

# 一种基于颜色的图像检索改进方法<sup>①</sup>

## An Improved Color-Based Image Retrieval Method

赵建民 袁 杰 朱信忠 徐慧英 (浙江师范大学 数理与信息工程学院 浙江 金华 321004)

**摘 要:** 基于颜色的图像检索由于其对图像的各种变形有很好的鲁棒性而得到了广泛的应用,但其对颜色的空间分布描述不足从而检索精度不佳。针对全局颜色直方图及局部颜色直方图的检索问题,提出了一种新的基于颜色的检索方法。该方法引入一种颜色转移矩阵,用于描述颜色的相对空间分布情况;同时改进传统局部颜色直方图匹配方法,使块匹配受图像形变的影响大为减少;最后结合上述两种图像子特征进行复合图像检索。实验表明,该方法能有效的提高图像检索精度。

**关键词:** 颜色量化 局部颜色直方图 颜色转移矩阵 最大匹配法 相似度

### 1 引言

基于内容的图像检索 (CBIR) 研究在最近十几年得到了长足的发展,目前研究得较多和较成熟的是利用图像的可视化底层特征进行检索。一般人们利用的底层特征有颜色、形状和纹理特征等<sup>[1]</sup>。

颜色是最直接描述物体或图像内容的视觉特征。颜色特征通常对噪声、尺寸的缩放、旋转等具有很强的鲁棒性。颜色特征的描述方法主要有颜色直方图<sup>[2]</sup>、累加直方图<sup>[3]</sup>、其它各种变形的颜色直方图<sup>[4]</sup>、颜色矩和颜色聚合向量等,但这些描述方法都不能较好的刻画颜色的空间位置分布信息,有时会找出所含内容不相关而颜色统计量相似的图像。

本文提出一种方法,结合颜色统计特性和颜色的各种位置特性,先规格化图像为统一大小,将 RGB 颜色空间转换到 HSV 颜色空间,再将 H、S、V 三个分量量化并线性组合成一个一维颜色矢量,大大减少了数据量,同时将规格化后的图像等分成块,一方面统计各分块的量化 HSV 特性,用直方图相交法得到检索例图中某块与目标图各块的相似度,找出目标图中最匹配块,每个块都这样处理后再将这些块最大相似度取平均得出两幅图的第一种相似度;另一方面取出每个更小块中的主色(跟上述块操作是同一过程,不过是在大块中分小块),每幅图构造一个主色矩阵,由主

色矩阵得出颜色转移矩阵,将两个颜色转移矩阵的相关系数作为两幅图间的第二种相似度,将上述两种相似度进行加权合并得到一个统一的相似度,最后将库中各图按它们与例图的相似度降序排列输出。该方法在实验中取得了较好的效果。

### 2 颜色量化

#### 2.1 HSV 颜色空间

常用的 RGB 颜色空间各颜色维之间的相关性很强,数据冗余度高,空间模型中的颜色距离与人的感知距离不一致。因此,需要用另一种符合人的视觉心理的颜色模型来表示颜色。

HSV 颜色空间采用色度 H、饱和度 S 和亮度 V 分离的方式实现颜色的定量描述,三维分量都存在明确的语义对应关系。相对于其它颜色模型,它更为准确地反映了人类知觉系统对色彩的理解方式。

RGB 到 HSV 的转换公式很多书上都有介绍,在此不再详述。转换后,  $H \in [0, 360]$ ,  $S, V \in [0, 1]$ 。

#### 2.2 HSV 颜色量化

为了减少数据量,需要对颜色空间进行量化。一般的,可以将 HSV 三个分量分别按式(1)和式(2)中的方案量化, C 为三个量量化后合成的颜色值<sup>[5]</sup>:

① 对于  $V \leq 0.2$  的颜色认为是黑色,  $C = 0$ ;

① 基金项目:国家自然科学基金(60773197);浙江省自然科学基金(Y107750)

收稿时间:2008-08-12

② 对于  $S \leq 0.1$ ，且  $V > 0.2$  的颜色按亮度  $V$  划分为三种灰度。分别为：

深灰(0.2,0.5]，浅灰(0.5,0.8]和白色(0.8,1]。C 的相应值分别为  $C=1,2,3$ 。

③ 其他颜色认为是彩色，将其划分为种彩色。

$$H' = \begin{cases} 0 & H \in [0,20) \cup [315,360] & \text{红色} \\ 1 & H \in [20,50) & \text{橙色} \\ 2 & H \in [50,75) & \text{黄色} \\ 3 & H \in [75,155) & \text{绿色} \\ 4 & H \in [155,195) & \text{青色} \\ 5 & H \in [195,275) & \text{蓝色} \\ 6 & H \in [275,315) & \text{紫色} \end{cases} \quad (1)$$

$$S', V' = \begin{cases} 0 & S, V < 0.2 \\ 1 & 0.2 \leq S, V < 0.7 \\ 2 & S, V \geq 0.7 \end{cases} \quad (2)$$

