基于内容的图像检索

傅展昇 2012013299 tarmafu@gmail.com
张凯 2012013311 nezharen@163.com
刘桐彤 2012013331 jshaltt7@163.com

摘要
利用颜色矩、颜色直方图和颜色自相关图得出图片的特征向量,利用 R 树建立图像索引,进行基于内容的图像检索。

关键字
R 树，颜色矩，颜色直方图，颜色自相关图，空间索引

1. 简介
在本次实验中，我们用颜色矩、颜色直方图、颜色自相关图分不同维度提取图片的特征向量，然后利用 C++版本的 R 树并进行部分改进后，使其可以利用特征向量建立图片的空间索引。利用批量检测测试了各种特征和维度的识别效果，并制作了图形界面以供单条查询。

这篇报告按如下方式组织：第一部分为简介；第二部分为实验概述，包括 R 树实现和各特征的定义；第三部分为实验必做部分，包括 R 树性能测试、不同特征值效果比较和；第三部分为实验选作部分，包括不同相关度度量函数的效果比较和页大小、块大小、扇区大小对 R 树性能的影响分析；第五部分为参考文献

2. 实验概述
2.1 R 树实现
本次实验中 R 树是基于 C++版本的改进。原 C++版本的 R 树每个节点的 FANOUT 数（即分支总数）是固定的，并且仅支持精确查询。

在本次实验中我们实现了 KNN 搜索算法，并支持切比雪夫距离、曼哈顿距离、欧几里得距离三种模式。此外，我们还研究了计算机预读页的大小（page size），并根据 page size 改变 FANOUT 数以提高 R 树的搜索效率。另外我们对 RTree 类进行了封装以便使用。

2.1.1 KNN 搜索算法
考虑到在实际的图片检索应用中，往往是在接收用户输入的检索图片后返回确定数量的同类图片，因此我们希望改进 R 树的搜索算法，使其可以返回最相似的 TopK 图片。

KNN 搜索算法利用了分支与界法的思想。利用两个数组：resultList 数组存放指向文件名的指针，distList 数组存放该文件与检索图片的距离，距离由小到大排布。首先将 distList 数组全设为一个不可能的最大值。从根结点开始，递归调用本算法。若当前结点为内部结点，如果 MBR 到检索图片的最小距离大于 distList 的最后一个值时，该 MBR 内不可能含 TopK 的图片，可减去该支。否则对该 MBR 的子树继续调用本算法。若当前结点为叶子结点，则将当前结点插入 resultList 和 distList 数组中。通过维护这两个数组可以实现 TopK 的准确检索。

2.1.2 相关性度量
我们定义了 CHEBYSHEV_DISTANCE、MANHATTAN_DISTANCE 、EUCLID_DISTANCE 三个宏，表示实验中分别利用切比雪夫距离、曼哈顿距离和欧几里得距离。效率比较见第四部分。

2.1.3 Page Size 影响
原 C++版本的 R 树的 FANOUT 数是确定的，这是不合理的。当维度不同时，同一节点的大小会发生改变。因此为了保证预读的效率足够高，应根据 page size 和维度来确定 FANOUT 的值。具体见第四部分，

2.1.4 R 树的封装
原 C++版本的 R 树为了通用性使用了模板类型，但是对于本例来说不够方便。因此我们构造了 RTreeApplicaition 类对 RTree 类进行了封装，以便于本次实验利用图片目录和图片特征构建 R 树，提供了直接从文件构建 R 树和 KNN 搜索的成员函数。进一步的，为了方便地接收来自图形界面的请求（例如，输入图片总数、特征名称、维度数、检索图片路径，输出检索结果），我们再次对 RTreeApplicaition 类进行封装，为 RTreeAppTest 类，从而可以仅仅通过输入图片总数、特征名称、维度数、检索图片路径等对用户很友好的参量即可进行图片检索。

