

彩色地图图象的聚色算法研究*

冯玉才 宋恩民

(华中理工大学计算机科学系 武汉 430074)

摘要 本文研究彩色地图图象中不纯净的颜色聚色问题,该问题是对地图图象进行自动识别的一个关键问题.本文对地图图象的颜色散射现象进行了分析,提出了对地图图象聚色的算法,该算法已经在 IBM486 等微机上编程实现了,效果很理想.

关键词 地图图象,聚色,识别,算法.

随着计算机技术的迅猛发展,人们对地图信息的自动处理的需求日益广泛和迫切,地图数据库、地图识别、电子地图导游、电子地图出版等实用系统正在逐步出现和完善,而如何将已有纸质地图的信息输入到计算机中则是这些系统能否实用的瓶颈.^[1~7]

作为地图信息来源的地图图象一般都是经扫描仪或摄像机输入到计算机中.一幅高质量的地图图象是正确提取地图信息的保证,而经扫描仪或摄像机输入的地图图象一般都会出现失真和畸变,最令人棘手的是颜色的散射,比如将一幅4色的彩色地图输入到计算机后,得到的真彩色地图图象中会出现成千上万种不同的色彩,虽然人们的肉眼分辨不出其中某些细微的差别,但这些色彩差别却给地图信息的正确提取带来了巨大的困难.^[1]

本文针对地图图象的散射进行了研究,给出了一个将地图图象聚色的算法,以使地图图象中散射出的千万种色彩恢复成原图的几种颜色.本文的算法已在 IBM486 微机上编程实现了,效果十分理想.

1 彩色图象的数学模型

本文有关术语和记号见文献[1~7],文中有上划线的字母均为向量,除 \bar{x} 为二维坐标向量外,其它向量均为三维颜色向量.

根据光学分色理论,各种不同的颜色构成一个三维的颜色空间 P ,它相当于笛卡儿坐标系的第一象限.在 P 中,由坐标原点引出任意一个颜色向量,其箭头所在位置即表示一种颜色,该向量的模为此颜色的亮度,而该向量在3个轴向上的分量即为此颜色含红、绿、蓝三原色的量.

设 \bar{C} 是任意一种颜色,则 \bar{C} 可被分解成红、绿、蓝三原色,也可由这三原色重新合成,这

* 本文研究得到国家自然科学基金资助.作者冯玉才,1945年生,教授,主要研究领域为数据库,多媒体技术,人工智能和图象识别.宋恩民,1962年生,副教授,主要研究领域为计算复杂性,算法设计与分析,地图输入识别.

本文通讯联系人:冯玉才,武汉 430074,华中理工大学计算机科学系

本文 1995-06-27 收到修改稿

可表示为:

$$\bar{C} = (r(\bar{C}), g(\bar{C}), b(\bar{C}))$$

其中 $r(\bar{C}), g(\bar{C}), b(\bar{C})$ 分别表示 \bar{C} 中所含红、绿、蓝三原色的量。

在实际中, $r(\bar{C}), g(\bar{C}), b(\bar{C})$ 的取值可以是某个实数区段中的连续量,而在计算机中,它们的取值被限定为 $[0, 255]$ 区段中的离散整数。对于一幅真彩色的地图,扫描仪根据选择的分辨率,将其分成若干象素点,每个象素点的颜色又按光学分色原理分成红、绿、蓝 3 种色素,每个象素点的 3 种色素再按其强弱程度分为 256 级,用 $0 \sim 255$ 的整数表示(即 $r(\bar{C}), g(\bar{C}), b(\bar{C})$),存放在一个 8 位的字节单元中。

一幅地图图象 Map, 可以表示成颜色 \bar{C} 在 x, y 二维空间中的变化,即 $\text{Map} = m(\bar{X})$, 其中 \bar{X} 为二维地图图象空间的坐标向量。

设 Map 上的坐标向量为 \bar{X} 的象素点具有颜色 $m(\bar{X}) = \bar{C}$, 则称 \bar{X} 占据了色空间 P 的 \bar{C} 位置,用 $R(\bar{C}, \bar{X})$ 表示色空间的 \bar{C} 与图象空间中的 \bar{X} 的关系,即

$$R(\bar{C}, \bar{X}) = \begin{cases} 1, & \text{若 } m(\bar{X}) = \bar{C}; \\ 0, & \text{否则.} \end{cases}$$

为了用统计参数来分析和描述图象,以达到聚色的目的,定义一幅地图图象 Map 的彩色直方图为

$$H(\bar{C}) = \frac{\text{Map 中颜色为 } \bar{C} \text{ 的象素点数}}{\text{Map 中象素点总数}}$$