式(1)和式(2)中  $H, S, V$  分别表示由  $R, G, B$  三个分量转换过来的值，而  $H', S', V'$  表示量化后的值。63 种彩色经量化后由式(3)得出合成的一维矢量  $C$ ：

$$C = 4 + 9H' + 3S' + V' \quad (3)$$

### 3 特征提取

#### 3.1 颜色直方图

在颜色表征方面，全局颜色直方图对图像颜色的统计分布特征的表征具有平移、尺度、旋转不变性，但全局颜色直方图只记录了颜色的统计特性，对颜色空间分布刻画不足。另一方面，传统局部颜色直方图对颜色空间分布有一定的描述能力，但它有时又对图像内容的平移、旋转等变化很敏感。权衡以上各方面因素，本文选用局部颜色直方图来表征颜色，将图像分成块，不过对传统局部颜色直方图在块匹配方面作一些改进，使整个方法既能达到较好的鲁棒性，又对颜色空间分布有一定的描述能力。

从上一节可知，一个像素的 RGB 值经映射最后得到一个之间的数。也即每一个的块维护一个柄的颜色直方图，一幅图像共有个局部颜色直方图。

#### 3.2 颜色转移矩阵

本文提出一种建立在 HSV 量化颜色空间之上的刻画颜色相对分布位置的方法，称之为颜色转移矩阵法。由于人眼对于颜色的色调尤为敏感，并且颜色的色调(Hue)是一个独立于观察点的颜色特征<sup>[6,7]</sup>，基于

此，本文只取  $H$  的量化值构成主色。设共有  $k$  级，即。本文实验中取  $k=10$ ，则的范围为 0 到 9。具体取值见下式(4)：

$$C' = \begin{cases} H' & C \neq 0, 1, 2, 3 \\ 7 & C = 0 \\ 8 & C = 1, 2 \\ 9 & C = 3 \end{cases} \quad (4)$$

$C$  是上节中  $H', S', V'$  线性组合得到的一维颜色矢量。

得到一幅图像的颜色转移矩阵步骤如下：

① 将一幅图像分成  $m \times n$  个小块，每一小块的大小都为  $s \times t$  个像素。

② 得出每一小块的主值  $C'$ ，也就是该小块中  $C'$  取得次数最多的那个值。这样就形成一个二维主色矩阵，大小为  $m \times n$ ，表示为  $A = \{a_{i,j}\} \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n$ 。

③ 建立一个  $k \times k$  的矩阵  $P$  ( $k$  是  $C'$  的取值个数)，各元素初始值为 0。将上步得到的矩阵  $A$  按 Z 字扫描顺序进行扫描，设  $H_{i,j}$  与  $H_{p,q}$  是扫描序列中的一对相继出现的颜色 ( $H_{i,j}$  在  $H_{p,q}$  的前面)，则  $P$  中相应元素  $P[H_{i,j}, H_{p,q}]$  自增 1，如此反复，直到扫描完成。

④ 建立  $k \times k$  的矩阵  $D$ ， $D$  中元素的计算公式如下(5)式：

$$D_{i,j} = P_{i,j} / \sum_{u=1}^k \sum_{v=1}^k P_{u,v} \quad (5)$$

则矩阵  $D$  就是该图像的颜色转移矩阵。

从上面的步骤可以看出，矩阵  $D$  中的每个元素都描述了图像中各不同颜色的相邻关系及这种相邻关系在整幅图像所有像素点对中发生的比率，从而矩阵  $D$  能描述颜色的空间分布情况。

### 4 相似性度量及图像检索

#### 4.1 图像颜色直方图相似性度量

计算两个块  $b_1$ 、 $b_2$  间的颜色直方图距离时，本文采用经典的直方图相交距离，其数学描述为：

$$d(b_1, b_2) = \sum_{i=1}^n \min(b_{1,i}, b_{2,i}) \quad (6)$$

而两个块间相似性度量为：

$$Similarity(b_1, b_2) = \sum_{i=1}^n \min(b_{1,i}, b_{2,i}) / \sum_{i=1}^n b_{1,i} \quad (7)$$

本文中  $n$  取 67，对应颜色直方图的 67 柄。而对于两幅图形  $A$  和  $B$ ，其各有 16 块，怎样确定  $A$  中的

某块  $b_{Ai}$  与  $B$  中的哪块进行匹配比较呢? 一般的分块直方图法中是与  $B$  中对应位置的块进行比较。这就存在一个问题, 当  $B$  中各块相对  $A$  中对应块的位置平移或旋转一段距离后, 即使原来相同的图, 匹配效果也很差。因此本文引入最大匹配法。方法如下:

①算出  $A$  中各块与  $B$  中各块的相似度, 组成相似度矩阵  $S = \{S_{i,j}\}$ ,  $i = 1, 2, \dots, 16$ ,  $j = 1, 2, \dots, 16$ 。的计算公式见下式(8)。

$$S_{i,j} = \text{Similarity}(b_{Ai}, b_{Bj}) \quad (8)$$

$b_{Ai}, b_{Bj}$  分别表示  $A$  的第  $i$  块和  $B$  的第  $j$  块。

②取出矩阵  $S$  的最大值, 设  $S_{p,q} = \max\{S_{i,j}\}$ , 存储  $S_{p,q}$  的值, 将矩阵  $S$  的第  $p$  行和第  $q$  列清零。

③重复第(2)步, 直到矩阵  $S$  全部元素为 0。

④取出存储的相似度, 取它们的平均数作为两幅图像的相似度。用公式表示如下式(9):

$$\text{Simi}_1(A, B) = \frac{1}{N} \sum_{k=1}^N \max\{\text{Simi}(b_{Ak}, E)\} \quad (9)$$

(注: 为方便排版, Similarity 简写成 Simi, 下同)

$N$  为块数,  $E$  是图像  $B$  中剩下未最大化匹配  $A$  中块的块集合。

这种方法能保证最相似的两块得到匹配, 从而在引入颜色空间分布的条件下克服图像平移、旋转等变形对检索效果的影响。

#### 4.2 颜色转移矩阵相似性度量

两幅图像, 当提取出它们各自的颜色转移矩阵后, 算出这两个矩阵的相关系数作为这两幅图像的另一种相似度。设  $D_A, D_B$  分别表示两幅图像  $A, B$  的颜色转移矩阵, 大小皆为  $k \times k$  (本文中  $k=10$ ), 则两幅图像间相似度定义如下(10)式:

$$\text{Similarity}_2(A, B) = \frac{\sum_{i=1}^k \sum_{j=1}^k (D_{A_{ij}} - \mu_A)(D_{B_{ij}} - \mu_B)}{\sqrt{(\sum_{i=1}^k \sum_{j=1}^k (D_{A_{ij}} - \mu_A)^2)(\sum_{i=1}^k \sum_{j=1}^k (D_{B_{ij}} - \mu_B)^2)}} \quad (10)$$

其中,

$$\mu_A = \frac{1}{k^2} \sum_{i=1}^k \sum_{j=1}^k D_{A_{ij}} \quad (11)$$

$$\mu_B = \frac{1}{k^2} \sum_{i=1}^k \sum_{j=1}^k D_{B_{ij}} \quad (12)$$

#### 4.3 图像的合成检索

由于分块颜色直方图与颜色转移矩阵能相互取长补短, 所以本文将这两者结合起来进行合成检索。因

两个相似度的取值范围已经是介于 0 与 1 之间, 所以不需对其再进行归一化操作。合成相似度如下(13)式:

$$\text{Simi}_{\text{last}}(A, B) = W_1 \text{Simi}_1(A, B) + W_2 \text{Simi}_2(A, B), \quad (13)$$

$$(W_1, W_2 \in [0, 1], W_1 + W_2 = 1)$$

$W_1$  和  $W_2$  的值可按实际情况灵活设置, 本文中设置  $W_1 = 0.6$ ,  $W_2 = 0.4$ 。

#### 4.4 检索流程

本文采用基于用户例图查询的工作方式, 由用户提供例图, 系统返回库中与例图计算相似度最大的前 30 幅图形。为简便起见, 图像统一放在某一特定目录下而并没放在数据库中, 图像特征提取也是实时进行的。具体流程如下:

①将例图和当前检索图像大小规格化为  $128 \times 128$  大小。

②将规格化后的图像颜色空间转换到 HSV 空间, 同时进行量化, 量化后共 67 级。

③将图像分成  $32 \times 32$  大小的块共 16 块, 得出每块的颜色直方图, 同时将这些块继续分成  $8 \times 8$  大小的小块, 提取各小块的主 C' 色, 得到整幅图形的主色矩阵。