2.2 图像特征
本次实验中我们共利用了颜色矩、颜色直方图和颜色自相关性三种特征进行图片特征提取。颜色直方图和颜色矩都使用到了 HSV 色彩空间。HSV 即色相、饱和度、明度（Hue, Saturation, Value）。设max等价于r, g和b中的最大者。设min等于这些值中的最小者，则 HSV 空间在数学上定义为在 RGB 空间中的颜色的 R, G 和 B 的坐标变换。具体定义如下

$$h = \begin{cases} 0^\circ & \text{if } max = min \\ 60^\circ \times \frac{g-b}{max-min} + 0^\circ & \text{if } max = r \text{ and } g \geq b \\ 60^\circ \times \frac{g-b}{max-min} + 360^\circ & \text{if } max = r \text{ and } g < b \\ 60^\circ \times \frac{b-r}{max-min} + 120^\circ & \text{if } max = g \\ 60^\circ \times \frac{r-g}{max-min} + 240^\circ & \text{if } max = b \end{cases}$$

$$s = max - min \qquad v = max$$

对于本系统的 HSV 各分量，我们调用了 Qt 中 QColor 类下的 hsvHue(), hsvSaturation()和 value()三个成员方法直接获取。

2.2.1 颜色矩（Color Moment）

记p_{ij}为第i个色彩分量的值，N 为图像中的所有点。对于有 3 个色彩分量的 HSV 空间，可用如下方法定义 9 维特征向量，即为颜色矩：

$$E_i = \frac{1}{N}\sum_{j=1}^{N} p_{ij}, \sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(p_{ij}-E_i)^2}, s_i = \sqrt[3]{\frac{1}{N}\sum_{j=1}^{N}(p_{ij}-E_i)^3}$$

2.2.2 颜色直方图（Color Histogram）

$$a = h/(360/d_1)$$
$$b = s/(256/d_2)$$
$$c = v/(256/d_3)$$

此点的在一个三维空间中可表示为(a, b, c)，若定义$i = a\times d_2\times d_3 + b\times d_3 + c$，则直方图可表示为$d_1\times d_2\times d_3$维向量。数

2.2.3 颜色自相关图（Color Correlogram）

颜色相关图是指对于颜色值为 C_i 的像素点 P_1，与 P_1 相距距离为 k 的另一个像素点 P_2 的颜色为 C_j 的概率的大小。其数学定义为：设一个合适的距离 k，对于图像的 N 种颜色值，C_i 表示第 i 种颜色值。定义图像 I 的颜色相关图为：

$$\gamma_{C_i,C_j}^{(k)}(I) = \Pr[p_2 \in I_{C_j}, |p_1 - p_2| = k, p_1 \in I_{Ci}]$$

颜色自相关图是指当上式中 C_i=C_j 时，即仅考虑一个与像素点相距为 k 的像素点的颜色与其相同的概率。这样，我们便可以得到一个基于颜色值的一维向量，从而可以用于 R-Tree 的建树及查询。对于图像的颜色，由于每种颜色都含有 R、G、B 三个分量，因此为了实现将三个分量统一为一个分量，我们使用了灰度来作为颜色自相关图的颜色值。对于本系统的灰度值，我们调用了 Qt 提供的 qGray 函数直接根据 R、G、B 分量计算出了灰度值，灰度值的计算公式为 gray = (R * 11 + G * 16 + B * 5)/32.

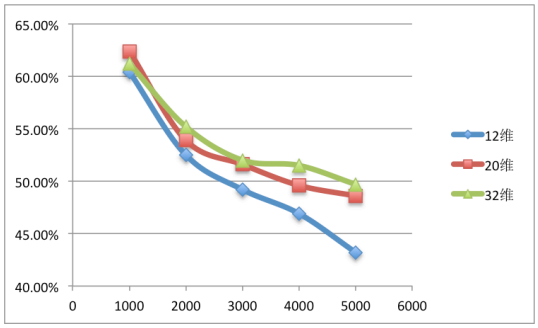
3. 必做实验

3.1 任务 1：R 树的性能测试

本次测试中 R 树的最大 MBR 数为 8，最小 MBR 数为 4，相关性度量采用欧几里得距离。取颜色直方图的特征向量，针对 12 维、20 维、36 维的特征向量，在插入对象数分别取 1000 到 5000，增量为 1000 时，测试求 10 近邻时平均检索访问的磁盘数占磁盘总数的比例(检测次数等于对象数取均值).