该直方图表示了 Map 中象素点占据颜色空间 P 的疏密程度。

对 Map 聚色即为根据 $H(\bar{C})$ 的统计特性,将色空间划分为几个子空间,再将每个子空间内的全部色向量都映射到与各自子空间所对应的同一个色向量。

2 地图图象聚色的总体思路

对地图图象聚色的基本思想是:首先求图象的色彩分布,然后用统计方法决定最佳的聚色中心值和阈值面。

为了准确有效地将地图图象聚色,便于利用数学分析的有效工具,我们将计算机中的离散地图图象(坐标和颜色均为离散的)连续化,再对连续的地图图象进行处理,以达到理想的效果。其步骤如图 1。

3 算法描述

第 1 步:求 Map 的彩色(三维)直方图

$$H(\bar{C}) = \frac{1}{N_1 \times N_2} \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} R(\bar{C}, (x_1, x_2))$$

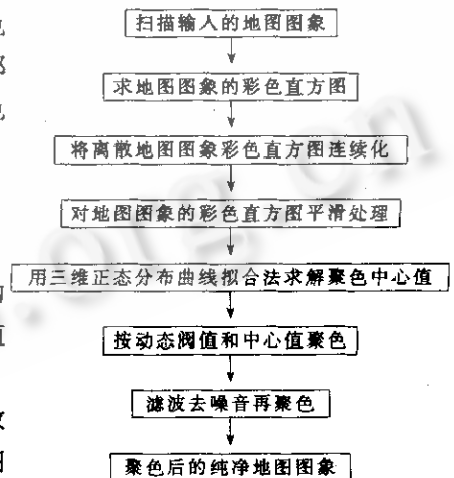


图1 地图图象聚色的步骤

其中 N_1, N_2 分别为 Map 每行和每列的象素点数.

第 2 步: 将地图图象的彩色直方图连续化.

为了将彩色直方图连续化, 可用正态分布函数作为核函数来代替离散彩色直方图中的函数值, 得到连续直方图,

$$\text{即 } F(\bar{U}) = \sum_{r=0}^{255} \sum_{g=0}^{255} \sum_{b=0}^{255} \left(\frac{1}{\sqrt{2\pi\alpha_1}} \exp[-\|\bar{U} - \bar{C}\|^2 / (2\alpha_1^2)] \cdot H(\bar{C}) \right)$$

其中 $\bar{C} = (r, g, b)$, \bar{U} 为颜色向量, α_1 为一个与原地图比例尺有关的常数.

第 3 步: 将彩色直方图平滑化.

$$\text{用滑动求积的方法平滑化. 选平滑函数 } h(\bar{U}) = \frac{1}{\sqrt{2\pi\alpha_2}} \exp[-\|\bar{U}\|^2 / (2\alpha_2^2)]$$

其中 α_2 为一个与地图图象分辨率有关的常数, \bar{U} 为颜色向量.

将 $F(\bar{U})$ 与 $h(\bar{U})$ 求卷积, 即可得到平滑后的彩色直方图

$$f(\bar{U}) = F(\bar{U}) * h(\bar{U}) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(\bar{C}) h(\bar{U} - \bar{C}) dr dg db$$

其中 $\bar{C} = (r, g, b)$.

第 4 步: 用正态分布曲线拟合法解聚色中心值.

统计数据表明, 地图的色散量是符合正态分布的, 若原彩色地图是 k 色地图, 则 Map 的彩色直方图将与 k 个正态分布函数的和相吻合, 这 k 个正态分布的数学期望即为原地图的 k 种颜色, 而均方差则与原地图的比例尺、质量(干净程度)、扫描分辨率三者相关.

设与原地图中第 i 种颜色相对应的色散分布函数为

$$\frac{1}{\sqrt{2\pi\sigma_i}} \exp[-\|\bar{U} - \bar{U}_i\|^2 / (2\sigma_i^2)]$$

其中 $\bar{U}_i = (r_i, g_i, b_i)$, 则总色散分布函数为

$$S(\bar{U}) = \sum_{i=1}^k \frac{1}{\sqrt{2\pi\sigma_i}} \exp[-\|\bar{U} - \bar{U}_i\|^2 / (2\sigma_i^2)]$$

