法首先借鉴分裂算法的特点构造数据结构链表,然后直接利用逐对聚类算法进行颜色合并,最后通过链表操作完成像素颜色的快速匹配。实验结果证明,本算法不仅具有易于实现、可扩展的特点,而且能够在保证结果图像质量的同时,提高色彩量化的效率。

参考文献

- [1] Heckbert P S. Color image quantization for frame buffer display [J]. *Computer & Graphics*, 1992(16): 297-307
- [2] Wu X, Witten I H. A fast k-means type clustering algorithm [R]. Canada; Departement of Computer Science, University of Calgary, Calgary, 1985
- [3] Xiang Zhigang, Gregory J. Color image quantization by agglomerative clustering [J]. *IEEE Computer Graphics and Application*, 1994, 14(3): 44-48
- [4] Velho L, Gomes J, Sobreiro M R. Color image quantization by pairwise clustering [A]// *X Brazilian Symposium of Computer Graphics and Image Processing [C]*. IEEE Computer Society, 1997: 203-210
- [5] Cheng S-C, Yang C-K. A fast and novel technique for color quantization using reduction of color space dimensionality [J]. *Pattern Recognition Letters*, 2001(22): 845-856
- [6] Omran. A color image quantization algorithm-based on particle swarm optimization [J]. *Informatica*, 2005(29): 261-269
- [7] Chang Yu-chou, Lee D-J, Hong Yi, et al. A Robust Color Image Quantization Algorithm Based on Knowledge Reuse of K-Means Clustering Ensemble [J]. *Journal of Multimedia*, 2008, 3(2): 20-27
- [8] Yoon K-J, Kweon I-S. Human Perception Based Color Image Quantization [C]// *17th International Conference on Pattern Recognition*. 2004(1): 664-667

(上接第 249 页)

- [7] Shi Kai-quan, Li Xiu-hong. Camouflaged information identification and its applications [J]. *An International Journal Advances in System Science and Applications*, 2010, 10(2): 208-216
- [8] Zhang Li, Cui Yu-quan. Outer P-sets and data internal-recovery [J]. *An International Journal Advances in Systems Science and Applications*, 2010, 10(2): 229-236
- [9] Liu Ji-qin. P-probabilities and its applications [J]. *An International Journal Advances in Systems Science and Applications*, 2010, 10(2): 237-244
- [10] Lin Hong-kang, Li Yu-ying. P-sets and its P-separation theorems [J]. *An International Journal Advances in Systems Science and Applications*, 2010, 10(2): 245-251

- [11] Huang Shun-liang, Wang Wei, Geng Dian-you. P-sets and its internal p-memory characteristics [J]. *An International Journal Advances in Systems Science and Applications*, 2010, 10(2): 252-260
- [12] Wang Yang, Geng Hong-qin, Shi Kai-quan. The mining of dynamic information based on P-sets and its applications [J]. *An International Journal Advances in Systems Science and Applications*, 2010, 10(2): 267-275
- [13] Zhang Guan-yu, Li En-zhong. Information gene and identification of its information knock-out/knock-in [J]. *An International Journal Advances in Systems Science and Applications*, 2010, 10(2): 278-286