

监控系统预警视频的分布式检索设计与实现

于萧榕, 席 屏, 黄健荣

(江苏科技大学 计算机科学与工程学院, 江苏 镇江 212003)

摘要: 视频监控系统在日积月累的监控过程中会产生大量的视频信息, 如何在海量视频中基于内容快速检索出需要的信息成为亟待解决的问题; 文章在视频检索过程中, 将视频图像的 SIFT 特征提取匹配与 MapReduce 并行模式相结合, 同时利用 LSH 映射对视频图像进行分组并存储在 HBase 数据库中, 在配合并行分析的同时减少了计算量; 最后实验证明, 采用 MapReduce 并行模式提取图像 SIFT 特征值, 时间会随着集群节点的增加而不断降低, 并最终趋于稳定; 由此可知, 对基于内容的视频检索采用并行化分析可以显著提高效率, 实现了 Hadoop 架构在视频分析方面的应用拓展。

关键词: 分布式视频检索; Hadoop HBase; 尺度不变特征转换; 局部敏感哈希

Design and Implementation of Distributed Retrieval in Video Monitoring System

Yu Xiaorong, Xi Ping, Huang Jianrong

(Jiangsu University of Science and Technology, Zhenjiang 212003, China)

Abstract: Video surveillance system will produce a lot of video information in the monitoring process over time, how to search needed information quickly, which based on content, in the massive videos has become an urgent problem to be solved. This paper, during the video retrieval process, combines the SIFT feature extraction and matching of the video image with the MapReduce parallel mode, and meanwhile uses the LSH mapping to divide the video image into groups and stores it in the HBase database, which reduces the amount of calculation in conjunction with the parallel analysis. Finally, the experiments show that by using MapReduce parallel mode to extract the SIFT extraction characteristic value of image, time will continue to decrease with the increase of the cluster nodes, and finally tend to be stable. Therefore, to the video retrieval based on content, it can improve the efficiency remarkably by using the parallel analysis, and it realizes the expansion of the application of Hadoop architecture in the video analysis.

Keywords: distributed video retrieval; HDFS HBase; scale-invariant feature transform; locality-sensitive hashing

0 引言

视频监控系统在安全防护和公安刑侦等方面有着广泛的应用, 当监控系统检测到“危险”行为, 为了事后考证都对当时进行了视频录像, 日积月累就形成了大量的视频信息。目前常用的监控视频检索还是基于文本标签, 文本内容多是通过视频内容的主观理解人为加入的, 存在主观片面性^[1]。检索时无论是通过定义的标签查找, 还是通过加派人力一一检查, 检索效率都不高, 并且查准率和查全率都无法保证, 这对监控视频资源是极大的浪费。

目前基于内容的图像检索 CBIR (Content-based image retrieval) 技术在应用方面获得了巨大的发展, 人们只需传入检索图像, 就可在图像库中获得相近的图像信息。并且随着图像特征提取与匹配算法的不断演变, 对于图像内容细节的要求也逐渐趋于主观判断的结果^[2]。实际上, 一般长度的视频包含的图像帧数量非常大, 并且相邻帧的内容差异不大, 如果再结合复杂的图像特征提取与匹配计算, 逐帧解析查找视频所消耗的时间将无法预计, 在应用中没有使用价值。

分布式并行计算是缓解运算时间过长的有效手段, 将海量视频数据分布到云端计算与存储^[3], MapReduce 是开源项目

Hadoop 中的并行编程模型。若对视频关键帧的匹配过程运用并行编程^[4], 可以在保证匹配质量的前提下缩短计算时间, 从而满足基于视频内容的检索需要。

1 视频分布式检索架构

对预警视频的检索是建立在分布式的处理平台上, 先对预警视频提取可以替代视频内容的若干关键帧, 再对海量关键帧提供分布式存储和计算 (特征提取与匹配), 建立关键帧与索引的对应关系, 以便图像检索时通过提取图像特征可以准确定位到关键帧以及对应的预警视频, 完成基于内容的视频检索。处理过程采用集群并发计算, 缩短了运算时间, 集群灵活的结构也使系统具有良好的扩展性和容错性。

