

# 70-71,69 电脑分色系统几个核心算法的设计\*

Some Key Color District Algorithms's Designing of the Color Image Color Analysis System

徐汀荣

TS193

(苏州大学计算机工程系 苏州215006)

**Abstract** The paper induced the technology of the designing and realizing of color image special district algorithm

**Keywords** Color image, Computer color district, Special district, Algorithm

## 1. 引言

现代印染业发展非常迅速,人们对印染品的质量和效果的要求也在不断提高,当前彩色印染方式为专色印染.专色印染是在一块白色的布匹上每次印染一种特定的颜色,经过若干次印染之后,生成完整的印染品.套色印染是将三种基本颜色(如:品红、青、黄)进行叠加印染生成各种颜色.无论是专色印染还是套色印染,都需要进行分色处理,传统方法是用手工进行分色处理,制约了企业生产能力的提高.

采用电脑分色系统是降低成本、增强功能的有效方法.其原理是首先由彩色扫描仪将要印染的样图扫描成数字信息写入磁盘文件,然后电脑分色系统按照用户的要求进行分色处理,生成分色图像,再根据分色图像进行制版.这样就能够得到印染所需的影印板底板(如:印染机上的铜板).采用电脑分色系统之后,能够有效地提高分色处理的效率和质量,从而缩短印染周期,节省人力,提高质量.因此,电脑分色系统的应用对印染行业生产能力的提高将起到很大的促进作用.

专色分色适用仅有几种颜色构成的彩色图像的印染,专色分色算法是将该彩色图像信息分解成若干种颜色并写入各自的单色图像文件.在实际操作中,彩色图像经过光电扫描仪输入后的颜色可能达数百种.即原来的一种颜色变为相近的几十种颜色,而实际印染不可能按这每种颜色印染一次.因此专色分色处理首先确定需要将原始图像分成几种颜色,其次要确定选取哪些颜色,最后根据选定的颜色生成分色图像.

## 2. 确定分色数的算法设计

确定分色数可以有两种方法,一是由用户直接指

定分色数,二是由计算机根据用户给出的条件通过某种算法间接地给出分色数.

由用户来指定最后应分成几种颜色,这种方法适合于颜色较为集中、色块较为分明的图像.

由计算机确定分色数,一般根据用户给出的条件,例如归并中最大的色差,然后依照一定的算法确定分色数.这种方法适合于色彩较为丰富、颜色过渡较为细腻的图像.在这里我们定义:

$$\text{色差} = (R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2$$

其中 $\{R_1, G_1, B_1\}$ 和 $\{R_2, G_2, B_2\}$ 分别表示两种颜色.由于在计算机中, $R, G, B$ 的值为 $0 \sim 255$ ,所以最大色差 $= 255^2 * 3$ .由此,可以引进参数“归并色差”来控制颜色归并过程中的色差范围,且归并色差 $=$ 最大允许色差/最大色差,则归并色差的取值范围是 $0 \sim 1$ .这样当用户输入归并色差后,由归并色差就可以得到最大允许色差,从而实现了对颜色归并过程的色差范围控制.因此,在由计算机确定分色数时,用户输入归并色差.但是,当用户输入归并色差之后,分色数并没有随之确定,而必须由计算机在允许的色差范围内对颜色进行归并,当计算机再也无法归并时,剩余的颜色数即为分色数,而在此过程中,也完成了对颜色的选择.因此确定分色数的过程与选择颜色的过程是合一的.所以,确定分色数的算法包含于下面所介绍的选择颜色的算法.

## 3. 专色分色选择颜色的算法设计

选择颜色的算法有很多.例如首先可以对彩色图像中的每一种颜色赋以权值(某颜色的权值即为图像中该颜色的点的数目),然后依权值的大小,选择权值高的颜色,依照这种算法,可以使图像中较多出现的颜

\* 本项目受江苏省自然科学基金和江苏省计算机信息处理重点实验室资助.



且仅出现一次的回路。这恰好又是我们算法 H2 的特例。也就是说，如果 G 中有 H 回路的话，在我们的 H2 算法中已经求出来了，我们可以从 H2 算法的推导中轻松地得到求 H 回路的算法，完全同理，下面我们仅给出求 H 回路算法的说明和修正部分，然后直接给出求 H 回路的算法。

$$18) \text{Hcycle} \leq = \{P | P \in \text{cycle2}(A) \wedge (P) = V\}$$

类似地，path2 将变成

$$19) \text{path4}(n, n', N, K) \leq = \{P | P = \langle n, n_1, \dots, n_l \rangle \wedge \forall i: 1 \leq i < l \quad n_{i+1} \in f(n_i, E) \wedge n_l \in N \wedge D(P) = K \wedge n' \in f(n_l, E) \wedge \text{sample2}(P)\}$$

