

一种颜色直方图计算相似度的资产图片核对算法

王宁邦¹, 徐 博^{1,2}, 夏百川³, 夏 娜⁴, 邵永航⁵, 李 琼⁶

(1 云南师范大学 信息管理处, 昆明 650500; 2 云南师范大学 扶贫驻村工作队, 云南 梁河 679200;

3 云南师范大学 校长办公室, 昆明 650500; 4 云南师范大学 外语学院, 昆明 650500;

5 云南师范大学 物理与电子信息学院, 昆明 650500; 6 云南师范大学 能源与环境科学学院, 昆明 650500)

摘 要: 固定资产清查是高校、企事业、科研院所等的一项例行性工作, 资产清查过程中一般要求对固定资产拍照存档。资产清查收集到的资产图片保存至文件夹, 其中有成千上万张资产图片, 需要从中挑出来一部分资产图片丢弃, 且需要被挑出来的资产有图有真相地存放在其它文件夹, 同时2个文件夹中资产图片文件名、清晰度、大小等均不一, 针对如何快速完成这个图片文件挑选过程, 文中提出一种基于颜色直方图的资产图片相似度统计核对算法, 并对其过程进行了验证, 证明了算法的有效性。

关键词: 资产清查; 颜色直方图; 图片相似度

An asset images checking algorithm using similarity calculation by color histogram

WANG Ningbang¹, XU Bo^{1,2}, XIA Baichuan³, XIA Na⁴, TAI Yonghang⁵, LI Qiong⁶

(1 Department of Information Management, Yunnan Normal University, Kunming 650500, China;

2 Poverty Alleviation Work Team, Yunnan Normal University, Lianghe Yunnan 679200, China;

3 President Office, Yunnan Normal University, Kunming 650500, China;

4 School of Foreign Languages, Yunnan Normal University, Kunming 650500, China;

5 School of Physics and Electronic Information Technology, Yunnan Normal University, Kunming 650500, China;

6 School of Energy and Environmental Science, Yunnan Normal University, Kunming 650500, China)

Abstract: Fixed assets inventory is a routine work in colleges and universities, enterprise and institution, research institutes, etc.. In the process of asset inventory, it is generally required to take photos of fixed assets. There are thousands of images in the asset-image file collected by asset assessment. As required, part of the assets should be picked out and these picked assets need to be put into another file, meanwhile its asset images in two folders have different file name, definition and size. Aiming at how to complete the process quickly, this paper puts forward the algorithm based on color histogram for gathering statistics and checking asset image similarity. After that, the process is verified and the effectiveness of the algorithm is also proved.

Key words: asset inventory; color histogram; picture similarity

引言

随着多媒体技术及 Internet 技术的迅速发展, 各行各业对图像的使用日益广泛, 图像信息资源的管理和检索也越发重要。通过手工标记和索引图像的传统方法已经难以满足现实需求, 因而亟待设计推出有效研发方法。目前, 颜色直方图的应用研究普遍存在于图像检索、目标跟踪粒子滤波^[1]、遥感

等各个领域。其中, 王向阳^[2]等人基于视觉权值的分块颜色直方图图像相似度计算提出了一种新方法用于图像检索; 姜兰池^[3]等也发表了一种新的基于 HSV 非均匀量化分块颜色直方图的图像检索方法; 胡屹群^[4]等人结合视觉词袋模型和颜色直方图探讨给出了一种遥感影像检索方法; 张俊杰^[5]等人利用颜色直方图进行二次识别实现了排除误检区域和敌我识别双重功能; 邵建荃^[6]将颜色直方图、SURF

基金项目: 国家自然科学基金(11363007); 云南省教育科学研究课题(BC15009); 云南省教育厅科学研究课题(2016ZZX083)。

作者简介: 王宁邦(1985-), 男, 硕士, 助理研究员, 网络管理科副科长, 主要研究方向: 信息安全与网络计算、计算机网络应用; 徐 博(1988-), 男, 硕士, 研究实习员, 主要研究方向: 智能信息处理、数据库技术; 夏百川(1978-), 男, 博士研究生, 助理研究员, 主要研究方向: 边疆经济与社会发展; 夏 娜(1986-), 女, 硕士, 助教, 主要研究方向: 外国语言学及应用语言学; 邵永航(1988-), 男, 博士研究生, 讲师, 主要研究方向: 图像处理、电路设计; 李 琼(1982-), 男, 博士研究生, 实验师, 主要研究方向: 太阳能热利用。