如图 1 所示, 预警视频分布式检索从层次上主要分为数据层、处理层、检索层共 3 个层次^[5]。

数据层是预警视频的来源, 通过监控设备直接获得监控视频并传给硬盘录像机进行暂存和分析。如果分析结果需要预警, 就对监控视频中相关部分进行截取, 并转发给媒体数据库进行保存, 同时将预警视频发送给媒体预处理服务器, 为检索处理做准备^[6-7]。

处理层是预警视频分布式检索的核心, 主要包括处理预警视频以及分析图像检索视频等功能^[8]。当进行预警视频处理时, 先对预警视频提取关键帧, 再通过 MapReduce 并行模型进行关键帧的 SIFT 特征提取, 再通过 LSH 对特征向量进行映射并创建索引, 最后将映射关系存储在分布式数据库

收稿日期: 2014-11-17; 修回日期: 2015-01-09。

作者简介: 于萧榕(1975-), 实验师, 硕士, 主要从事软件工程、云计算、智能信息处理方向的研究。

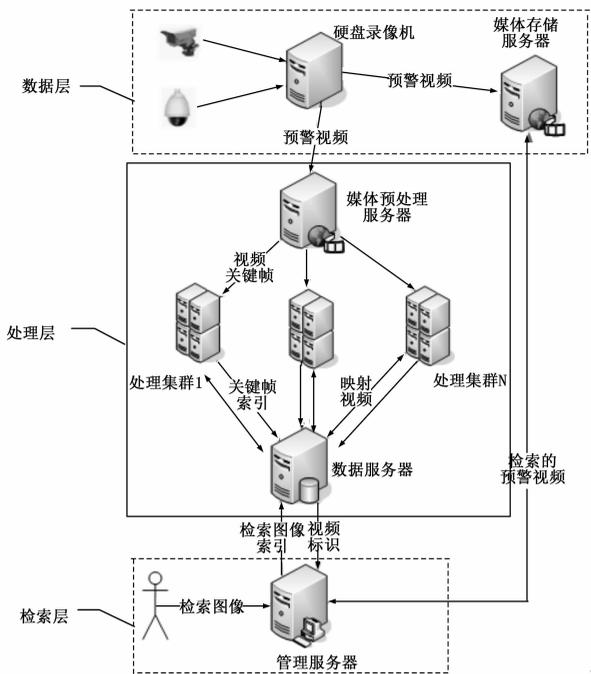


图 1 分布式检索架构图

HBase, 以供查询人员检索。预警视频处理的流程如图 2 所示。

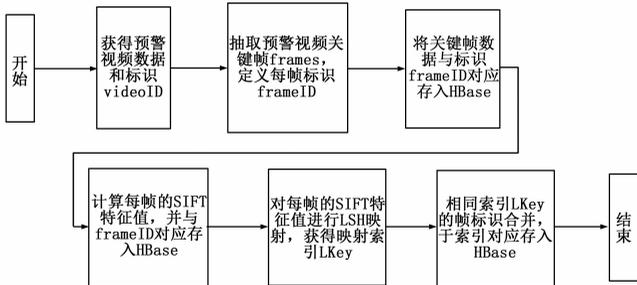


图 2 预警视频处理流程图

当根据图像进行检索时, 在管理服务器对检索图像进行 SIFT 特征提取及 LSH 映射, 通过获得的索引查询数据库 HBase 中对应的关键帧集, 再通过 MapReduce 并行计算每个关键帧与检索图像的相近度, 按照相近度由高到低的顺序返回给查询人员。图像检索的流程如图 3 所示。

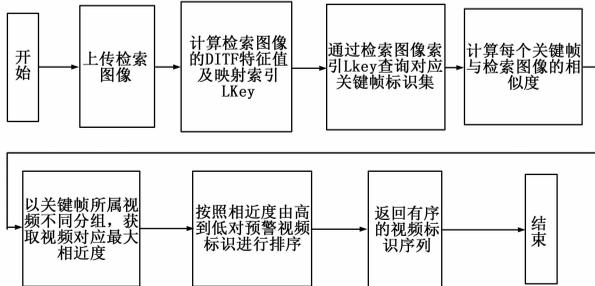


图 3 图像检索流程图

检索层是为用户提供检索服务的终端, 是基于分布式处理层的一个应用平台。用户通过功能操作将本地图像进行上传,

经过图像特征计算和映射后, 将检索请求与索引值发送给集群进行检索计算。当集群返回视频序列后, 向媒体数据库请求获得视频数据, 并将最终结果返回给检索用户^[9]。

