

Big Data Storage Method for Seabed Monitoring Video Based on Hbase

Chuan Wang¹, Guosun Zeng¹, Chunling Ding², Rufu Qin^{3,4}

¹Department of Computer Science and Technology, Tongji University, Shanghai

²College of Chemical Science and Engineering, Tongji University, Shanghai

³School of Ocean and Earth Science, Tongji University, Shanghai

⁴State Key Laboratory of Marine Geology, Tongji University, Shanghai

Email: wangchuan_tj@tongji.edu.cn, se2003@sina.com

Received: Jul. 10th, 2019; accepted: Jul. 22nd, 2019; published: Jul. 29th, 2019

Abstract

Submarine observations produce large-scale, sparse video data, and current storage schemes are less efficient. To this end, a big data storage method for seabed monitoring video based on Hbase is proposed. First, the moving object detection method is used to segment the video to high-density and low-density parts, so the low-density parts can be greatly compressed, at the same time, feature extraction is performed on the dense content part to facilitate content-based video retrieval. Then, a big data table is designed to uniformly store the above two parts of the video, it can dynamically add compressed information columns when inserting low-density video parts and feature columns when inserting high-density video parts, also we use the trailing portion of the row-key to identify them, which facilitates query and filtering. Experiment results show that the proposed method can save nearly 75% of the storage resources compared with traditional methods. Besides, it can also provide feature-based retrieval, which greatly facilitates the management of video data.

Keywords

Submarine Monitoring, Video Data, Hbase, Large Database Storage, Video Retrieval

基于Hbase的海底监测视频大数据存储方法

王川¹, 曾国荪¹, 丁春玲², 覃如符^{3,4}

¹同济大学计算机科学与技术系, 上海

²同济大学化学科学与工程学院, 上海

³同济大学海洋与地球科学学院, 上海

⁴海洋地质国家重点实验室(同济大学), 上海

Email: wangchuan_tj@tongji.edu.cn, se2003@sina.com

摘要

海底观测会产生规模大、稀疏性强的视频数据，目前的存储方案效率较低。为此，本文提出一种基于Hbase的海底监测视频大数据存储方法。首先，采用运动目标检测方法对视频中的稀疏空白部分和内容密集部分进行分段，对稀疏空白段进行重度压缩，并对密集内容部分进行特征提取，为基于内容的视频检索做准备。接着，设计一种能统一存储上述两部分视频的大数据表，它能在插入稀疏视频段时动态添加压缩信息列，在插入密集内容视频时动态添加特征列，并用行键键尾对两种视频进行了区分，方便查询过滤。实验分析表明：本文方法相比传统方式能节约近75%的存储资源，并能实现基于特征的检索，极大的方便了视频数据的管理。

关键词

海底监测，视频数据，Hbase，大数据存储，视频检索

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

海底视频监测是海洋观测中最直观有效的手段之一，通过海底观测网的持续供能和信息传输优势[1]，可以将海底最直接的影像数据实时全天候地传输至岸上[2]，为海洋科考人员提供重要信息[3]。然而海洋空间巨大，监测将产生规模空前的视频数据，使得现有的存储系统难以负担，导致有价值的视频数据可能丢失，严重影响开发和利用海洋事业的发展。

