

基于内容搜索的图像索引

谭新宇

2016010649

周泽龙

2016013231

刘昀瑞

2016011134

实验大纲

摘要：本次实验中我们小组测试了 RTree 高维索引的效率，提取了数据集中所有图的多种形式若干维度的特征向量，还以 RTree 作为高维索引结构实现了对于输入图的高效率查找相关图的实验要求，并对不同特征不同维度的特征索引结果进行了分析。最后我们对于 RTree 优化提供了自己的一点思路。

关键字：RTree 特征向量 高维索引 正确率 命中率

实验观察及分析

任务一

由于第一个任务是为了检验 R 树检索效率，故在此处分别随机生成 5000 个 n (4,8,12,16,20) 维数据作为提取特征值的维数，分别检验插入 1000,2000,3000,4000,5000 个数据时访问磁盘次数的平均值。（注：以下表格中所得数据为随机在数据库中选取 10000 组插入点划定随机范围求得的磁盘访问次数的平均值）

范围约定：对于一个点，查询其所有维数坐标的 0.5 倍至 1.5 倍。

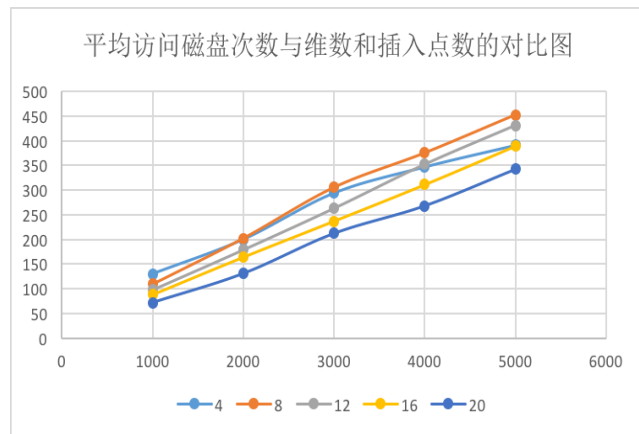
	1000	2000	3000	4000	5000
4	130	200	294	347	391
8	109	202	305	375	452
12	98	179	263	352	431
16	88	164	237	311	389
20	72	131	212	268	342

当维数一定时，随着 RTree 中点的数量线性增加，访问磁盘次数也近似线性增加。这符合 RTree 的构造原理，当插入的点的数量线性增加时，其查询范围所形成的 n 维体积（ n 已固定）中的叶子节点也近似线性增加，从而最终呈现如此的效果。

当插入点数目一定时，随着 RTree 中维数的增加，可以看出平均访问磁盘次数降低，似乎维数越高 RTree 的效率越好，那么 RTree 真是一种完美的数据结构了，越高维效率越高。实际上却不是这样，出现这种情况主要与我们查询范围的约定有关，由于我们的范围是对于一个点，查询其所有维数坐标的 0.5 倍至 1.5 倍。所以随着维数的增加，查询范围占总范围的占比实际上越来越小了，从而导致了这样的情况出现。为了验证我的猜想，接下来我对于查询范围作了一定更改，将范围约定为对于一个点，其每一个维数的坐标加一个小于 4000 的随机数为其最大值，减一个小于 4000 的随机数为其最小值，每个维度坐标之间无关联，形成最终的查询范围。请看下图分析。

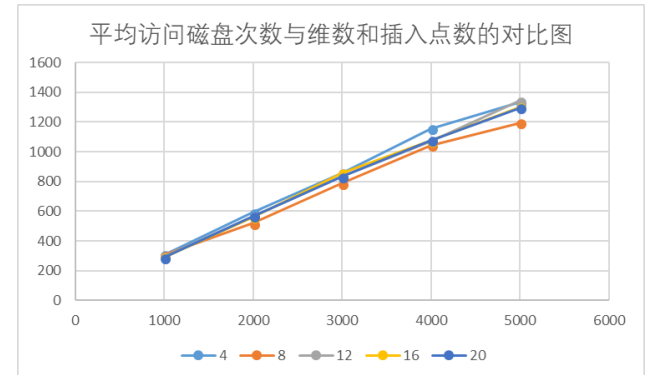
范围约定：对于一个点，其每一个维数的坐标加一个小于 4000 的随机数为其最大值，减一个小于 4000 的随机数为其最小值，每个维度坐标之间无关联，形成最终的查询范围。

	1000	2000	3000	4000	5000
4	308	592	860	1155	1332
8	307	520	790	1044	1195
12	299	567	837	1071	1344
16	292	562	859	1075	1303
20	290	566	834	1077	1297



分析：

既然我们约定以查询过程中的磁盘访问次数来衡量 RTree 的查询效率，则由此图可以较为直观地发现：



分析：

接上处分析，我们还是从两个方向考虑：当维数一定时，我们能得到与上处分析同样的结论：随着 RTree 中点的数量线性增加，访问磁盘次数也近似线性增加，由于原因相同，此处不再赘述。