2 基于 Hadoop 的视频分布式检索实现

2.1 基于 HBase 的视频检索信息存储

HBase 是基于 Hadoop 的开源数据库, 用于存储粗粒度的结构化数据, 实现了高可靠性、高性能、列存储、可伸缩、实时读写的数据库系统。HBase 以表形式存储数据, 每个表由行和列组成。

在物理模型中, HBase 是按照列存储的稀疏矩阵, 由行键、时间戳和列 (包括所属列族) 唯一确定元素。相同行键和列默认获得最新时间戳的元素值, 所有的更新和删除操作都是通过新增不同时间戳的数据来完成, 物理上所有数据都存储在 HDFS 上。

预警视频分布式检索中用 HBase 数据库负责对数据进行分布式存储和保持数据的一致性。按照需要主要划分为两个表: 第一个表是视频关键帧数据表, 用于存储关键帧标识 frameID 与相应的图像数据 frameData、SIFT 特征向量 siftVector。按照 HBase 结构存储关键帧数据表, 其中关键帧标识 frameID 作为行键, 图像数据 frameData 和 SIFT 特征向量 siftVector 作为列族 frame 的两个列。表 1 表示关键帧数据表结构。

表 1 关键帧数据表结构

Row Key (frameID)	Time Stamp	frame	
		frameData	frameVector
Id1	t1
..

为了体现关键帧与视频的对应关系, 关键帧标识定义为视频标识 + “.” + 关键帧偏移量。例如: 视频标识为 001, 关键帧在视频中的偏移量为 23190, 则关键帧标识为 001. 23190。

第二个表是索引表, 用于存储 LSH 构建索引 LKey 与相近关键帧标识集 frameIDs 的对应关系。按照 HBase 结构存储索引表, 其中索引 LKey 作为行键, 相近关键帧标识集 frameIDs 作为列族 frames 的唯一列。表 2 表示索引表结构。

表 2 索引表结构

Row Key (LKey)	Time Stamp	frameIDs
		frame
Id1	t1
..

相近关键帧标识集 frameIDs 的内容用分隔符进行分隔。为了方便分析, 相同视频的关键帧排列在一起用 “,” 分隔, 不同视频关键帧集之间用 “;” 分隔。例如: 001. 23190, 001. 23198, 001. 23205; 002. 23002, 002. 23015。

2.2 基于 MapReduce 的视频关键帧特征提取:

每个预警视频为完整表现内容, 会提取出多个关键帧。每个关键帧采用检测局部特征的 SIFT 算法获得特征点描述子, 描述子是个 128 维的方向参数。为了方便在高维数据上进行近似查找, 采用 LSH 技术使相近关键帧的描述子能够映射到邻

近位置^[10]。

1) SIFT 特征提取

尺度不变特征转换 SIFT (Scale-invariant feature transform) 是一种检测局部特征的算法, 该算法通过求一副图像中的特征点及其有关 scale 和 orientation 的描述子得到特征。特征具有尺度不变性, 算法具有很强的匹配能力。整个算法包括以下几个部分: 创建尺度空间; 检测 DOG 尺度空间极值点; 筛选特征点; 为特征点指定方向参数。

为了使算子具备旋转不变性, 利用关键点领域像素的梯度方向分布特性来指定方向参数。在关键点为中心的领域窗口内采样, 根据采样点的梯度方向画出直方图, 梯度直方图的范围是 0~360°, 其中每 10° 一个柱, 总共 36 个柱, 随着距中心点越远的邻域其对直方图的贡献也相应减小。直方图的峰值代表了该关键点处领域梯度的主方向, 即该关键点的方向。采样点 (x, y) 的梯度模值和方向公式为:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \text{atan2}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y)))$$