视频存储技术一直是计算机学科的一个研究热点。早期的视频数据主要采用文件存储的方式，例如将视频存入到操作系统中的文件中，并利用视频的拍摄时间、地点等信息对文件和目录命名。这种方式简单直接，但随着文件数量的增多，文件目录层数也越来越多，视频检索变得很困难。为了方便查找，使用数据库来存储和管理视频文件的方法开始流行起来。通过将视频的特征信息提取出来存放到数据库中，并设置一个字段记录存储源视频文件的路径，使得视频检索得以快速进行。随着视频数据量的进一步扩大，以前采用单机存储也慢慢演化为采用网络分布式存储，甚至数据中心大数据存储。例如，在交通视频监测领域，刘高辉[4]等提出了采用分布式数据库 Hbase 和分布式文件系统 Hdfs 相结合的视频存储方案，即先将原始视频上传到 Hdfs，并进一步对视频内容进行摘要提取，将视频的元数据和摘要一同存入 Hbase 数据表，使得视频检索可以在 Hbase 表中快速进行。然而，Hdfs 倾向于存储大文件，因此该方法无法开展细粒度的视频检索。为解决这一问题，江冕[5]等针对文件系统存储视频效率低、粒度大的问题，提出了一种专用的磁盘逻辑结构，将视频和对应的索引信息直接写入该结构，实现了视频的秒级检索。操顺德[6]等在该文献工作的基础上，设计了专门的分布式视频存储架构，使系统在满足细粒度检索的同时，支持海量数据的并发读取，但是这种专门的体系结构使用起来不方便，不易推广。文献[7]利用 Hbase 大数据存储来存储视频数据，设计了一种面向海量小视频的存储方案，将连续的安防监控以 10 分钟为单位进行存储，降低了视频的连续读写频率，在保证数据库中数据的实时性和视频准确检索方面取

得了较好的效果。然而，在海洋视频监控领域，由于海底视频摄像十分困难，所以海洋视频存储研究的进展也相对缓慢，纵观目前的文献资料报道，目前采用的仍然是传统的文件存储方法。显然，无法达到海量、高效、持续存储的目的。近年，海底监测网络已列为国家长期战略项目[3]，是国家开发和利用海洋工程的重要支撑。因此，研究针对海底监测视频的大数据存储方法是必要和紧迫的。

本文通过分析海底监测视频的实际拍摄环境，关注海底监测视频内容存在稀疏特性，为了有效存储视频中的稠密部分，同时压缩稀疏空白视频，首先设计基于运动目标检测的视频分段算法，将视频数据分段，然后给出基于 Hbase 大数据库的视频数据存放方法。该方法可以满足不断增长的视频数据的存储，同时能够实现不同视频的快速准确检索。

2. 海底监测视频和存储需求

2.1. 海底监测视频流处理框架

海底监测信息系统框架如下图 1，由 4 个部分组成。1) 视频采集端：由诸多安置在海底的摄像头和视频编码器组成，对某一固定区域进行监视，并将影像通过海底光缆传输至海岸基站。2) 视频预处理服务器：接受视频数据并做视频入库前的预处理工作，包括视频分段、压缩和特征提取，同时将处理后的数据提交给 Hbase 数据库。3) Hbase 数据库：组织和存储海量视频数据，提供高效的视频数据管理机制。4) 视频应用检索服务器：响应用户的视频查询请求，并传递给 Hbase 去执行，将结果返回给用户分析。