单点情况变化如下：

$$20) \text{path}_k(n, n', \{v\}, K) \leq = \{P | P = \langle n \rangle \rightarrow \langle v, m_1, \dots, m_k \rangle \wedge \forall i: 1 \leq i < k \quad m_{i+1} \in f(m_i, E) \wedge m_k = n' \wedge m_1 \in g(v, k - \{n\}) \wedge n' \in f(n_1, E) \wedge D(P) = K \wedge \text{sample2}(P)\}$$

实例化方程 19 并根据 RS4 和 NO3

注意到此时若  $n' \in f(v, E)$ ，且  $K - \{n\} = \{v\}$ ，则一条符合条件的路已经求出，并且从  $v$  出发不可能再走下去，由此上面方程可继续推导为：

$$\leq = \{P | P = \langle n, v \rangle\}$$

若  $n' \in f(v, E) \text{ and } K - \{n\} = \{v\}$

$$\leq = \{P | P \in \langle n \rangle \rightarrow P_1 \wedge P_1 = \text{path4}(v, n', g(v, K - \{n\}), K - \{n\})\}$$

其它

最后有求解 H 回路的算法 H4 如下：

$$H4_1 \text{Hcycle} \leq = \text{path4}(u_0, u_0, g(u_0, V), V) + \{u_0\}$$

$$H4_2 \text{path4}(n, n', N_1 \cup N_2, K) \leq = \text{path4}(n, n', N_1, K) \cup \text{path4}(n, n', N_2, K)$$

$$H4_3 \text{path4}(n, n', \{v\}, K) \leq = \{P | P = \langle n, v \rangle\}$$

若  $n' \in f(v, E) \text{ and } K - \{n\} = \{v\}$

$$\leq = \{P | P \in \langle n \rangle \rightarrow P_1 \wedge P_1 = \text{path4}(v, n', g(v, K - \{n\}), K - \{n\})\}$$

其它

$$H4_4 \text{path4}(n, n', \Phi, K) \leq = \Phi$$

$$H4_5 \text{path4}(n, n', N, \Phi) \leq = \Phi$$

**结束语** 算法综合是提高软件可靠性、可维护性、可重用性、最终实现软件自动生成的重要手段之一。本文旨在利用演绎和重用技术寻求一条高效的形式化开发同类问题串型算法族的有效途径，将算法综合转向并行化和部件化，以更大提高算法软件的开发效率和执行效率是我们正在研究的新课题。

### 参考文献

- 1 Darlington J. A synthesis of several sorting algorithms. Acta Informatic. 1978(11):1~30
- 2 Manna Z, Waldinger R. Synthesis: Dream  $\Rightarrow$  Program. IEEE Transation on Software engineering. 1979, SE-5(4):291~328
- 3 黄明和, 邓少敏. 程序变换综述. 计算机与现代化, 1991. 1
- 4 许卓群, 张乃孝, 杨冬青, 唐世渭. 数据结构. 高等教育出版社, 1987
- 5 黄明和. 树遍历算法的演绎综合. 计算机工程与科学, 1997. 1

(上接第 71 页)

2	4	8	4	2
或者:	X	5	3	
2	4	5	4	2
	2	3	2	

有抖动算法不能够建立颜色对应表，因为在原图像中同一种颜色的点经过分色处理后的颜色未必一样（因为抖动处理时，被分散的色差将改变点的颜色），所以只能动态地生成对应颜色。其分色算法描述如下：

(1) 读入原图像的一点的颜色，在被选中的颜色表中选择与之色差最小的颜色，在其相应的分色图像中置该颜色的标志，在其它的分色图像中置空白（背景）色。

(2) 求得处理之后该点的颜色与原始图像中该点的色差，然后根据抖动算法将色差分散，改写相应点的颜色（原始图像中该点的颜色并不能改变）。

(3) 重复进行 (1)、(2) 的处理，直至原图像中所有的点均被处理。

以上介绍了两种专色分色处理的算法

### 5. 分色的色点省略

在图像中往往有一些颜色的权值很小，但其颜色却很特别，即与其它颜色的色差很大，这些颜色在确定分色数的归并中就可能被保留，并致使其权值大的颜色被归并掉，使得处理后得到的图像失真严重；或者在确定归并色差的归并中，使分色数过大，为了解决这个问题，可以将一定权值以下的颜色先归并成相近的颜色，再进行专色分色处理。为此要引进“色点省略比率”这个系数，该系数定义如下：

$$\text{色点省略比率} = \frac{\text{要省略颜色的最大权值}}{\text{总权值}}$$

用户定义了色点省略比率之后，即可得到被省略颜色的最大权值（因为总权值即为图像总点数），小于该权值的颜色被归并为与之最为接近的颜色，归并算法同 3.1.2 中介绍的算法。通过省略权值过小的颜色，能够有效保留主要颜色，使专色分色处理获得较好的结果。