以下只需求适当的 σ_i 和 \bar{U}_i 使 $S(\bar{U})$ 与 $f(\bar{U})$ 拟合的最好, 为此将 $S(\bar{U})$ 视为 σ_i, \bar{U}_i 的函数, 并引入总色散分布函数与彩色直方图的平方差函数, 它是 σ_i 和 \bar{U}_i 的函数

$$t(\sigma_1, \sigma_2, \dots, \sigma_k, \bar{U}_1, \bar{U}_2, \dots, \bar{U}_k) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (S(\bar{U}) - f(\bar{U}))^2 dr dg db$$

选取 σ_i 和 \bar{U}_i 使得 $t(\sigma_1, \sigma_2, \dots, \sigma_k, \bar{U}_1, \bar{U}_2, \dots, \bar{U}_k)$ 达到最小值时, $S(\bar{U})$ 与 $f(\bar{U})$ 拟合得最好, 这时的 $\bar{U}_1, \bar{U}_2, \dots, \bar{U}_k$, 即为原地图中的 k 个颜色.

为求 $\bar{U}_1, \bar{U}_2, \dots, \bar{U}_k$, 将 $t(\sigma_1, \sigma_2, \dots, \sigma_k, \bar{U}_1, \bar{U}_2, \dots, \bar{U}_k)$ 对 \bar{U}_i 的各分量和对 σ_i 依次求偏导并令其为 0, 再从得到的方程组中求 \bar{U}_i 和 σ_i , 方程组为:

$$\begin{cases} \frac{\partial t(\sigma_1, \sigma_2, \dots, \sigma_k, \bar{U}_1, \bar{U}_2, \dots, \bar{U}_k)}{\partial \bar{U}_i} = 0 \\ \frac{\partial t(\sigma_1, \sigma_2, \dots, \sigma_k, \bar{U}_1, \bar{U}_2, \dots, \bar{U}_k)}{\partial \sigma_i} = 0 \end{cases} \quad (i=1, 2, \dots, k)$$

第 5 步: 按动态阈值和中心值聚色.

上一步中求出了 σ_i 和 \bar{U}_i, \bar{U}_i 即为原地图中第 i 色, σ_i 为第 i 色在 Map 中的色散分布的

均方差. 因此各色的色散分布函数为:

$$q_i(\bar{U}) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\|\bar{U} - \bar{U}_i\|^2 / (2\sigma_i^2)\right] \quad (i=1, 2, \dots, k)$$

以下是将色空间 P 划分为几个子空间, 第 i 个子空间含有颜色 \bar{U}_i , 划分 P 的原则是: 对于 P 中的任意一个颜色 \bar{C} , 按色散分布函数, 若第 i 色散射到 \bar{C} 的概率最大, 就将 \bar{C} 划入第 i 个子空间. 按此原则, 划分子空间的色彩阈值将与色散分布函数和色空间的位置有关, 即阈值是随位置变化的, 称为动态阈值.

用 Map' 记聚色后的地图图象, 则有:

$$\text{Map}' = m'(\bar{x}) = \begin{cases} \bar{U}_1, & \text{若 } q_1(m(\bar{x})) \geq q_2(m(\bar{x})), q_3(m(\bar{x})), \dots, q_k(m(\bar{x})); \\ \bar{U}_2, & \text{若 } q_2(m(\bar{x})) > q_1(m(\bar{x})) \text{ 且 } q_2(m(\bar{x})) \geq q_3(m(\bar{x})), \dots, q_k(m(\bar{x})); \\ \dots & \\ \bar{U}_k, & \text{若 } q_k(m(\bar{x})) > q_1(m(\bar{x})), q_2(m(\bar{x})), \dots, q_{k-1}(m(\bar{x})). \end{cases}$$

第 6 步: 滤波去噪音.

在地图图象 Map 中, 可能会有一些噪音象素点, 在聚色的过程中, 也可能产生少量的噪音象素点, 因此很有必要对聚色后图象 Map' 进行滤波去噪音. 根据实际统计分析, 聚色过程产生的少量的噪音都具有高频特性, 而聚色后的整幅地图图象信号具有低频特性, 因此可用付里叶变换经二维低通滤波器对 $m'(\bar{x})$ 滤波去噪音.

令对 $m'(\bar{x})$ 作二维付里叶变换后的函数为 $M(u, v)$, 则

$$M(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} m'(x_1, x_2) \exp[-j2\pi(ux_1 + vx_2)] dx_1 dx_2$$

其中 $j = \sqrt{-1}$.