通讯作者: 王宁邦 Email: popularmp3007@163.com

收稿日期: 2018-03-02

特征、时间信息三者相结合进行车辆跟踪。

固定资产清查是高校、企事业、科研院所等部门的一项常规例行性工作,资产清查过程中一般要求对固定资产拍照存档。资产清查将收集到的资产图片保存至文件夹,其数目甚至可达成千上万张,需要从中挑选出一部分资产图片,并将这部分资产图片及详情资料存放在其它文件夹,与此同时2个文件夹中资产图片文件名、清晰度、大小等均为不一。针对如何减少人工消耗并明显缩短图片文件挑选过程时间,文中设计提出了一种基于颜色直方图相似度计算的资产图片核对算法。

1 颜色直方图的相似度计算方法原理

通过将原始图像与待比较图像的像素数据进行采集,生成特定的直方图数据,运用巴氏系数^[7-10] (Bhattacharyya coefficient) 算法,计算出2个图像的相似程度,数值在[0,1]之间,0表示极其不同,1表示极其相似(相同)。

1.1 直方图计算

文中采用的是单一直方图,每个颜色的值范围为0~255。假定将颜色分为一定范围等份,如果分为16等份的话,此时每个等份值范围为16,再假设RGB值为(14,68,221),16等份后,其所对应直方图索引值(index)分别为:(0,4,13)。研究推得,索引值的计算公式为:

$$index = R + G * 16 + B * 16 * 16 \quad (1)$$

对应的直方图 $index = 0 + 4 * 16 + 13 * 16 * 16$, 依次遍历所有RGB像素值,

(1) 核心代码处理。研发代码如下:

```
String fileurl = "C:\\Tepfile\\backgrounds";
String fileurl2 = "C:\\Tepfile\\image";
ArrayList<String> anArray = ReadFile.traverseFolder1(fileurl);
ArrayList<String> anArray2 = ReadFile.traverseFolder1(fileurl2);
for (int i = 0; i < anArray.size(); i++) {
    try {
        sourceImage = null;
        sourceImage = ImageIO.read(new File InputStream(anArray.toArray()[i].toString()));
        this.repaint();
    } catch (FileNotFoundException e1) {
        e1.printStackTrace();
    } catch (IOException e1) {
        e1.printStackTrace();
    }
}
repaint();
for (int i1 = 0; i1 < anArray2.size(); i1++) {
    try {
        candidateImage = null;
        candidateImage = ImageIO.read(new File InputStream(anArray2.toArray()[i1].toString()));
        ImageComparer imageCom = new ImageComparer(sourceImage, candidateImage);
        similarity = imageCom.modelMatch();
        System.out.println(similarity);
        this.repaint();
    }
}
```

即可完成对直方图数据计算。

1.2 巴氏系数计算

巴氏系数^[3]数学运算公式为:

$$\rho(p, p') = \sum_{i=1}^N \sqrt{p(i)p'(i)} \quad (2)$$

其中, p, p' 分别表示原始与待比较的图像直方图数据,对2个图像中的每一个第*i*部分数据点相乘后进行开平方再相加得到的值就是图像相似度值。

2 资产核对算法设计流程与实现

2.1 算法流程

(1) 初始化待处理数据源、比较数据源、目标数据结果。

(2) 分析处理内容为:

① 提取待处理数据源中一张图片,与目标数据源进行图像匹配,达到设定阈值的图片便是目标数据,将该目标文件复制到指定文件夹;

② 如果待处理数据源资产还未为空,则循环执行①。

2.2 实现过程

(1) 开始;读取文件(待处理图片文件、比较对象图片文件)路径。

(2) 打开待处理图片文件。

(3) 打开比较对象图片文件。

(4) 比较2张图片相似度。

(5) 将比较结果提供一定处理:复制、删除等。

(6) 循环以上读取到复制的过程;结束。

2.3 重点程序代码设计

```

    } catch ( FileNotFoundException e1) {
        e1.printStackTrace();
    } catch ( IOException e1) {
        e1.printStackTrace();
    }
}
}
}

