

Experiment Project: Indexing Images for Content Based Retrieval

刘家维

软件 61 2016013246

chiawei.liu.1201@gmail.com

曾铮

软件 62 2016013263

hasbeenoccupied@126.com

摘要

此实验研究 R Tree 的搜索效率与准确度在不同特征与维数下的变化。搜索数据为 5613 张给定的图片。R Tree 的数据结构引用老师提供的现成 packages，并在其中稍作修改以符合此次实验需求。特征方面透过 Matlab 对所有图片进行了处理，提取了 4 到 16 维的图片灰度特征向量，以及 9 维的 hsv moment 与 9、16 维的 gray singular，搭配老师提供的 color moment 特征一并进行实验比较。实验后发现，维度增大时，搜索效率振荡下降但逐渐趋于稳定；数据量与搜索次数呈线性正相关；实验所用到的特征中，gray singular 的搜索效率明显高于其他特征，而经处理的 gray hist 则有着远高于其他特征的搜索准确率。

1. 概述

R Tree 是一种带有渐进求精思想的索引数据结构，只要将数据转换成多维向量型式，就可以构建多维空间数据点，并将数据点层层组合建成 R Tree。图片搜索是数据搜索的一个分支，如何将像素点的信息有效压缩提取成较低维的向量需要多方尝试。本次实验就针对几种不同的图片特征与不同维度的特征，进行比较，尝试观察 R Tree 在这些情况下的搜索效率与准确度。

2. 实验环境

OS: Windows 10.

Compiler: C/C++ Optimizing Compiler Version 19.11.25547 for x86

Matlab R2018a

3. 项目文件解释

3.1 Matlab 文件说明

3.1.1 ./matlab/loadimage.m

将原始图片转化为灰度图并保存。读入灰度图，将得到的灰度矩阵的直方图信息存入 gray_hist.txt。

3.1.2 ./matlab/hsv_moment.m

将原始图片的 RGB 值转化为 HSV 值，计算其中心矩，得到的 hsv 颜色矩向量保存在 hsv_moment.txt 中。

3.1.3 ./matlab/gray_hist_dimension.m

将 gray_hist.txt 中存储的数据降维输出到 gray_hist_4dim.txt, 5dim.txt, 6dim.txt, etc.

3.1.4 ./matlab/singular_value.m

对每一张灰度图的灰度矩阵进行奇异值分解，取前 16 大的奇异值为特征向量，按 ∞ -范数归一化后存入 gray_singular.txt 中。

3.1.5 ./matlab/graySingular.txt, ./matlab/graySingular9dim.txt

保存主成分分析降维后的 singular 向量。

3.2 源码说明

3.2.1 ./src/RTree.h

RTree.h 引用自老师所提供 Greg Douglas 的 C++ 版本 R Tree，此 R Tree 的相关信息可见 ./src/README.TXT。本次作业对 RTree.h 的改动不多，主要是将搜索命中次数改写为节点访问次数，具体逻辑参考了 Antonin Guttman 的 C 版本的 RTree 中的 index.c 中的 SeTouchCount1。另外还在回调函数上做了更动，回调函数传入 vector *ids，用于存取单次搜索获得的所有结果的 id。

3.2.2 ./src/query.cpp

此文件为此次实验运行的入口与主体，设计了 interface，方便输入搜索与数据信息，调用 RTree.h 中的数据结构与功能，搜索后将结果打印，同时也输出成外部文件，便于实验操作与结果统计。特征的维数不能在 interface 中输入，如要改动必须在 query.cpp 中改 NUMDIMS，并重新编译产生 exe 文件。

3.2.3 ./exe/input

存放输入用的特征文件、搜索文件。

3.2.4 ./exe/output

存放输出的实验结果。输出的文件路径与名称不能在 interface 中决定，需在 query.cpp 中修改，当前默认输出到 output/result.txt 中。

3.2.5 编译

如果想更改维度或输出文件名称，则需对 query.cpp 代码进行修改，修改后需重新编译。使用 MSVC 编译器对 query.cpp 编译即可得 exe 执行程序。将 query.exe 放置到 ./exe 中，双击可运行。

¹ Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In ACM SIGMOD International Conference on Management of Data, pages 47–57, 1984.

