文章编号:1671 - 6833(2015)04 - 0109 - 05

基于 Hadoop 的图像纹理特征提取

赵进超, 朱颢东, 申 圳, 李红婵

(郑州轻工业学院 计算机与通信工程学院, 河南 郑州 450002)

摘 要: 随着数字图像规模的不断增加,图像纹理特征提取已成为制约数字图像处理性能的一个关键步骤. Hadoop 是一个性能卓越的开源大数据处理云平台,其向用户提供了 MapReduce,HDFS 等模块. 首先对 Hadoop 平台、编程框架和 Tamura 纹理特征进行了介绍,然后将图像纹理特征提取过程在 Hadoop 平台上进行了实现. 在这个过程中,每个 Map 任务对应一个图像文件,各节点可以同时提取集合内图像的纹理特征. 实验表明:在图像数量较少和分辨率较低的情况,Hadoop 不同节点数量所用时间并无太大差异. 在图像分辨率较高且数量较多的情况下,Hadoop 平台表现出较高的计算效率.

关键词: Hadoop;Tamura 纹理特征;图像处理;特征提取

中图分类号: TP301 **文献标志码**: A doi:10.3969/j.issn.1671 - 6833.2015.04.025

0 引言

基于人类对纹理视觉感知的心理学研究, Tamura 等[1]提出了一种新的纹理特征表达方法. 该方法包含 6 个分量,分别与心理学上对纹理特征定义的 6 种属性相对应,它们依次是粗糙度(Coarseness)、对比度(Contrast)、方向度(Directionality)、线像度(Linelikeness)、规整度(Regularity)和粗略度(Roughness). 这些特征中最重要的是粗糙度(Coarseness)、对比度(Contrast)、方向度(Directionality). 图像分辨率越高,图像细节部分的信息就会得到更好的体现,我们就能得到更好的纹理特征,但是随之而来的是计算量和计算时间的增加. 为了缩短纹理特征提取时间,笔者拟将 Tamura 算法和 Hadoop 相结合,提出一种基于云计算 Hadoop 的 Tamura 算法,以实现纹理特征快速提取.

1 Hadoop 简介

Hadoop[2]是一个并行计算平台,在与图像相关的领域有广泛应用. 例如朱义明[3]在 Hadoop 上实现的图像分类系统;张良将等[4]在 Hadoop 平台下实现的 canny 边缘检测、尺寸调整运算;陈广钊[5]以 Hadoop 为基础开发出海量图像检索平台;李倩等[6]根据 Hadoop 平台对内部数据类型的设计要求,实现了一种功能可扩展的支持图像文件的 Hadoop 数据类型;Ranajoy Malakar 等[7]将 NVIDIA 开发的 CUDA 技术与 Hadoop 相结合实现了一个高性能图像处理系统;Liu 等[8]实现了一个基于 HBase 和 Hadoop 的海量图像管理系统.

1.1 MapReduce

MapReduce[9]是谷歌开发的并行数据处理框架,该框架具备高可靠性和良好的容错能力,基于它编写的 Hadoop 程序可以在由数千台计算机构成的大型集群上安全高效的运行,对海量数据进行并行处理. Hadoop 能够实现对多种类型文件的处理,比如文本、图像、视频等. 我们可以以特定需求为依据来编写特定的应用程序完成任务目标. 下面以 Hadoop 自带的 WordCount 程序为例来说明 MapReduce 执行流程,如图 1 所示.

首先,由 TextInputFormat 把目标文件分割为逻辑上的 split,每个 split 会被应用到一个单独的 Mapper 上;同时提供 RecorderReader 的实现,用来对逻辑分片中的数据进行处理并形成键值对 < key, value > ,作为 Mapper 任务的输入.