对象数	1000	2000	3000	4000	5000
12 维	60.4%	52.5%	49.2%	46.9%	43.2%
20 维	62.4%	53.9%	58.6%	49.6%	48.6%
32 维	61.2%	55.2%	52.0%	52.5%	49.7%

作图像如下：



实验结果分析如下：

除维度和 R 树对象数外，其余变量均保持一定。1.在维度一定时，R 树本身所含的对象数越多，平均检索访问的磁盘数所占比例越低；2.在 R 树包含的对象树一定时，维度越高，平均检索访问的磁盘数所占比例越高，并且这种差别随对象数增多而变得显著。

分析以上两种现象的成因：1.维度一定时，若 R 树所含对象书越多，样本空间越大，特征向量的分布约密集，R 树层数越大，MBR 对空间的划分会更细，所以检索时利用 KNN 搜索时剪枝会越频繁，访问磁盘数的比例降低；2.对象数一定时，若维度越高，样本空间维度增大，总的空间体积快速增长，MBR 对空间的划分效率降低，导致 KNN 剪枝效率下降，访问磁盘的比例增大。

结合上述结论，结果准确度有保证的情况下，在 R 树对象数越大、维度越小时 R 树访问磁盘数所占比例越小，性能越好.

3.2 任务 2：特征的效果测试

本次测试中 R 树的最大 MBR 数为 8，最小 MBR 数为 4，相关性度量采用欧几里得距离，R 树包含对象数为 5000。测试不同的特征提取方法的效率。比较方法为对每张图片搜索 10 近邻，通过文件头判断是否正确，比较 5000 次的平均正确率.

特征名称_维度	平均正确率
ColorMoment_9	35.56%
ColorHistogram_12	34.09%
ColorHistogram_20	36.35%
ColorHistogram_32	36.71%
ColorCorrelogram_8	30.54%
ColorCorrelogram_16	30.87%
ColorCorrelogram_64	30.41%

不同特征提取方法的效果分析如下：

在取相近维度时，颜色矩和颜色直方图的正确率相近，高于颜色自相关图的正确率。随着维度提升，正确率均呈上升趋势，但是正确率都无法超过 40%。

分析以上现象的成因：颜色矩和颜色直方图利用的是图片所包含的颜色的分布，利用的是像素颜色分布的特征，颜色自相关图利用的是相似像素点的空间分布，利用的是像素空间分布的特征。以上结果表明，在提供的图片库中做检索时，像素颜色分布的关联性要优于像素空间分布的关联性。但是

这三种方式都只考虑了图片的某一个小方面的性质，不够全面，而图片是否相关的判断标准是内容是否相关，而图片内容是三种特征提取都无法考虑全面的，所以正确率都比较低.

3.3 任务 3：特征对相关性排序的影响

考虑特征对相关性排序的影响，也就是考虑特征对近邻搜索结果正确率的影响。对本次测试中 R 树的最大 MBR 数为 8，最小 MBR 数为 4， R 树包含对象数为 5000。比较不同近邻数条件下图片检索的正确性

特征名称	维度	10	20	50
ColorMoment	9	35.56%	30.07%	25.38%
ColorHistogram	12	34.09%	28.36%	23.50%
	20	36.35%	30.34%	25.26%
	32	36.71%	31.04%	25.87%
ColorCorrelogram	8	30.54%	25.64%	21.66%
	16	30.87%	25.69%	21.49%
	64	30.41%	25.04%	21.01%