```

(2) 相似度计算。研发代码可见如下:

```

public double modelMatch() {
    HistogramFilter hfilter = new HistogramFilter();
    float[] sourceData = hfilter.filter( source Image ,null);
    float[] candidateData = hfilter.filter( candidate Image ,null);
    double[] mixedData = new double [source Data.length];
    for( int i = 0; i < sourceData.length; i ++ ) {
        mixedData [i] = Math.sqrt( sourceData [i] * candidateData [i] );
    }
    double similarity = 0;
    for( int i = 0; i < mixedData.length; i ++ ) {
        similarity += mixedData [i];
    }
    return similarity;
}

```

(3) 直方图数据采集。实现时可运行如下代码:

```

public float[] filter( BufferedImage src ,BufferedImage dest) {
    int width = src.getWidth();
    int height = src.getHeight();
    int[] inPixels = new int [width* height];
    float[] histogramData = new float [redBins * greenBins * blueBins];
    getRGB( src ,0 ,0 ,width ,height ,inPixels );
    int index = 0;
    int redIdx = 0 ,greenIdx = 0 ,blueIdx = 0;
    int singleIndex = 0;
    float total = 0;
    for( int row = 0; row < height; row ++ ) {
        int ta = 0 ,tr = 0 ,tg = 0 ,tb = 0;
        for( int col = 0; col < width; col ++ ) {
            index = row * width + col;
            ta = ( inPixels [index] >> 24) & 0xff;
            tr = ( inPixels [index] >> 16) & 0xff;
            tg = ( inPixels [index] >> 8) & 0xff;
            tb = inPixels [index] & 0xff;
            redIdx = (int) getBinIndex( redBins ,tr ,255);
            greenIdx = (int) getBinIndex( greenBins ,tg ,255);
            blueIdx = (int) getBinIndex( blueBins ,tb ,255);
            singleIndex = redIdx + greenIdx * redBins + blueIdx * redBins * greenBins;
            histogramData [singleIndex] += 1;
            total += 1;
        }
    }
    for ( int i = 0; i < histogramData.length; i ++ )
    {
        histogramData [i] = histogramData [i] / total;
    }
    return histogramData;
}

private float getBinIndex( int binCount ,int color ,int colorMaxValue) {
    float binIndex = ( ( ( float) color) / ( ( float) colorMaxValue)) * ( ( float) binCount);
    if ( binIndex >= binCount)
        binIndex = binCount - 1;
    return binIndex;
}

public int[] getRGB( BufferedImage image ,int x ,int y ,int width ,int height ,int[] pixels ) {

```

```

int type = image.getType();
if ( type == BufferedImage.TYPE_INT_ARGB || type == BufferedImage.TYPE_INT_RGB )
    return ( int [] ) image.getRaster().getDataElements( x , y , width , height , pixels );
return image.getRGB( x , y , width , height , pixels , 0 , width );
}