其次, Map 接收 RecorderReader 形成的 <

收稿日期:2014 - 12 - 01;**修订日期**:2015 - 02 - 12

基金项目:国家自然科学基金资助项目(61201447);河南省高等学校青年骨干教师资助计划项目(2014GGJS - 084);河南省科技创新杰出人才计划项目(134200510025);河南省教育厅科学技术研究重点项目(13A520367);郑州轻工业学院校级青年骨干教师培养对象资助计划项目(XGGJS02)

作者简介:赵进超(1978 -),男,河南登封人,郑州轻工业学院讲师,硕士,主要研究方向为智能信息处理、智能计算, E-mail: zhaojinchao101@163.com.

key,value > 对,根据程序设定的处理逻辑对数据进行处理,生成新的 < key, value > 对. 获得 map 计算所得的 < key, value > 对后,Mapper 会以 key 值大小为基础,按照字典排序的方法对上述 < key, value > 对进行排序,并执行 Combine 过程,将 key 值相同的 value 值累加,从而得到 Mapper 的最终输出结果 Intermediate Files.

最后,Reducer 会先将接收到的 Intermediate Files 进行排序,再交由用户自定义的 reduce 方法进行处理,得到新的 < key, value > 对,并作为程序的处理结果,按照程序设计者设定的输出格式,由 RecordWriter 写入指定位置.

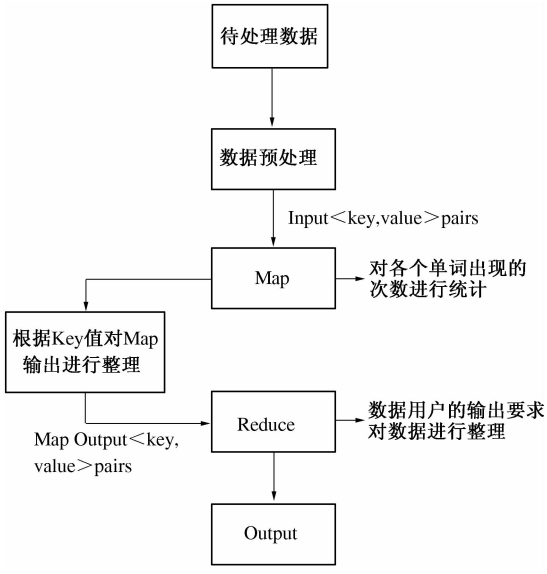

图 1 MapReduce 执行流程图

Fig. 1 MapReduce Execution Process

1.2 HDFS

Hadoop 是一个能够让用户轻松构建和运行的开源并行云计算平台,用户可以在 Hadoop 上实现对海量数据的高效处理,其中,Hadoop 分布式文件系统(HDFS)[3]扮演了非常基础的作用,它以文件系统的形式为应用提供海量数据存储服务. HDFS 具备现有的分布式文件系统的很多共同点,例如高可用性、高安全性和负载均衡等,但是它也存在一些新的特点,例如支持超大文件、流式数据访问等. 因此,HDFS 在高并发、高吞吐量的环境下得到了广泛的应用.

HDFS 的架构如图 2 所示,整体上是以 Master/Slave 架构为主,主要包括包括 4 个功能模块:Client,NameNode,Secondary NameNode 和 DataNode.

(1) Client:用户与 NameNode、DataNode 进行信息交换,实现对 HDFS 中文件的存取.

(2) NameNode:HDFS 文件系统的控制核心,负责对系统中文件目录信息、元数据信息等进行管理维护,随时监控各个 DataNode 的健康状态.

(3) Secondary NameNode:定期合并 fsimage 和 edits 日志,并传输给 NameNode.

(4) DataNode:每个节点配置一个 DataNode,数据以若干个大小固定的 block 块的形式在其上存储,在规定时间内与 NameNode 进行通信,汇报本节点内的空间利用和数据存储情况.