其中: L 为关键点所在图像尺度。

生成关键点描述子。通过旋转坐标轴为关键点的方向, 使关键点具有旋转不变性。以关键点为中心获取大小为 16 * 16 的窗口, 将该窗口划分为 4 * 4 个子窗口, 计算每个子窗口在 8 个方向上的梯度方向直方图, 该直方图形成一个尺度为 8 的向量, 将 16 个子窗口按序进行连接, 得到一个长度为 128 维的向量, 即关键点描述子。

2) 局部敏感哈希 LSH:

局部敏感哈希 LSH (Locality-sensitive hashing) 是一种用于海量高维数据的近似最近邻快速查找技术。通过映射变换操作, 原始数据集被分成了多个子集合, 每个集合中的数据间是相邻的。查找时只需将查询数据进行哈希映射得到其桶号, 再取出该桶号对应所有数据进行匹配即可找到相近数据。其中关键在于 hash functions 的选取, 这些 hash functions 需要满足以下两个条件:

- (1) 如果 $d(x, y) \leq d1$, 则 $h(x) = h(y)$ 的概率至少为 $p1$ 。
- (2) 如果 $d(x, y) \geq d2$, 则 $h(x) = h(y)$ 的概率至多为 $p2$ 。

其中 $d(x, y)$ 表示 x 和 y 之间的距离, $d1 < d2, h(x)$ 和 $h(y)$ 表示对 x 和 y 的 hash 变换。

对关键帧进行 SIFT 特征提取和 LSH 映射变换都只局限于当前帧, 每帧可以认为是独立的计算单元。在 MapReduce 并行模式中 Map 阶段同时处理多个关键帧, 可以提高特征提取的效率。再在 Reduce 阶段通过 LSH 映射将相近关键帧合并, 并与映射索引对应进行存储, 为视频检索提供了基础对应关系。

Map 阶段中单独处理一个关键帧, 以关键帧标识 frameID 与关键帧数据 frameData 的键值对作 <frameID, frameData> 为输入, 如图 4 所示。

Reduce 阶段中, 获得输入键值对 <Lkey + VideoID, frameID> 列表, 由于所有键值对 Lkey 都相同且按照 VideoID 进行了排序, 循环读取所有键值对的 value 值组成关键帧标识

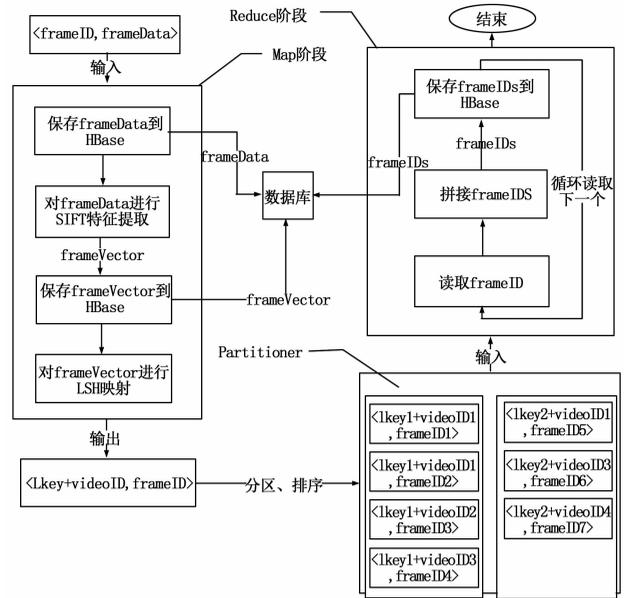


图 4 特征提取 MapReduce 过程

串 frameIDs, 当与前一个 value 值的 videoID 相同, 用分隔符 “,” 连接, 否则用分隔符 “;” 连接。frameIDs 完成后保存到 HBase 的索引表中, 以 Lkey 作为行键值。

2.3 基于 MapReduce 的检索图像特征匹配

检索图像上传后会进行 SIFT 特征提取以及 LSH 映射, 根据获得的索引值查询 HBase 中索引表, 找出对应的关键帧标识串。检索图像特征值依次和关键帧特征值进行 SIFT 匹配, 若相近度大于设定阈值, 则认为该视频符合检索要求。检索出的多个视频再按照匹配相近度的值由大到小进行排列, 并将有序视频列表作为结果返回给检索用户。