```

3 算法验证

过程处理的运行界面如图1所示。



图1 资产核对界面

Fig. 1 Asset check interface

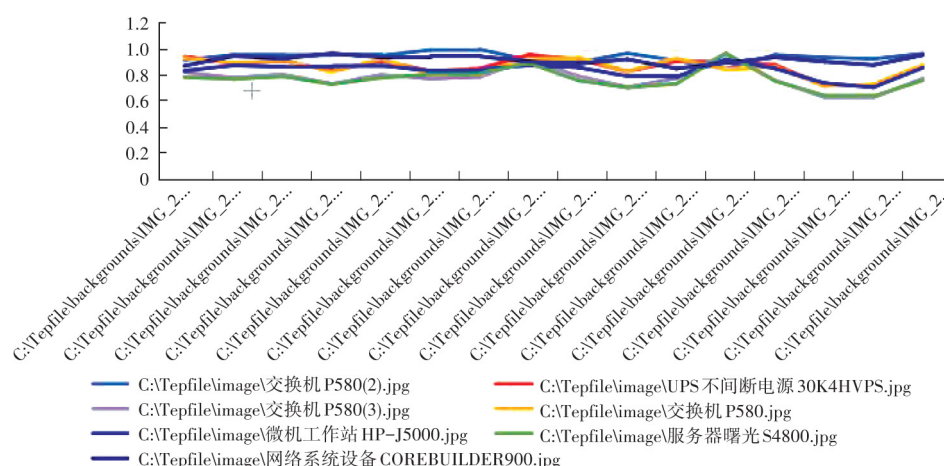


图2 资产核对相似度

Fig. 2 Asset check similarity

资产图片相似度计算过程耗时则如图3所示。

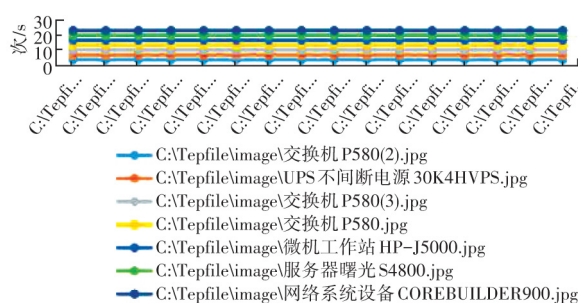


图3 资产核对相似度计算耗时

Fig. 3 Asset check similarity computing time

4 结束语

文中针对国有资产清查事件中遇到的图片核对过程工作量大等特点,设计提出了基于颜色直方图相似度计算的资产核对算法。首先分析探讨颜色直方图的基础算法原理,而后论述了文中研发的算法流程和实现过程,最后则对算法的应用进行了有效

颜色直方图是在许多图像检索系统中已获得广泛采用的颜色特征,算法简单方便,经过多次试验,如图1中相似度为0.955 325 295 373 811 1的情况,是由于2个资产的色调高度相同所致,所以研究将最高阈值设置为大于等于0.999 999 99,即只有相似度到达这个数值,才会判定为是2个相同的资产。阈值的设定需要自定义。

实验模拟从16张资产图片中挑选出7张,相似度计算数值如图2所示,可见其中有一张相似度近似为1满足丢弃的要求。

性验证。虽然颜色直方图算法简单、应用普遍,也高效达成了预期研究指定的资产清查图片核对的任务内容,但是相似度计算过程表现出耗时长,这是接下来可以进一步研究和改进的方向。

参考文献

- [1] 陶立超,赵宇明.基于分块颜色直方图和粒子滤波的物体跟踪[J].计算机工程与应用,2012,48(7):165-168.
- [2] 王向阳,杨红颖,郑宏亮,等.基于视觉权值的分块颜色直方图图像检索算法[J].自动化学报,2010(10):1489-1492.
- [3] 姜兰池,沈国强,张国焯.基于HSV分块颜色直方图的图像检索算法[J].机电工程,2009,26(11):54-57.
- [4] 胡屹群,周绍光,岳顺,等.利用视觉词袋模型和颜色直方图进行遥感影像检索[J].测绘通报,2017(1):53-57.
- [5] 张俊杰,丁淑艳,李伦波.基于类Haar特征和颜色特征的NAO机器人识别[J].计算机与现代化,2017(2):30-35.

(下转第67页)

生多次递归,实现过程中即需要消耗更多的处理时间。而算法 WFP-SW 在挖掘过程中,WE-tree 仅需存储加权频繁模式信息,对于不频繁模式则免于存储,因此所需的内存更少。同时,图 7 又研究指出了 WFP-SW 算法的查全率 (Recall) 和查准率 (Precision) 都要优于 WIT-FWI 算法的挖掘结果。

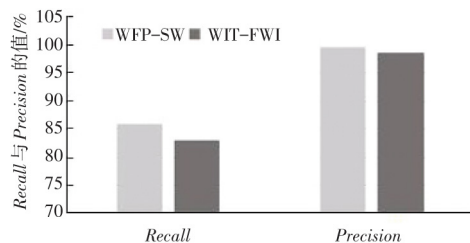


图7 算法 WFP-SW 和 WIT-FWI 挖掘效果比较

Fig. 7 Mining results comparison between WFP-SW and WIT-FWI

4 结束语

已有研究表明,仅仅通过模式的频度挖掘会引起数据丢失,而数据流加权频繁模式挖掘能提炼得到更富价值的信息。为此,本文提出了一种基于滑动窗口模型的数据流加权频繁模式挖掘算法 WFP-SW。算法采用枚举树 WE-tree 结构通过维护 Tid 列表计算模式加权支持度,列表间的交集获得更高层的模式信息;通过虚权维持 WE-tree 的向下闭合特性防止信息丢失。大量的仿真实验表明, WFP-SW 算法有较低的运行时间,并且所需的内存也将会更少。

参考文献

- [1] MEMAR M, DEYPIR M, SADREDDINI M H, et al. An efficient frequent itemset mining method over high-speed data streams [J]. Computer Journal, 2012, 55(11): 1357-1366.
- [2] LIN J C W, GAN Wensheng, FOURNIER-VIGER P, et al. Weighted frequent itemset mining over uncertain databases [J]. Applied Intelligence, 2016, 44(1): 232-250.
- [3] YUN U. On pushing weight constraints deeply into frequent itemset mining [J]. Intelligent Data Analysis, 2009, 13(2): 359-383.
- [4] TANBEER S K, AHMED C F, JEONG B S, et al. Sliding window-based frequent pattern mining over data streams [J]. Information Sciences: An International Journal, 2009, 179(22): 3843-3865.