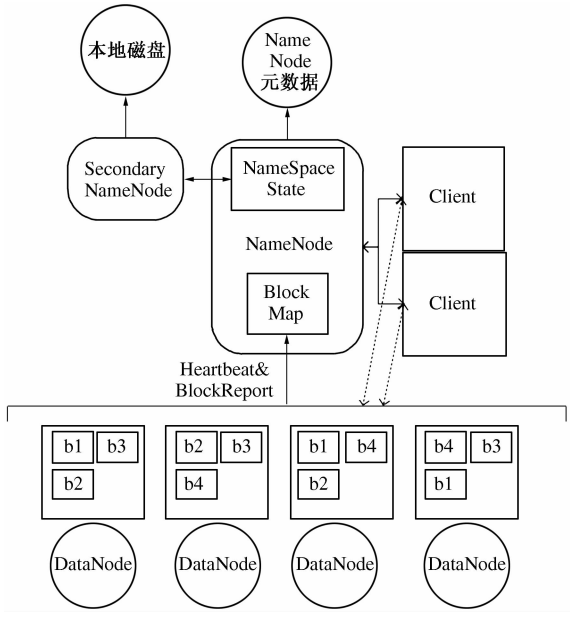

图 2 HDFS 架构图

Fig. 2 HDFS Framework

2 Tamura 纹理特征

2.1 粗糙度

粗糙度[1]是纹理最基本的特征之一,是反映纹理中颗粒度一个量. 当窗口大小不同时,具有较大窗口的纹理模式让人觉得更为粗糙. 粗糙度的具体计算步骤如下:

首先,设定窗口大小为 $2^k \times 2^k$,用公式(1)计算目标图像中窗口范围内像素的平均灰度值.

$$A_k(x,y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}} \sum_{j=y-2^{k-1}}^{y+2^{k-1}} f(i,j)/2^{2k}. \quad (1)$$

其中,$k = 0,1,\cdots,5$;$f(i,j)$ 是坐标(i,j) 处像素的灰度值.

其次,分别计算当前位置像素在水平和垂直方向上互不相交的窗口之间的平均灰度差值,如公式(2)所示.

$$E_{k,h}(x,y) = | A_k(x+2^{k-1},y) - A_k(x-2^{k-1},y) |;$$

$$E_{k,v}(x,y) = |A_k(x,y+2^{k-1}) - A_k(x,y-2^{k-1})|. \quad (2)$$

最佳尺寸计算公式为 $S_{\text{best}}(i,j) = 2^k$,若当前 k 值可以使差值 E 达到最大,即为最佳尺寸.

最后,通过计算图像中各像素位置最佳窗口的平均值即可得到粗糙度,如公式(3) 所示.

$$F_{\text{crs}} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} S_{\text{best}}(i,j). \quad (3)$$

其中 m 和 n 分别为图像的长和宽.

2.2　对比度

对比度[1] 是对目标图像的灰度值分布进行统计得到的. 一般情况下,可以通过 $\alpha_4 = \mu_4/\sigma^4$ 来定义. 对比度是通过公式(4) 衡量的.

$$F_{\text{con}} = \frac{\sigma}{\alpha_4^{1/4}}. \quad (4)$$

式中:μ_4 是四次矩;σ^2 是方差. F_{con} 给出整个图像或区域内对比度全局度量.

2.3　方向度

由于不同的纹理图像具有不同的方向性,因此 Tamura 用方向度[1] 来描述纹理在某些方向上发散或者集中. 首先,计算当前像素位置的梯度向量. 该向量的模和方向定义如公式(5) 所示.

$$\begin{aligned} &|\Delta G| = (|\Delta_H| + |\Delta_V|)/2; \\ &\theta = \tan^{-1}(\Delta_V/\Delta_H) + \pi/2. \end{aligned} \quad (5)$$

其中 Δ_H 和 Δ_V 是使用图3 所示两个 3 × 3 算子与图像做卷积得到的.

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

(a)算子 1　　(b)算子 2

图 3　3 × 3 算子示意图

Fig. 3　3 × 3Schematic diagram of operator

其次,利用公式(6) 来获得 θ 的直方图.

$$H_D(k) = N_\theta(k)/\sum_{i=0}^{n-1} N_\theta(i). \quad (6)$$

式中:n 为方向角度的量化等级;t 为阈值. $N_\theta(k)$ 是当 $|\Delta G| \geqslant t$, $(2k-1)\pi/2n \leqslant \theta \leqslant (2k+1)\pi/2n$ 时像素的数量. 若当前图像的方向性并不突出,则直方图 H_D 比较平缓,反之会出现较为明显的峰值.