为了既能滤去图象 Map' 中的噪音, 又保持图象中的边缘不受损失, 选择如下平滑低通滤波器的传递函数:

$G(u, v) = (e^2 D^d)^{-D^d}$, 其中 $D = \sqrt{u^2 + v^2}$, d 是一个与 Map 分辨率有关的常数. 当 d 值为 0.7 时, 其图象如图 2.

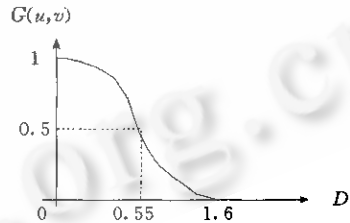


图 2 $d=0.7$ 时 $G(u, v) = (e^2 D^d)^{-D^d}$ 的图象

经滤波后的地图图象 Map'' 为:

$$\text{Map}'' = m''(\bar{x}) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} M(u, v) \cdot G(u, v) \exp[j2\pi(ux_1 + vx_2)] dudv$$

第 7 步: 再次聚色

经滤波后, Map'' 中还可能出现少量的杂色, 对 Map'' 再重复 1 遍第 5 步的过程, 得到的图象 Map''' 即为纯净的地图图象了.

4 算法实现

以上描述的算法在理论上能得到十分理想的地图图象, 经对小块样张地图图象试算表明有很好的结果, 但因其公式演算较复杂, 计算时间冗长, 故对大张地图进行处理时还需根据实际情况和客观要求进行一定的化简和近似, 方能实用.

我们用本文给出的算法分别对以 120~250dpi 扫描输入的几幅比例尺分别为 1:50 000, 1:250 000, 1:500 000, 1:1 000 000 的地图图象进行了聚色处理, 对算法中的各参数反复多次地试代了不同数值, 发现参数按如下规律选取时效果最为理想.

(1) 第 2 步中的常数 α_1 的选取. 下表是对不同比例尺的地图的经验参数.

比例尺 α	1:50 000	1:250 000	1:500 000	1:1 000 000
α_1	3	2.7	2.4	2.3

(2) 第 3 步中常数 α_2 的选取. α_2 与地图图象分辨率 β (单位为 dpi) 关系的经验公式为:

$$\alpha_2 = \sqrt{\beta/3}.$$

(3) 为了简化运算和减少求解时间, 第 4 步中的 σ_i 可以取为 1 个常量, 一般取 $\sigma_i = 10$ 即可, 对于 \bar{U}_i 的求解可选用统计方法求得或人为输入 1 个估计值, 再让机器在估计值附近搜索精确值.

(4) 第 6 步的 d 的选取. d 与地图图象分辨率 β (单位为 dpi) 的关系的经验公式为:

$$d = \sqrt{\beta}/20.$$

(5) 积分式的计算可以用累加运算来近似, 第 4 步中偏微分方程的求解可用分区段的折半查找法搜寻近似解.

本文的算法不仅对由较清洁的地图扫描输入的地图图象能较好地聚色, 也能对由被轻度污染了的地图扫描输入的地图图象自动聚色, 关于对不清洁的地图图象的聚色和颜色校正算法将另文探讨.

参考文献

- 1 冯玉才, 宋恩民, 孙小微等. 地图自动识别系统中按颜色分层的算法及实现. 软件学报, 1995, 6(7): 435~439.
- 2 周源华, 权淑媛, 刘惠娟. 地图的计算机识别. 上海交通大学学报, 1993, (6): 26~32.
- 3 王积分, 张新荣. 计算机图象识别. 北京: 中国铁道出版社, 1988.
- 4 程民德, 浓燮昌. 图象识别导论. 上海: 上海科学技术出版社, 1983.
- 5 Kast Uri R, Alemang J. Information extraction from images of paper-based maps. IEEE Trans. on Software Engineering, 1988, 14(5): 671~675.
- 6 吕建平等. 一种基于预测跟踪技术的地图识别方法. 自动化学报, 1991, 17: 77~82.
- 7 蒋庆权译. 地图识别输入技术. 现代电子工程, 1989, 1: 69~75.

AN ALGORITHM TO GATHER COLORS FROM IMAGE OF COLOR MAPS

Feng Yucai Song Enmin

(Department of Computer Science Huazhong University of Science and Technology Wuhan 430074)

Abstract The problem of gathering colors from image of color maps with impure colors is studied in this paper. This is a key problem in automatic recognition of map image. The phenomena of color scattering in map images are analyzed, and an algorithm to gather colors from image of color maps is presented. The algorithm has been implemented in the environment of computer of the type of IBM486 and it has been shown that the algorithm was efficient.

Key words Map image, gather color, recognition, algorithm.