- [5] YUN U, PYUN G, YOON E. Efficient mining of robust closed weighted sequential patterns without information loss [J]. International Journal on Artificial Intelligence Tools, 2015, 24(1): 1550007(1) -1550007(28).
- [6] 李国徽, 陈辉. 挖掘数据流任意滑动时间窗口内频繁模式 [J]. 软件学报, 2008, 19(19): 2585-2596.
- [7] YUN U, LEE G, RYU K H. Mining maximal frequent patterns by considering weight conditions over data streams [J]. Knowledge-Based Systems, 2014, 55(55): 49-65.
- [8] WOO H J, LEE W S. estMax: Tracing maximal frequent item sets instantly over online transactional data streams [J]. IEEE Transactions on Knowledge & Data Engineering, 2009, 21(10): 1418-1431.
- [9] 韩荫, 王志海, 原继东. 一种基于时间衰减模型的数据流闭合模式挖掘方法 [J]. 计算机学报, 2015, 38(7): 1473-1483.
- [10] RYANG H, YUN U, RYU K H. Discovering high utility itemsets with multiple minimum supports [J]. Intelligent Data Analysis, 2014, 18(6): 1027-1047.
- [11] SHIE B E, HSIAO H F, TSENG V S, et al. Mining high utility mobile sequential patterns in mobile commerce environments [M] // YU J X, KIM M H, UNLAND R. Database Systems for Advanced Applications. DASFAA 2011. Lecture Notes in Computer Science, Berlin/Heidelberg: Springer, 2011, 6587: 224-238.
- [12] YUN U, RYANG H. Incremental high utility pattern mining with static and dynamic databases [J]. Applied Intelligence, 2015, 42(2): 323-352.
- [13] AHMED C F, TANBEER S K, JEONG B S, et al. Interactive mining of high utility patterns over data streams [J]. Expert Systems with Applications, 2012, 39(15): 11979-11991.
- [14] LIN M Y, TU T F, HSUEH S C. High utility pattern mining using the maximal itemset property and lexicographic tree structures [J]. Information Sciences, 2012, 215(23): 1-14.
- [15] SUN Hong, BIE T D, STORMS V, et al. ModuleDigger: An itemset mining framework for the detection of cis-regulatory modules [J]. BMC Bioinformatics, 2009, 10(S1): S30.
- [16] KIM Y S. Streaming association rule (SAR) mining with a weighted order-dependent representation of Web navigation patterns [J]. Expert Systems with Applications, 2009, 36(4): 7933-7946.
- [17] BARALIS E, CAGLIERO L, CARZA P. Planning stock portfolios by means of weighted frequent itemsets [J]. Expert System with Applications, 2017, 86(15): 1-17.
- [18] VO B, COENEN F, LE B. A new method for mining frequent weighted itemsets based on WIT-trees [J]. Expert Systems with Applications, 2013, 40(4): 1256-1264.

(上接第 62 页)

- [6] 邵建奎. 非重叠场景下多摄像机车辆跟踪研究 [J]. 现代计算机 (专业版), 2017(2): 49-53.
- [7] BHATTACHARYA A. On a measure of divergence between two statistical populations defined by their probability distributions [J]. Bulletin of the Calcutta Mathematical Society, 1943, 35: 99-109.
- [8] 王宁邦, 刘江涛, 梁红飞, 等. Cacti 在可视化校园网络管理中的

应用 [J]. 云南民族大学学报 (自然科学版), 2018, 27(2): 129-135.

- [9] 王宁邦, 徐博. 一种桌面门户移动化解决方法 [J]. 昆明冶金高等专科学校学报, 2017, 33(3): 53-58.
- [10] 徐博, 王宁邦, 陈彬, 等. 网络设备故障解决一体化平台初探 [J]. 网络安全技术与应用, 2016(8): 24-26.