最后,使用公式(7) 来计算方向度.

$$F_{\text{dir}} = \sum_{p}^{n_p}\sum_{\phi\in w_p}(\phi - \phi_p)^2 H_D(\phi). \quad (7)$$

式中:n_p 是对直方图 H_D 中峰值数量的统计值;p 为直方图 H_D 中的峰值,对于任意一个峰值 p,w_p 为图像中达到该峰值的所有区域;ϕ_p 是 w_p 中最大直方图值中的波峰中心位置.

3　图像纹理特征提取实现

笔者把一个图像文件作为一个 split,把整个图像集合视为一个作业进行处理,每个 Map 任务对应一个图像文件,进而可以同时提取集合内图像的纹理特征. 还使用单独一个 Reduce 任务将计算结果按照设定格式写入到指定输出位置. 为了实现上述功能,首先,需要实现一个新的数据类型 Image,用来存储图像像素信息;其次,与文件输入相关的 InputFormat 和 RecordReader 也需要重定义,用于图像文件和特定数据类型之间的转化;最后,在 Map 处理阶段实现图像纹理特征提取.

3.1　数据类型 Image

Hadoop 本身没有定义和图像相关的类作为 Key 和 Value 的备选类型. Hadoop 规定,用户自定义的类型只有通过实现 Writable 接口才能使用. 为解决上述问题,笔者自定义了数据类型 Image,该数据类型是以 Bufferimage 为基础进行扩展,对 Hadoop 中 Writable 所定义的用于输入输出的基本方法进行了重写. 与其他类型相比,该类型在实现读取图像尺寸、图像路径等功能的基础时,又根据实际需要增加了相应的功能模块,例如灰度变换、颜色空间变换等功能. 部分扩展内容如图 4 所示.

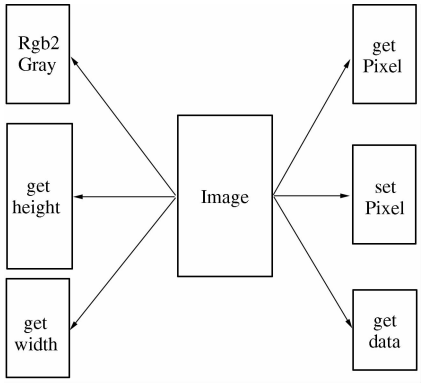

图 4　Image 类型内容缩略图

Fig. 4　Image content thumbnail

3.2　图像文件的输入格式

InputFormat 描述了 Hadoop 作业输入的细节规范,而 FileInputFormat 则是所有以文件作为其数据来源的 InputFormat 实现的基础类型. Hadoop 提供的 API 实现了下面两个类:

(1) ImageFileInputFormat:继承自 ImageFileInputForma 类的实现,将一个图像文件视为一个 split,不再对图像进行分割.

(2) ImageRecordReader:继承自 RecordReader 类的实现,把输入分片转化为一个 <key, value> 对.

3.3 图像文理特征中粗糙度计算

Tamura 对粗糙度计算中 k 值设定描述为两种情况:①$k=0,1,2,3$:图像无噪声,在该范围内 S_{best} 恒定 Max,计算量小;②$k=0,1,\cdots,5$:图像存在噪声,S_{best} 结果表现不稳定,不仅对计算结果造成影响,而且计算量与无噪声情况下相比要大.因此在提取纹理特征前需要对输入图像进行预处理以祛除噪声,保证结果质量的同时还可以减少计算.

4 对比实验及结果分析

本 Hadoop 实验平台由 5 台计算机组成,操作系统均为 CentOS - 6.4 64bit,配置均为八核 Intel-Corei7 处理器,4GB 内存,1TB 硬盘,Hadoop 版本为 1.1.2,Java 版本为 1.7.25,每个节点通过 100 Mb/s 的局域网连接.

为了验证该算法在不同图像分辨率、不同图像数量和不同节点数目情况下 Hadoop 平台的提取效率,笔者选用 3 个数据集:Flavia,ICL 和 ImageClef,下载网址如表 1 所示.从 3 个数据集中抽取 2 000 张图片,分 100 张、200 张、500 张、1 000 张、2 000 张 5 组,分别使用 3 节点和 4 节点进行纹理特征提取结果对比.所用时间如图 5、图 6 所示,算法加速比如图 7 和图 8 所示.

