

数据结构图像检索作业

陈超
图片索引与搜索
软件工程 52
2015013222
86+ 13621044823
1505690897@qq.com

侯建国
提取图像特征
软件工程 51
2015013212
86+ 13051338813
thss15_houjg@163.com

李在弦
可视化
软件工程 52
2015080121
86+ 15510380063
15510380063@163.com

摘要

本文的核心内容是图像检索。我们介绍特征提取的方法(HSV 颜色矩和轮廓中心矩)，R 树搜索，可视化，和相关信息。

关键词

R 树;图像索引与搜索; 特征提取; 图形界面

1. 引言

本次作业主要核心的图像检索的重点在于特征提取，相似图像的匹配，和多维数据的处理。以下的内容是关于实现方法，分析，实现结果。

2. 实验环境

操作系统: Windows 10

IDE: Visual Studio 2015/Visual Studio 2012

编程语言: C++, MFC, openCV

3. R 树搜索

特征数据量 ^o	图片数据量 ^o	R 树节点访问量 ^o	平均查询结果 ^o	特征有效性 ^o
6 ^o	1000 ^o	9937317006 ^o	3.36812e+006 ^o	79.0319% ^o
6 ^o	2000 ^o	86099242810 ^o	1.49856e+007 ^o	61.5348% ^o
6 ^o	4000 ^o	1193490596064 ^o	5.91743e+007 ^o	35.8786% ^o
6 ^o	5613 ^o	4333495490358 ^o	1.2729e+008 ^o	30.12% ^o
9 ^o	1000 ^o	7136680416 ^o	3.36903e+006 ^o	84.1835% ^o
9 ^o	2000 ^o	61610639702 ^o	1.53124e+007 ^o	75.6244% ^o
9 ^o	4000 ^o	787252163392 ^o	6.36155e+007 ^o	55.0021% ^o
9 ^o	5613 ^o	2996658835206 ^o	1.43416e+008 ^o	45.0521% ^o
9(助教) ^o	1000 ^o	2874132254 ^o	633009 ^o	72.7698% ^o
9(助教) ^o	2000 ^o	28497078252 ^o	3.61198e+006 ^o	60.7816% ^o
9(助教) ^o	4000 ^o	333392203868 ^o	1.58022e+007 ^o	44.7619% ^o
9(助教) ^o	5613 ^o	1270394066463 ^o	3.40021e+007 ^o	36.1835% ^o

Figure 1

注：

“6”表示 6 维轮廓中心矩特征，

“9”表示 9 维 HSV 颜色矩特征，

“9（助教）”表示文档中所给出的 9 维颜色矩特征

以下表格同

3.1 R 树搜索图片实现方法

本次特征我们选择的特征是 9 维的 HSV 颜色矩以及 6 维的轮廓中心矩，我们通过文件读取的方式从文件中读取每一个图片的特征，根据特征维度建立相应维度的 R 树，由于每一个图片的特征相当于一个 n (n 可以取 6 或 9) 的点，我们定义一个 RANGE，使这个点的坐标的每一个维度均减去一个 RANGE 得到最小点，均加上一个维度得到最大点，最小点和最大点可以确定一个 n 维的矩形。由此，每一个图片都可以对应一个以各自特征所得到的坐标为中心的矩形。搜索的时候我们通过判断有哪些矩形与待搜图片的矩形有重叠区域，来确定我们要找的图片。

3.2 R 树节点访问与平均查询结果

我们分别设置不同的特征 (HSV9 维颜色矩和轮廓中心矩) 加上助教所给出的 9 维颜色矩特征，每一种特征又分别对应不同的图片规模。以下我们简称 HSV9 维颜色矩为 HSV，助教所给的颜色矩为颜色矩。根据不同特征和图片规模我们对搜索进行了批处理，从而得到上表。(参考 Figure 1) 从表中我们可以看出，不论是哪一种特征，R 树的节点访问量总是随着节点数 (图片规模) 的增多而增多。相比较特征之间的区别，在图片数据量相同的情况下，我们发现 R 树节点访问量的大小始终保持为：轮廓中心矩 > HSV > 颜色矩。由此可见，轮廓中心矩搜索效率是最低的。而颜色矩是效率最高的。HSV 介于中间。从而图片的平均查询效果也是以轮廓中心矩最低，HSV 次之，颜色矩最好。

3.3 R 树搜索有效性比较

R 树搜索的有效性结果说明：在给定特征和图片规模 M 的情况下，我们搜索该规模下的每一张图片，对每一张图片所得的结果，假设我们得到了 A 个图片，我们通过判断所得图片的文件名前缀是否与原图前缀相同来判断是否搜索有效，从而我们在这 A 个图片中得到了 B 个有效的图片，定义有效性为 $\frac{B}{A} \times 100\%$ ，从而得到了 M 个有效性结果，取平均即得到最终的有效性结果。