4. 实验方式

4.1 特征提取

4.1.1 灰度直方图(Gray-level Histogram)

将图像的 R、G、B 值通过 $0.2989R + 0.5870G + 0.1140B$ 的加权平均转化为灰度值，取区间长度为 16，以图像的灰度直方图作为图像的特征，特征维数为 16 维。

对该特征进行主成分分析(Principal Component Analysis): 将所有 5613 张图片的向量放入同一个矩阵 $A_{(5613 \times 16)}$ 中，对 A 进行奇异值分解(Singular-Value Decomposition³), 得 $A = U \Sigma V^T$ 。取 $A_{(5613 \times i)} = U_{(5613 \times i)} S_{(i \times i)}$, $i=4,5,\dots,16$, 以实现 对原 16 维数据的降维，结果存放在 gray_hist_4dim.txt、……、gray_hist_16dim.txt 中。

4.1.2 灰度矩阵奇异值

对图像的灰度矩阵 A 作 SVD 分解，取前 i 大的奇异值($i=9,16$) 为图片的 i 维特征向量。

4.1.3 HSV 颜色矩

将图像各像素的 RGB 表示转化为 HSV(色调、饱和度、明度) 表示，对每一张图片，求出所有像素的 h 、 s 、 v 值的一、二、三阶中心矩(期望、方差、斜度)，得到一个 9 维的特征向量。

4.2 程序运行方式

运行 ./exe/query.exe，使用者/实验者可接连输入：

4.2.1 query file

内含欲搜索的图片的名称，每行一个图片名称，程序将读入每行直到文件空。

输入例: input/queryAllImage.txt。将 5613 张图片都搜索一次。

4.2.2 feature file

内含多行特征向量，空格隔开。

输入例: input/gray_hist_8dim.txt。读入 8 维灰度特征向量。

4.2.3 data amount

想要在 feature file 中取前多少行读入建造数据库(RTree)。实验者必须确保此数量足以覆盖所有 query file 中出现的图片名，否则一旦搜索不存在数据库中的图片名，程序运行将无法终止。

输入例: 5613。将 5613 行特征向量都读入建树。

4.2.4 range

将搜索的图片从数据点改为数据范围。向量的每一维都加減 range，使其成为一范围，搜索到的多个数据点(多张图片)就会是那些落在此范围的数据点(图片)。Range 为 0 的话就是精确搜索。

输入例: 500。要搜索数据点变为以及为中心，各为 ± 500 的范围。

搜索完毕后，会给出以下结果：

4.2.5 Search Amount

总共搜索了几张图片(即 query file 中的行数)

4.2.6 Average touched times per search

平均每次搜索访问了几个节点(评估效率的基准)

4.2.7 Average results found per search

平均每次搜索获得了几个结果

4.2.8 Average same kind rate per search

平均每次搜索有多少比例的图片为同类型(1 即 100%)

输出的结果会同步输出到 ./exe/output/result.txt。

4.3 程序执行流程与逻辑

程序一开始会先根据 /input/imagelist.txt 建造双向的图片名称索引，也就是 id 到图名以及图名到 id。接着根据 feature file 的内容建树，同时也将所有特征向量存到一二维数组中。之后依序读入 query file 中的图名，将图名对应到 id，再利用 id 对应到二维数组中该图片的特征向量(因为此次实验中搜索图片都是数据库中的某一图片，故必能获取该图的特征向量)。得到要搜索的图片的特征向量(数据点)后，用 range 对其扩展成一个数据范围(若 range=0 则还是数据点)，然后对 tree 调用搜索函数，搜索此数据范围。因为我们对 RTree.h 做了些修改(见 3.2.1)，现在 search 回传的值为节点访问次数，同时，传入的参数 ids 则已经存放了所有搜索到的结果(落在数据范围内的数据点的 id)。将搜到的 id 透过 kind() 函数来判断其图片类型(事先知道的结果)，便能统计出准确率(即搜索到的图片有多少比例与原搜索的图片是同一类型)。