表 1 数据集网址

Tab.1 Data set website

图像数据集	下载网址
Flavia	http://flavia.sourceforge.net/
ImageClef	http://www.imageclef.org/2012/plant
ICL	http://www.intelengine.cn/dataset

图 5、图 6 说明随着图像数量的增加,Hadoop 平台的特征计算时间基本呈倍数级增长.原因有如下两点:①Flavia 库中图像分辨率均为 800 × 600,ImageClef 的为 500 × 800 左右,ICL 的为 300 × 400 左右,Tamura 纹理特征中粗糙度的计算量和图像分辨率密切相关,因此计算时间增长较为明显;②计算特征时采用的是 Hadoop 的默认调度策略,并未针对并行图像处理的特点对调度策略进行调整.

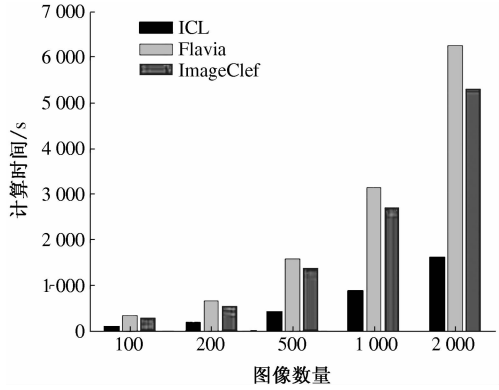

图 5 3 节点时 Hadoop 平台计算时间

Fig.5 Consumed Time of Hadoop Platform with Three Nodes

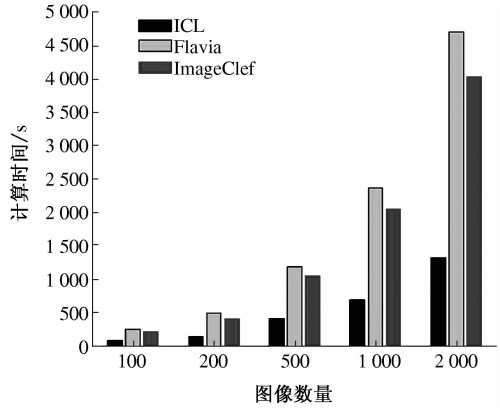

图 6 4 节点时 Hadoop 平台计算时间

Fig.6 Consumed Time of Hadoop Platform with Four Nodes

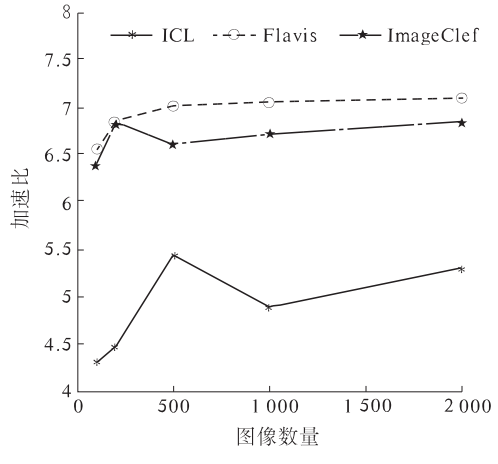

图 7 3 节点时加速比

Fig.7 Speedup ratio of hadoop platform with three nodes

将图 5 和图 6 对比,结合图 7、图 8 可以看出:在图像数量较少和分辨率较低的情况下,不同的节点数量对处理时间的影响并未呈现出明显的差异,加速比则表现出一定的差异;随着图像数量的增加和图像分辨率的提高,不同节点数量的处理

时间和加速比的差异尤为明显.实验表明基于 Hadoop 平台的 Tamura 算法可以有效地运用于大规模图像数据集的特征提取.