根据图像内容检索预警视频时, 计算所有关键帧与检索图像特征向量的相近度最耗费时间, 但是每个关键帧与检索图像的相近度计算是彼此独立的。通过 LSH 映射得到的关键帧与检索图像相近的概率高, 这样缩小了计算范围。再通过并行的 Map 阶段计算关键帧与检索图像 SIFT 特征向量的相近度, 大大缩短了运算时间。最后在 Reduce 阶段对相近度由高到低进行排序, 得到的视频标识有序序列即为查询的结果。

在进入 Map 阶段之前, 先对检索图像进行 SIFT 特征提取和 LSH 映射, 得到检索图像特征向量值 queryVector 和索引值 Qkey。再通过 Qkey 查询 HBase 中索引表获得对应关键帧标识串 frameIDs。考虑到同一视频会有多个关键帧与检索视频相近, 在分隔 frameIDs 时将属于一个视频的关键帧分到同一个 Map 阶段, 方便比较并取最大相近值作为视频的相近度值。

在 Map 阶段中, 获得输入键值对 <queryID+frameIDs_1, queryVector>, 其中 queryID 为检索图像标识, frameIDs_1 为属于同一视频的关键帧标识子串。循环获得关键帧特征向量值 frameVector, 并计算 queryVector 和 frameVector 的相近度, 如小于比例阈值, 再比较是否大于前一帧相近度, 保存最大的相近度。循环结束后输出以 queryID 与最大相近度 Similar_Max 组合为 key, 以对应视频标识 videoID 为 value 的键值对 <queryID+Similar_Max, videoID>。

为了对相近度进行排序, 需要重写 Partitioner 的 getParti-

tioner () 方法。在自动根据 Map 的 key 排序的情况下, 依然根据 queryID 进行分区, 保证进入 Reduce 阶段的键值对列表都属于同一视频。

在 Reduce 阶段, 为了减少数据传输量, 可以设定返回的结果集个数, 只返回 videoID 列表中的前 N 个。输出的键值对为 $\langle \text{queryID}, \text{videoIDs} \rangle$ 。

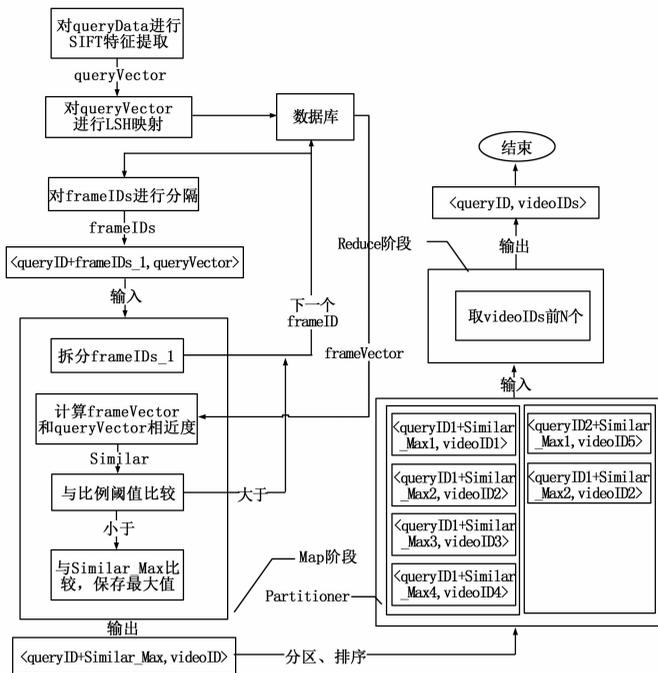


图 5 特征匹配 MapReduce 过程

3 实验分析

基于 MapReduce 分布式视频检索的性能主要从关键帧 SIFT 特征提取的速度以及图像检索的查全率、查准率两个方面进行分析。

3.1 特征提取实验分析

实验数据为 10 段监控视频, 视频长度约在 3 至 5 分钟不等, 视频分辨率统一为 176×144 。分布式实验平台有 3 个节点 (包括: Slave01、Slave02、Slave03), 每个节点的 cpu 核心处理能力之比约为 1: 1: 1。分别统计一个节点、两个节点和集群 (包括 3 个节点) 提取所有视频关键帧的 SIFT 特征所花费的时间, 具体结果如图 6 所示。