确定规模的方法是从第一张图片开始顺序选取相应规模的图片放入 R 树搜索。因此，三种维度的特征在规模较小的时候有效率均比较高的原因是图片文件名前缀是按顺序排列的，导致图片类型比较单一，无论怎样搜索都有比较高的几率搜

到有效图片。因此只有在规模较大的时候才能表现出这三个特征真正情况下的有效性。

虽然在图片搜索效率方面 HSV 和轮廓中心矩较颜色矩均较慢，但是在数据范围适中的情况下，轮廓中心矩搜索出的结果有效性要比颜色矩更高，而在图片规模更高的情况下轮廓中心矩的结果有效性又逐渐降低。原因可能是图片规模越大，图片中复杂因素越多，有着同样轮廓但不是相同类型物体的图片就越多。仅仅通过一张图片中某个物体的轮廓判断与其他图片是否有相关性显然是不够的。而整体看来，不论图片的规模如何，HSV 所得到的图片的正确率始终是最好的。因为它提取的是整个图片的一个量化特征（颜色分布），不会受到图片具体内容的影响，只根据 H,S,V 三个量判断相关性，而两张差异很大的图的 H,S,V 相近的概率有很小。因此，不论图片规模多大，HSV 总可以提取出一定量的有效图片。

3.4 R 树结果的相关性及排名实现方法

每一个图片对应一个 n (n 为 6 或 9) 维的点，即矩形的中心，我们通过 R 树搜索到了 M 个图片，对这 M 个图片中的每个图片，与原图计算二者中心之间的欧式距离的平方，该值越小，说明两个图片相关性越高。因此，我们用这个值表示两个图片之间的相关性。由于每一张图片所得到的结果会有 M 个相应的相关性，我们根据这个相关性对结果进行排序。相关性越高排位越靠前。因此，为表现总体相关性，我们选择 M 个图片中与原图相关性最高的图片作为该原图的相关性结果。在给定规模的情况下，我们计算每个图片的相关性结果，然后所有图片的相关性结果取平均，由此我们得到了不同特征和维度的情况下的相关性，并在图片规模上进一步分类，从而得到上表数据。（注：不同特征之间由于特征数据数量级本身不同，导致欧氏距离平方的结果在不同的数量级，相互之间没有可比性）

由下表可知，在给定特征和维度的情况下，图片规模越大，欧氏距离平方越小，相关性越高。这是由于随着规模增大，更多相关性更高的有效性图片进入 R 树中(参考 Figure 2)

特征数据量	图片数据量	相关性
6	100	9.93766
6	500	9.0892
6	1000	8.64719
6	2000	7.18739
6	4000	5.79512
6	5613	5.03
9	100	78.6816
9	500	130.196
9	1000	144.505
9	2000	145.269
9	4000	150.746
9	5613	151.401
9(助教)	100	635049
9(助教)	500	608962
9(助教)	1000	557588
9(助教)	2000	549039
9(助教)	4000	516733
9(助教)	5613	494841

Figure 2

3.5 R 树球状节点搜索（选作）

除了以矩形为节点，我们还实现了以球为节点的 R 树。每一个图片为一个 n 维球，调整半径为合适大小。判断两个图片是否相关的方法是二者对应的球的球心之间的欧式距离是否为半径的二倍。以下为球型搜索的实验数据 (参考 Figure 3)

特征数据量	图片数据量	R 树节点访问量	平均查询结果	特征有效性
6	1000	28597342506	9.57203e+006	78.7304%
6	2000	403970823353	5.98771e+007	58.8709%
6	4000	3921859002957	1.91154e+008	35.6017%
6	5613	8577024490305	2.48801e+008	29.9705%
9	1000	20282509857	404247	58.9115%
9	2000	212434791100	1.61396e+006	39.063%
9	4000	2749205034308	5.61933+006	24.5822%
9	5613	3550026800558	3.80345e+006	18.3417%
9(助教)	1000	6784599870	2.69819e+006	80.7078%
9(助教)	2000	67495417151	1.411036e+007	70.2976%
9(助教)	4000	816916636790	6.94112e+007	56.2576%
9(助教)	5613	3041743121961	1.56619e+008	47.7048%

特征数据量	图片数据量	相关性
6	5613	5.79374
9	5613	248.914
9 (助教)	5613	377265

Figure 3

与矩形搜索相比结果相差并不大，但节点访问量明显增多。HSV 所得结果中有效性降低。这可能与节点球的半径选取有关。

4. 特征提取

4.1 HSV 颜色距

根据给定的图片，将所有的像素点的 RGB 读出来并存储起来，然后通过以下公式(参考 Figure 4)，将 RGB 转化为 HSV (undefined 设为 0)

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

Figure 4

之后通过以下公式(参考 Figure 5)，计算 H, S, V 的三个颜色维度的一二三阶距。

$$\mu_i = \frac{1}{N} \sum_{j=1}^N p_{i,j}$$
$$\sigma_i = \left(\frac{1}{N} \sum_{j=1}^N (p_{i,j} - \mu_i)^2 \right)^{\frac{1}{2}}$$
$$s_i = \left(\frac{1}{N} \sum_{j=1}^N (p_{i,j} - \mu_i)^3 \right)^{\frac{1}{3}}$$