当插入点数目一定时，随着维数的增加，我们可以看到，这张图中的平均磁盘访问次数在误差范围内没有发生明显变化，这也就印证了我之前的猜想。由于这一次我约定的查询范围并不出和上次约定范围一样出现高维占比减小的情况，各个维度坐标之前都是无关的，所以平均磁盘访问次数近似相等。

总结：

此前谈到 RTree 并不像第一张图展现的那样好像具有维度越高效率越高的完美特性，但从第二张图来看，RTree 的确是一种很好的高维索引结构，其能够在维度增加时保留几乎相同查询效率的优良特性。

此外由于随机生成的数据不能检验其有效性，故此处不再检验。

注：本实验中的 RTree 修改自资料包中提供的 RTree.h 头文件。

任务二

RGB Histogram:

基于 RGB 色彩空间的直方图，分为 RGB 三个通道，并将三个分量都划分为 N 个区间，然后统计各个区间内的像素数。例如：将 RGB 三个分量都分为 4 个区间（0-63，64-127，128-191，192-255），再统计各个区间的像素点数，最后以各个区间的像素数构成特征向量。

HSV Histogram:

色调 H，用角度度量，取值范围 $0^{\circ} \sim 360^{\circ}$ ；饱和度 S，表示颜色接近光谱色的程度，取值范围 $0\% \sim 100\%$ ；明度 V，表示颜色的明亮程度，取值范围 $0\% \sim 100\%$ 。HSV 可由 RGB 转化而来，转换公式为：

$$h = \begin{cases} 0^{\circ} & \text{if } \max = \min \\ 60^{\circ} \times \frac{g-b}{\max-\min} + 0^{\circ}, & \text{if } \max = r \text{ and } g \geq b \\ 60^{\circ} \times \frac{g-b}{\max-\min} + 360^{\circ}, & \text{if } \max = r \text{ and } g < b \\ 60^{\circ} \times \frac{b-r}{\max-\min} + 120^{\circ}, & \text{if } \max = g \\ 60^{\circ} \times \frac{r-g}{\max-\min} + 240^{\circ}, & \text{if } \max = b \end{cases}$$
$$s = \begin{cases} 0, & \text{if } \max = 0 \\ \frac{\max-\min}{\max} = 1 - \frac{\min}{\max}, & \text{otherwise} \end{cases}$$
$$v = \max$$

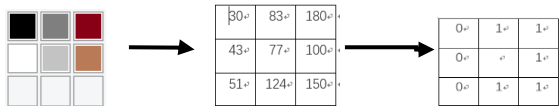
按照上述转换公式，求得每个像素点的 HSV 向量，然后分为 H、S、V 三个通道，将其进行划分，按照人的视觉分辨能力，将 H 分为 8 份，S 和 V 分为 3 份，然后统计各个区间内的像素数，以此构造 HSV 的特征向量。

LBP Feature:

LBP，是一种用来描述图像局部纹理特征的算子，具有旋转不变性和灰度不变性等特点。本次 LBP 特征提取算法定义在 3×3 的窗口内，以中心像素点灰度值为阈值，将周围 8 个像素的灰度值与其比较，若周围灰度值大于阈值，标记为

1，否则为 0。这样，就可产生 8 位二进制数，转化为十进制便得到中心像素点的 LBP 值，共 256 种。

以此算出每个像素点的 LBP 值，然后均分图像，例如划分为 4×4 的区间，每个区间统计 256 种 LBP 值的像素数，以此构成 LBP 特征向量。



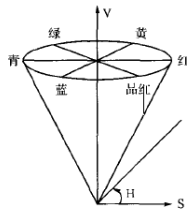
$(01111000)_{10} = 124$ （上述数值只是为了方便理解，并非真实的数值）

其中灰度值计算采用浮点算法： $\text{Gray} = R \times 0.3 + G \times 0.59 + B \times 0.11$

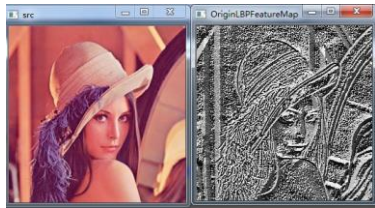
Comparison:

RGB 颜色模型是图像处理中最为基础的模型，但是 RGB 颜色空间并不直观，其次，RGB 颜色空间是不均匀的，比如，距离为 50 的 $(0, 0, 0)$ 和 $(50, 0, 0)$ 两种颜色对于人眼视觉而言都是黑色，而距离同样为 50 的 $(100, 50, 100)$ 和 $(100, 100, 100)$ 两种颜色对于人眼而言就是紫色和灰色两种不同的颜色。