特征队相关性排序的影响分析如下：

对于同一特征提取方法，1.在搜索的近邻数一定的条件下，维度越高，正确率越高，但是提高不显著。2.在维度一定的条件下，搜索的紧邻数越多，正确率越低，并且降低显著。

分析以上两点现象的成因；1.在搜索的近邻数一定的条件下，维度越高，对图像的识别和区分约精细，正确率越高，但是正确率受特征本身的限制更大，仅靠维度提高，正确率难以显著提高；2.在维度一定的条件下，特征能够区分出的同类图片有限，在较远的距离上，区分效果下降，因此近邻数增多时正确率显著下降。

结合以上两个任务的结论分析，为了 R 树搜索得到较好的结果，应使用识别度较好的特征，采用合适的维度（为了不明显提高磁盘访问的比例），在近邻数较少时得到的结果越可靠。

4. 选作实验

4.1 任务 c:不同相关性度量函数效果测试

在本次实验中，我们共实现了切比雪夫距离、曼哈顿距离和欧几里得距离。现比较不同距离定义下的效率。本次测试中 R 树的每个节点最大 MBR 数为 8，最小 MBR 数为 4， R 树包含对象数为 5000，比较在相近的维度下，不同特征提取方法的 10 近邻搜索的正确性如下：

特征名称_维度	CHEBYSHEV	MANHATTAN	EUCLID
ColorMoment_9	34.86%	35.31%	35.56%
ColorHistogram_12	33.70%	34.12%	34.09%
ColorCorrelogram_8	29.67%	31.41%	30.54%

不同相关性度量函数效果测试结果分析如下：

颜色矩：CHEBYSHEV <<MANHATTAN=EUCLID

颜色直方图：CHEBYSHEV << EUCLID=MANHATTAN

颜色自相关图：CHEBYSHEV << EUCLID<<MANHATTAN

分析以上现象的成因：切比雪夫距离指关注某一维度上向量的差别，导致该距离不能反映其他维度上的分别，因此效果明显差于另外两者。曼哈顿距离和欧几里得距离对于颜色矩和颜色直方图的效果几乎相同，而对于颜色自相关图，曼哈顿距离明显优于欧几里得距离。可能的原因是曼哈顿距离对于空间分布特征的效果比欧几里得距离更好。

4.2 任务 e:Page Size 对检索效率的影响

原 C++版本的 R 树种每个节点的最大 MBR 默认为 8，最小 MBR 数默认为 4。而根据 Branch 结构体的定义可知，不同维度下，Branch 的大小不同，因此 Node 包含的 Branch 数应与维度相关。所以我们对此进行了研究。

首先我们了解了系统扇区大小（Sector Size）、块大小（Block Size）和页大小（Page Size）的概念。Sector size 是针对磁盘的物理属性，Block size 是针对文件系统的属性。而 Page size 是磁盘读取时预读区的大小，可见访问的磁盘数是与 Page size 相关的。然后通过查询我们知道一般计算机 Page size 为 4K，所以结合 Node 和 Branch 的定义，在维度为 NUMDIMS 时，每个节点的最大 MBR 数为

(PAGESIZE – 2 * sizeof(int)) / (sizeof(double)*2*NUMDIMS + sizeof(int*))

并且通过测试证实了按照该定义，一般情形下磁盘读取效率较高。

但是有特殊情况。在维度较高时，如本例的 64 维中，由于 Branch 的 size 很大，导致如果每次只读一个磁盘块，实际每个节点的最大 MBR 数很小，这样 R 树的效果反而较差。针对这种情况我们的设想是可以使每个 Node 的 size 等于 2 个或更多个 Page 的大小。这样虽然每次读取多个 Page，但是由于 R 树每个节点的最大 MBR 数增大且对 Page 利用率很高，读取效率依然较高。

4.3 图形界面

利用 Qt 实现。操作流程详见文件夹中 ReadMe 文件。

5. 参考文献

1. Markus A. Stricker, Markus Orengo. 1995. Similarity of color images. Proc. SPIE 2420, Storage and Retrieval for Image and Video Databases III, 381.
2. 王娟，孙兵，贾巧丽. 基于图像颜色特征的图像检索技术[N]. 计算机系统应用, 2011 年(第 20 卷第 7 期).