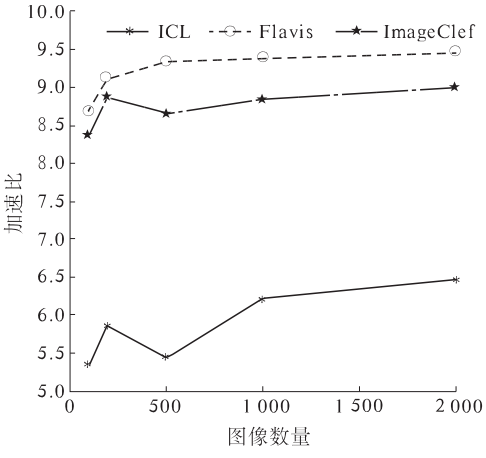

图 8　4 节点时加速比

Fig. 8　Speedup ratio of hadoop platform with four nodes

5　结论

笔者主要基于 Hadoop 平台,利用 Tamura 算法实现图像纹理特征的快速提取,针对 Hadoop 平台无法直接读取图像文件的实际情况,设计实现了一种新的输入格式 ImageInputFormat 和数据类型 Image 来满足图像输入、数据处理的需要.该方法充分发挥了 Hadoop 平台对大数据并行处理的能力,在保证数据精度的同时也缩短了计算时间,对比实验表明了该方法的有效性.然而,在实验过程中,由于 Hadoop 的 Block 块大小为 64 MB,而实验所用图像大小不超过 1 MB,浪费了大量的存储空间.同时,受限于 Hadoop 平台的调度策略,使得算法的时效性受到影响.如何提高系统在存储大量小尺寸文件时的存储空间利用率,设计出更好的调度策略,是笔者下一步的研究重点.

参考文献:

[1] TAMURA H, MORI S, YAMAWAKI T. Textural features corresponding to visual perception [J]. IEEE Transactions on Systems, Man and Cybernetics, 1978, 8(6): 460 – 473.

[2] ARMBRUST M, FOX A, GRIFFITH R, etal. A view of cloud computing[J]. Communications of the ACM, 2010, 53(4): 50 – 58.

[3] 朱义明. 基于 Hadoop 平台的图像分类[J]. 西南科技大学学报, 2011, 26(2): 70 – 73.

[4] 张良将,宦飞,王杨德. Hadoop 云平台下的并行化图像处理实现[J]. 信息安全与通信保密, 2012, 20(10): 59 – 62.

[5] 陈广钊. 基于 MapReduce 的海量图像检索技术研究 [D]. 西安: 西安电子科技大学计算机学院, 2012.

[6] 李倩,施霞萍. 基于 Hadoop MapReduce 图像处理的数据类型设计[J]. 软件导刊, 2012, 11(4): 182 – 183.

[7] MALAKAR R, VYDYANATHAN N. A CUDA-enabled hadoop cluster for fast distributed image processing[C]//Proceedings of the 2013 National Conference on Parallel Computing Technologies. Bangalore, India: IEEE, 2013: 1 – 5.

[8] LIU Yue-hu, CHEN Bin, HE Wen-xi, et al. Massive image data management using hBase and mapreduce [C]//Proceedings of the 2013 21st International Conference on Geoinformat ics. Kaifeng, China: IEEE, 2013: 1 – 5.

[9] MARSTON S, LI Zhi, SUBHAJYOTI B, et al. Cloud computing the business perspective[J]. Decision Support Systems, 2011, 51(1): 176 – 189.

Image Texture Feature Extraction Based on Hadoop

ZHAO Jin-chao, ZHU Hao-cong, SHEN Zhen, LI Hong-chan

(School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China)

Abstract: With the increasing amount of digital image data, image texture feature extraction has become a key step of digital image processing. As an excellent massive data processing and storage capacity of the open source cloud platform, Hadoop provides a parallel computation model MapReduce, HDFS distributed file system module. In this paper, we firstly introduced Hadoop platform programming framework and Tamura texture features. And then, the image texture feature extraction was carried out on the Hadoop platform. In the process, every Map task corresponds to an image file, every nodes work at same time. The comparison results show that number of nodes have no influence about the processing time, when we have little images and the image has low-resolution. On the contrary, Hadoop paltform is more effective.

Key words: Hadoop; Tamura texture feature; image processing; feature extraction