Figure 5

(其中 $p_{i,j}$ 表示彩色图像第 j 个像素的第 i 个颜色分量)

(N 表示图像中像素个数)

最后将三个分量 H , S , V 的三个颜色距组成一个 9 维直方图向量(参考 Figure 6)

$$F_{hsv} = [\mu_H, \sigma_H, s_H, \mu_S, \sigma_S, s_S, \mu_V, \sigma_V, s_V]$$

Figure 6 图像的颜色特征表示

4.1.1 使用的类: HSV 和 Features

HSV 类存储一个像素点的 HSV; Features 存储图像的 9 维特征。

4.2 轮廓中心距

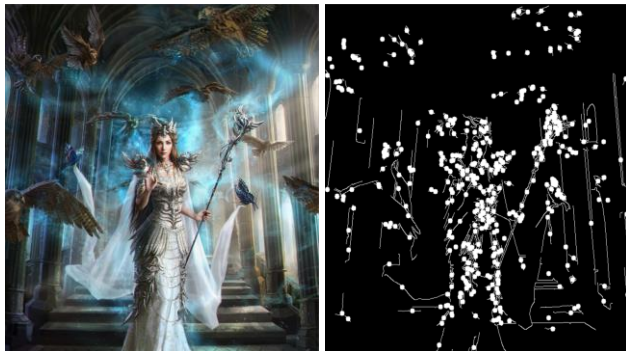
4.2.1 利用 openCV 提取轮廓特征的步骤

[1] 首先对图像进行空间转换, 通过 `cvtColor()` 函数将图像进行灰度化; [2] 然后通过 `GaussianBlur()` 函数对灰度图进行滤波; [3] 利用 `Canny()` 函数对灰度图进行 canny 边缘检测, 得到二值图像; [4] 利用 `findContours()` 函数提取图像轮廓数据, 得到一组向量; [5] 利用 `moments()` 函数得到轮廓的距; [6] 然后利用得到的轮廓距计算中心距, 得到一组数量各异的轮廓中心距数据; [7] 将这些轮廓中心距数据分 x , y 方向求其一二三阶距 (其中可以以一阶矩周围范围的 $rate$ 比率 (0 到 100%) 取一定范围数据, 以防边缘无用数据 (比如边角的 logo 轮廓数据) 对中心距数据产生影响, 默认 $rate$ 为 100%); [8] 由于中心距特征是积聚在图片中物体附近, 可能由于图像大小变化而影响二三阶距的稳定性 (即图像大小变化不影响距的变化), 所以对二三阶距除以图像面积的开方; [9] 最后得到一组 6 维特征数据 (S 表示图像面积大小) (参考 Figure 7)

$$F_{x,y} = [\mu_x, \frac{\sigma_x}{\sqrt{S}}, \frac{s_x}{\sqrt{S}}, \mu_y, \frac{\sigma_y}{\sqrt{S}}, \frac{s_y}{\sqrt{S}}]$$

Figure 7

4.2.2 效果图



原图

处理后图像(白点为中心距点)

4.2.3 使用的类: features

features 存储图像的 6 维特征

5. 可视化

为了使用户方便看程序结果, 利用计算机图形学和图像处理技术, 将数据和图像显示在屏幕上。

5.1 操作方法

先把所有图片放 `src/MFCApplication1/MFCApplication1/image` 文件夹里面, 然后打开 `exe` 文件夹内的 `MFCApplication1.exe` 文件就会弹出图形界面(参考 Figure 8)



Figure 8 图形界面

在左上方可以选择检索方式, 显示节点访问数和平均查询结果(以文件 `n03877845_1458.JPEG` 为例) (参考 Figure 9)



Figure 9

在输入框输入文件名 (ex. `n03877845_1458.JPEG`), 然后点击右边的检索按钮。(结果参考 Figure 10)

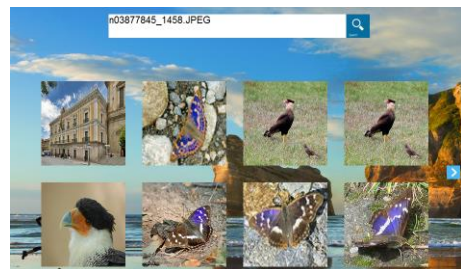


Figure 10 检索结果

图片越在左边, 越在上边, 图像的关联度越小(即图片越相似)。如果还想看剩下的图片的话, 点击图片右边的蓝色箭头就可以看了。

6. 总结:

总体特征: HSV>助教所给>轮廓中心距

效率: 助教所给>HSV>轮廓中心距

7. 参考文档

[1] openCV 官方文档地址:

<http://www.opencv.org.cn/opencvdoc/2.3.2/html/index.html>