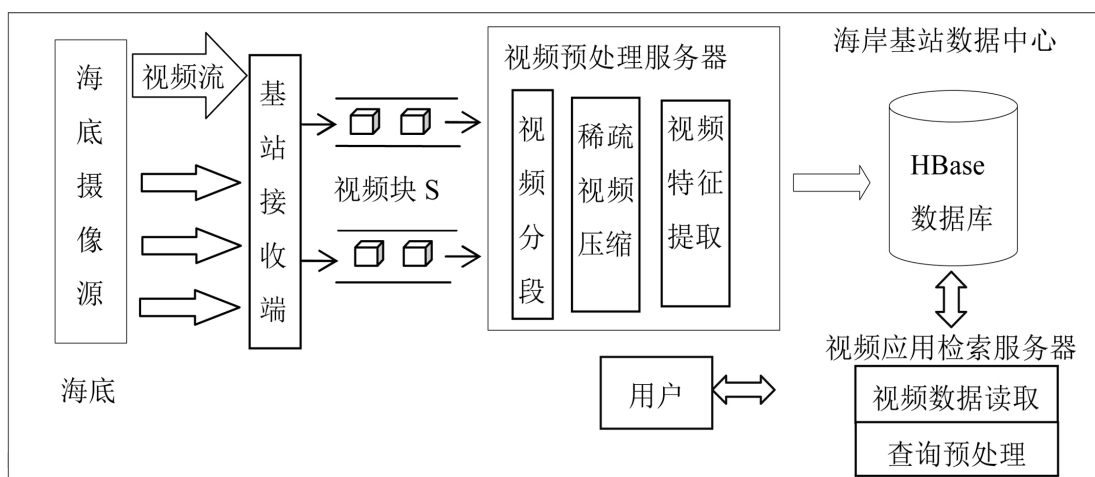


Figure 1. Submarine monitoring video stream processing framework

图 1. 海底监测视频流处理框架

2.2. 海底监测视频特征

视频的稀疏性

海底监测的对象多种多样，可以是位于大洋水层或海底的生物、海底重要基础设施、以及发生在海底的自然现象等。由于海底环境空旷，监控区域只是整个环境空间的一小部分，在有限的区域内能捕获到监测目标的次数是十分稀少的。以监控鱼群为例，当鱼群游到监控区域并驻足时，监控装置会拍摄到大量鱼类画面，并随着鱼的游动不断变换，而当鱼群随着洋流离去，视频中的信息量明显变少，大部分是无价值的背景信息。多数情况下，鱼群会在海水中四处活动，只是偶尔出现在监控区域内，因此视频中的内容有时是稠密的、大部分是稀疏的。我们把这类没有实质内容，或者没有太多价值的视频认为是稀疏视频。下图 2 描述了一段典型的稀疏视频 S，从左往右代表时间先后顺序。

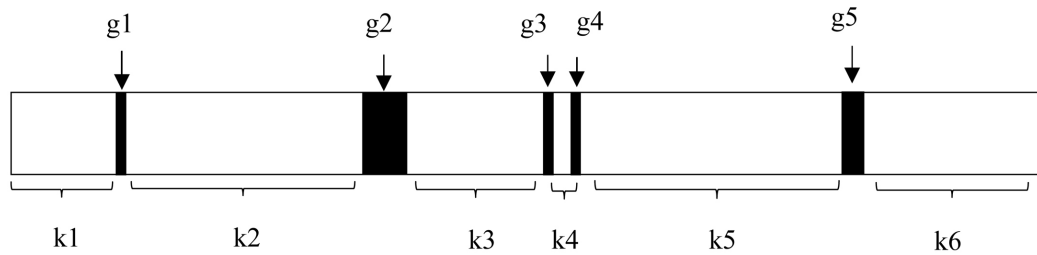


Figure 2. A segment sketch map of video S

图 2. 一段视频 S 分段示意图

在图 2 中, 黑条(用 g 标识)表示视频中信息密集的部分, 它在一段视频序列中的出现具有随机、不均匀、持续时间短等特征。相反, 白条(用 k 标识)表示视频中的信息稀疏部分, 它占视频序列的绝大部分。很明显, 如果将图 2 所示视频的各部分平等对待, 全部存入数据库, 那么信息使用价值不能区别对待, 有用信息将会淹没在大量无用信息中, 而且还会造成大量的存储冗余, 影响视频检索效率。因此要对海底监测视频进行分段处理, 视频中信息密集和稀疏部分的最大区别在于视频内运动量的变化, 当对象出现或发生时, 视频画面会发生明显的改变, 因此可以利用运动目标检测的方法对视频进行分段, 将在后面 4.2.1 节中讨论。

2.3. 海底监测视频的存储需求

为了能持续和长期存储海底监测视频流, 必须针对视频稀疏性进行高效压缩, 以免浪费不必要的存储资源, 其次必须使用大数据存储平台, 支持海量数据存储, 且随着视频的不断存入, 系统可以平滑扩容。为了能更好地存储分段后的视频数据, 不采用粒度大且不易管理的文件存储结合数据库存储的方式, 选用 Hbase 大数据数据库统一存储全部的视频数据。为了方便视频流的存入, 需要将时间上相邻的视频数据, 在存储装置上也位置相邻, 同时对视频流中密集内容部分进行特征标记, 使得系统既支持播放完整视频, 又可以只播放密集内容视频。为了便于视频数据的分析和利用, 需要在视频存储前进行特征识别, 将提取到的特征和视频关联保存, 使用户可以基于视频特征进行快速检索。