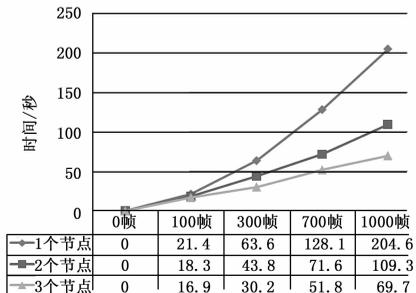


图 6 不同节点数下处理关键帧的性能对比

图 6 横坐标表示关键帧的个数, 纵坐标表示处理时间。刚开始处理时, 由于分布式平台启动需要检查集群环境和节点状态, 所以和单节点计算相比没有明显优势。随着处理帧数的不断增加, 多节点计算在时间上就有了明显的优势, 并逐渐趋于稳定。

3.2 特征匹配实验分析

本次实验的特征匹配采用在 LSH 索引上采取随机采样的方式, 将多维向量转换为 0 或 1 字符串, 再随机选取其中 M 位拼接后作为 LSH 索引。查全和查准率都和索引长度 M 有关。通过设定 $M = 6$ 和 $M = 10$ 分别进行实验, 实验结果如表 5、表 6 所示。

表 5 M = 6 时, 查全率和查准率

关键帧数	100	300	700	1 000
查全率 (%)	56%	39%	33%	30%
查准率 (%)	92%	93%	86%	90%

表 6 M = 10 时, 查全率和查准率

关键帧数	100	300	700	1 000
查全率 (%)	34%	23%	20%	21%
查准率 (%)	72%	74%	67%	71%

如果 M 值较大, 则查询速度快, 但是查全率不高; 如果 M 值较小, 则查询速度慢, 但查全率较好。需要通过实验确定 M 的取值, 来保证查询速度和查全率的平衡性。

4 结语

基于内容的视频检索是当前信息检索的研究热点, 由于计算的复杂性, 在性能的优化方面还有很大的提升空间。文本结合 MapReduce 并行模式, 从并发的角度考虑解决计算性能问题, 在保证计算精度的前提下, 以低廉的集群设备完成大型运算, 具有实用价值, 也是对监控系统检索领域的一个拓展。

参考文献:

- [1] 陈强国, 胡瑞敏, 张 华, 等. 面向监控录像的视频检索系统设计 [J]. 电声技术, 2011, 35 (11): 60-62.
- [2] 王 梅, 朱信忠, 赵建民, 等. 基于 Hadoop 的海量图像检索系统 [J]. 计算机技术与发展, 2013 (1): 204-208.
- [3] 徐 兵. 基于网络的智能定时视频监控系统设计 [J]. 计算机测量与控制, 2011, 19 (8): 36-38.
- [4] 张佳琦. 基于 Map-Reduce 的图像匹配算法研究与实现 [D]. 西安: 西安电子科技大学, 2013.
- [5] 彭 俊. 基于内容的图像分布式处理平台的设计与实现 [D]. 武汉: 华中科技大学, 2013.
- [6] 朱 琳. 基于云计算的分布式校园视频监控系统的的设计 [J]. 计算机测量与控制, 2013, 21 (10): 2676-2679.
- [7] 朱丽英, 梁 辰, 张鸿洲. 基于运动目标的监控视频检索方法 [J]. 计算机应用与软件, 2011, 28 (12): 96-98.
- [8] 李 军, 李松斌, 王玲芳, 等. 一个基于 P2P 技术的分布式视频检索系统 [J]. 微计算机应用, 2010, 31 (8): 35-42.
- [9] 郑 焱, 肖碧宇. 基于 MPEG-7 的视频检索系统的设计与实现 [J]. 计算机系统应用, 2010, 19 (10): 1-6.