touchCountSum, resultAmount, sameKindRateSum 都是用来统计总量，最后计算平均值(多个 query 后的平均)用的。

5. 实验结果与分析

5.1 Problem 1 之不同维度的特征对搜索效率的影响

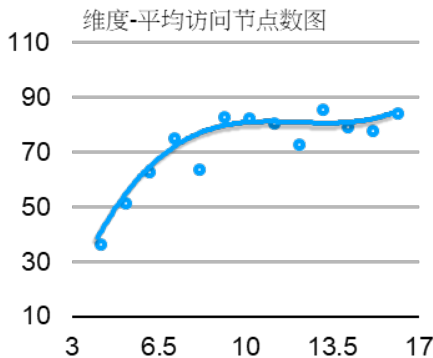
5.1.1

原始结果见 ./exe/output/differentDimensionResult.txt

² <https://ww2.mathworks.cn/help/matlab/ref/rgb2gray.html>

³ Dan Kalman : A Singularly Valuable Decomposition: The SVD of a Matrix

5.1.2 图表



5.1.3 分析

取 Range 为 0, 以 4~16 维的灰度直方图向量为特征, 节点数为 5613(最大节点数)时, 维数与平均访问节点数如图所示。

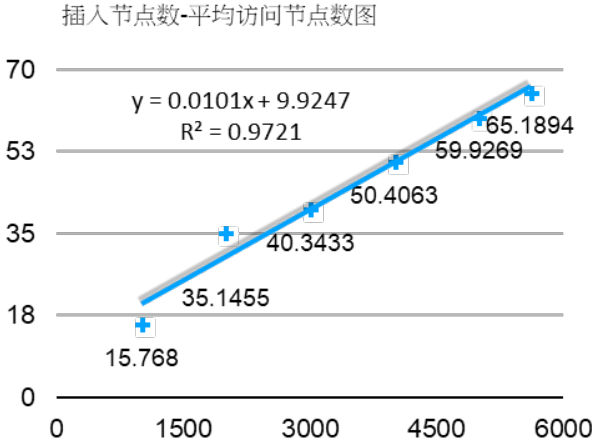
平均访问节点数总体呈现上升趋势, 但存在一定的震荡, 预计最终随着维数的增长, 平均访问节点数可能会稳定在某个值附近。

5.2 Problem 1 之不同数据量对搜索效率的影响

5.2.1

原始结果见./exe/output/differentDataAmountResult.txt

5.2.2 图表



5.2.3 分析

取 Range 为 0, 以 8 维灰度直方图向量为特征, 节点数为 1000、2000、3000、4000、5000、5613, 平均节点访问数如上。

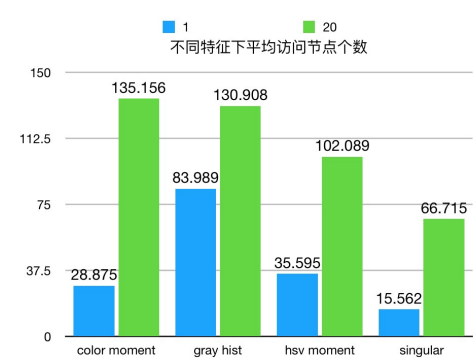
用最小二乘法对结果进行线性拟合, $R^2=0.9721$, 线性度较高, 维数与平均访问节点数总体呈线性增长。

5.3 Problem 2 不同特征对搜索效率的影响

5.3.1

原始结果见./exe/output/differentFeatureTypesResult.txt

5.3.2 图表



5.3.3 分析

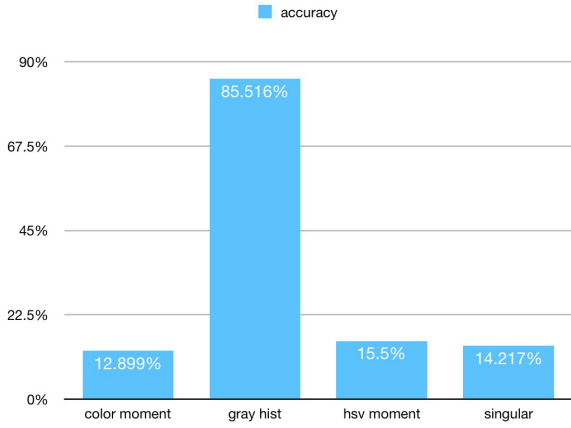
可知在点查找下, 灰度直方图向量(gray hist)的访问效率最为低下, 而范围查找中则颜色矩(color moment)的访问效率同灰度直方相近, 都是最差的。范围查找下, hsv 颜色矩的访问效率相对较高于前两种特征, 而奇异值向量的访问效率始终远优于其他特征。说明奇异值向量特征可以很好的把单张图像相互区别开来, 而灰度直方在这方面的表现是最差的。

5.4 Problem 3 不同特征对搜索准确率的影响

5.4.1

原始结果见./exe/output/differentFeatureTypeRateResult.txt

5.4.2 图表



5.4.3 分析

取维数为 9, 插入节点数 5613, 调整 Range 使平均搜索结果数在 20 附近。搜索结果中同类型图片看作正确结果, 显而易见的是, gray hist 的准确率高达 85%, 远远高于其他特征, 其他几组特征的准确率都在 12~16%以内。可见, 对于实验中的数据, 灰度直方图特征比较准确地反映了不同种类图片的本质区别。

6. 参考文献与参考资料

- [1] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In ACM SIGMOD International Conference on Management of Data, pages 47–57, 1984.

[2] <https://ww2.mathworks.cn/help/matlab/ref/rgb2gray.html>

[3] Dan Kalman : A Singularly Valuable Decomposition: The SVD of a Matrix