④按上面介绍的块间最大匹配法算出两幅图像的相似度。

⑤扫描主色矩阵, 得出图像的颜色转移矩阵, 并算出两幅图像的转移矩阵相关系数, 也即  $\text{Similarity}_2$ 。

⑥合成两个相似度, 得到两幅图像总的相似度。

⑦对图像库中的每幅图像重复以上步骤, 算出每幅图像与例图的相似度。

⑧将图像按相似度的逆序排列, 返回最前面的 30 幅给用户。

### 5 试验结果及评介

为了检验本文提出方法的效果和性能, 用 MATLAB 在 SIMPLcity 集<sup>[8]</sup>的一个子集上进行了实验。SIMPLcity 测试集是从 Corel 图像库中选取的, 共包含有 10 个语义类。10 个语义类分别为: 非洲人、海滩、建筑物、公共汽车、恐龙、大象、花、马、高山和食物。其中每个语义类包含有 100 幅图像, 共计 1000 幅图像, 每幅图像大小均为  $256 \times 384$  或  $384 \times 256$ 。本文实验中在每个语义类中取前 30 幅图片, 这样测试库由 300 幅图像组成。具体测试时每种

方法每个类别中取 4 幅图像作为例图检索, 这样每种方法共进行 40 次检索, 待比较的所有 4 种方法共进行 160 次检索。本测试中只显示前 30 幅图像, 且以下各种参数计算也都只针对前 30 幅来进行。实验中比较了全局颜色直方图法、传统局部颜色直方图、本文中提出的最大匹配局部颜色直方图及本文中提出的综合最大匹配局部颜色直方图和相邻颜色转移矩阵的方法(简称综合法)共 4 种的检索效果。检索效果评测如下图 2:

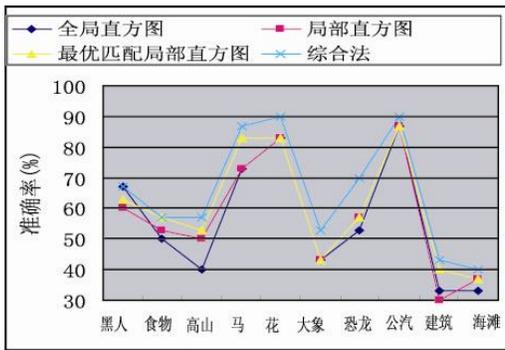


图 2 检索准确率曲线

由于各语义类别图像有很大不同, 所以查询准确率波动很大。总的说来, 最大匹配局部颜色直方图整体比一般局部颜色直方图检索效果好一些, 而将最大匹配局部颜色直方图和颜色转移矩阵结合的综合法效果又比最大匹配局部颜色直方图效果好一些。图 3 显示的是一个查询例图为花的查询结果, 因篇幅限制只截下了各方法查询结果的前 3 行即前 18 幅图像。其中 1-3 行是第一种方法得出的结果, 余此类推, 每种方法得出的查询结果的第一幅是查询例图。可以看出, 本文提出的综合法检索结果明显优于其它方法。通过多次的查询比较, 可以得出结论: 本文中提出的综合方法具有更好的检索精度, 相似图像的排序更靠前。这是由于提取了多种图像特征, 兼顾各方面的要求。当然, 另一方面, 需要处理的数据也更多, 所以本文中的综合方法速度上不如其它方法。

## 6 结论

本文提出了一种新的方法, 将彩色图像的颜色特征和空间信息特征结合起来用于彩色图像的相似性检索, 即采用最大匹配局部直方图在一定程度上提高了一般局部直方图法对图像内容的位移、旋转等的鲁棒

性, 同时采用相邻颜色转移矩阵更好的刻画图像中颜色的空间相邻关系。试验结果证明, 本文提出的方法取得了很好的效果。下一步将研究提高该方法的时间性能以及与其它图像特征的融合检索。



图 3 一次查询例图为花的检索结果

## 参考文献

- 1 周明全, 耿国华, 韦娜. 基于内容图像检索技术. 北京: 清华大学出版社, 2007.
- 2 Swain MJ, Ballard DH. Color indexing. International Journal of Computer Vision, 1991, 17(1): 11-32.
- 3 Stricker M, Orengo M. Similarity of color images. IS&T/SPIE Conf. on Storage and Retrieval for Image and Video Database 3, Vol 2420, San Jose, CA: Feb. 381-392.
- 4 王向阳, 周璐, 芦婷婷. 基于彩色边缘网格直方图的图像检索方法. 小型微型计算机系统, 2007, 9(9): 1705-1709.
- 5 王涛, 胡事民, 孙家广. 基于颜色空间的图像检索. 软件学报, 2002, 13(10): 2031-2036.
- 6 Gevers T, Smeuder AWM. Evaluating color and shape invariant image indexing of consumer photograph. In: Proceedings of the 1st International Conference on Visual Information Systems, Melbourne, Australia, 1996: 254-261.
- 7 Gevers T, Smeuder AWM. Content based image retrieval by viewpoint invariant image indexing. Image and Vision Computing, 1999, 17(7): 475-488.
- 8 <http://wang.ist.psu.edu/docs/related.shtml>.