而 HSV 颜色空间则较好的符合了人眼的颜色感知的三个要素：色调、饱和度和亮度，HSV 模型如图所示。HSV 颜色空间有两个特点：① 亮度分量与图像的彩色信息无关；② 色调与饱和度分量与人感受颜色的方式紧密相连。



而 LBP 特征，则不太关注图像的颜色色彩，而是关注局部特征，或者说是局部纹理。它关注的是第一像素点与其周围像素点的对比，由此分析出图像的纹理特征。下图为网上的一个 LBP 测试结果。由此可见，LBP 在人脸识别上、物体识别上应该会有极佳的表现。



任务三

操作流程：

为了方便处理，我们首先对所有读入的向量进行归一化。这样子下来我们第一步划定大的范围时将每个维数的最小范围置为 0，最大范围置为其坐标值的 5 倍。然后对该范围进行查询，将查询到的数据存到一个列表里，再对这个列

表里的所有点遍历分析，通过巴氏距离法，欧式距离法或直方图相似法与查询点进行比较，将偏差较大的点去掉。剩下的这个列表就是我们返回的最终有序列表。以下是我们测试结果。

定义：

正确率：返回的列表中正确的个数/返回列表的总个数；

命中率：返回的列表中正确的个数/数据库中同类别的总个数。

匹配方法比较：（以下皆为测试多次一百组数据得到的平均结果）

助教提供的 9 维颜色矩特征向量：

	正确率	命中率
巴氏距离	11%	92%
欧式距离	13%	59%
直方图相似	12%	85%

自己提取的 27 维 RGB 特征向量：

	正确率	命中率
巴氏距离	20%	5%
欧式距离	30%	1.5%
直方图相似	30%	2.5%

自己提取的 64 维 RGB 特征向量

	正确率	命中率
巴氏距离	20%	2%
欧式距离	39%	0.6%
直方图相似	50%	0.6%

自己提取的 64 维 LBP 特征向量：

	正确率	命中率
巴氏距离	15%	68%
欧式距离	46%	8%
直方图相似	15%	70%

自己提取的 72 维 HSV 特征向量：

	正确率	命中率
巴氏距离	56%	1%
欧式距离	47%	1%
直方图相似	50%	1%

结果：

- (1) 由②、③组数据可知，RGB 特征向量维数越高，正确率越高，然而命中率会降低。

- (2) 由③、④、⑤组数据可知，在这几种特征中，正确率与命中率成负相关关系，二者很难兼得。

- (3) 综合下来，64 维 LBP 特征向量中的欧氏距离法表现最佳，最终提交的源码中也是用此方法索引。

分析：

- (1) 特征向量维数越高，筛选、排错能力越强，这样得出的结果中正确结果的比例就更大。然而，由此带来的缺点就是，被排除在结果之外的正确结果会更多，因此，命中率就下降。
- (2) 正确率与命中率成负相关的原因，便是因为为了提高正确率，特征向量的匹配算法就需要具备更高的筛选能力，结果图片的数量便会减少，由此带来的结果便是获得正确图片数量在总的正确图片数量中的比例会更低，也就是命中率下降。

展望：

要想同时提升命中率和正确率，应该构建一个神经网络来训练数据集或者提取更多局部特征，这样子才有希望将两者同时提升！希望下一个大作业我们小组能够实现。

任务四：（e）R 树优化

- (1) 在 R 树叶节点中存储适当的定位数据，从而使索引结果集中的位置信息可以用来帮助建立读取聚集的结果集读取序列。
- (2) 在遍历 R 树时剪枝：在 R 树中，一个树结点的最小边矩形总是包围着它的子孙结点的最小边矩形。一对结点需要进行连接仅当它们父结点的最小边矩形相交，称为剪枝。简单的从顶至底的图遍历算法可以在任何层次上使用这种剪枝。此处剪枝是在同时深度或广度优先遍历两棵输入的 R 树时进行的。相比简单的嵌套循环的遍历方式剪枝将使得被遍历的不相交的结点对数大大减少。

参考文献：

[1]俞浩,孙燕.基于颜色矩和形状不变矩的图像搜索研究[J].电脑知识与技术,2015,11(19):174-175.

[2]张笃振.基于颜色特征与 LBP 的图像检索算法研究[J].微计算机应用,2009,30(06):35-38.

[3]吴成玉. 基于颜色特征的图像检索方法与实现[A]. 中国计算机学会数据库专业委员会.第二十一届中国数据库学术会议论文集（研究报告篇）[C].中国计算机学会数据库专业委员会.;2004:5.

[4]基于树查询的性能优化 胡茂胜 刘书良 左泽军 中国地质大学信息工程学院 湖北武汉

[5]基于 R 树的空间查询连接处理优化与实现 吕闯晖 1,吕敏蓉 2 (1.海军工程大学 装备经济研究所,湖北 武汉 430033; 2.湖南女子学院,湖南 长沙 410004)