3. 海底监测视频存储模型

3.1. 视频存储模型

3.1.1. 基本概念

定义 1 密集内容视频片段: 密集内容视频片段是在原始视频中, 监测目标出现且连续变化的一段视频画面, 它可以抽象表示为 $g = (id, t, lo, le, co, sh, te, se)$, 其中 id 是拍摄该段视频的摄像源编号, t 、 lo 是视频拍摄的起始时间和地点, le 表示 g 的持续时长, co 、 sh 和 te 代表视频 g 中提取到的颜色、形状和纹理特征, se 表示 g 中内容的语义特征。

定义 1 中, id 、 t 、 lo 、 le 是 g 本身固有的信息, 可以在视频上传时直接获知, 而 g 对应的视频特征, 需要预先作提取操作。由于目前的视频特征检索技术大多是基于视频中的图像的检索, 考虑到视频时间维度和更高层内容检索仍处于初级阶段。因此, 我们认为视频特征 co 、 sh 、 te 、 se 就是 g 中的图像特征。考虑到 co 、 sh 、 te 属于图像的底层特征, 计算机提取它们比较简单且快速, 所以在存储视频段 g 之前, 首先对 g 进行图像特征检测, 然后将检测到的特征提取出来, 并与 g 进行关联保存, 使得系统可以根据底层特征对视频进行检索。而 se 特征的提取非常复杂、耗时多, 不在视频存储时同步提取, 对此本文在 5.3 节给出了一种 se 特征检索写回策略。在实际存储时, g 中包含的各个属性数据将映射到 Hbase 数据表

中, 具体实施方法在 4.3.4 节讨论。

定义 2 稀疏空白视频片段: 稀疏空白视频片段是在原始视频中, 大多只有背景且几乎没有监测目标出现的一段视频画面, 它可以抽象表示为 $k = (id, t, lo, le, p, n)$, 其中 id 、 t 、 lo 和 le 分别表示 k 的摄像源编号、起始时间、地点和持续时长, p 是 k 中的一个图像组画面(GOP, Group of Picture), n 是该段视频的解压参数。

为了实现视频的完整播放, 我们不是直接丢弃稀疏空白视频段, 而是选择大力度地加以压缩, 将其中包含背景、起止时间等有用信息保留下来。定义 2 中, id 、 t 、 lo 、 le 是视频本身固有的, 可以在视频上传时直接获知。 p 和 n 分别用来保留视频 k 压缩后的信息, 具体的压缩和解压过程将在 4.3.3 详细讨论。由于稀疏空白视频段包含的有用信息量特别少, 对其进行特征提取得不偿失, 故模型 k 中没有特征属性。同样地, 在实际存储时, k 中的各个属性数据将映射到 Hbase 数据表中, 具体方法在 4.3.4 节讨论。

定义 3 完整海底监测视频集: 所有密集内容视频段和稀疏空白视频段按照拍摄起始时间顺序组合在一起的视频, 是完整海底监测视频集, 它可以抽象表示为 $F = \{v_1, v_2, \dots, v_n\}$, 其中 v_i 是一段视频数据的抽象, $v_i = g$ 或 k , 记为 $v_i = v_{ig}$ 或 v_{ik} 。

定义 4 视频段标识: 视频段标识是唯一指向一段视频 v 的指针, 它可以抽象表示为 $r = (id, t, f)$, 其中 id 是拍摄该段视频的摄像源编号, t 是视频拍摄的起始时间, f 是视频类型的标识, 用于指示当前视频 v 是 g 还是 k 。

3.1.2. 索引定义

定义 5 主索引 I_R : 是视频数据 $v \in F$ 的一个二元组集合 $\{(r_1, v_1), (r_2, v_2), \dots, (r_i, v_i), \dots, (r_n, v_n)\}$, 其中 r_i 是视频集 F 的视频标识集合中的一个标识, v_i 表示一段视频数据。

定义 6 辅助索引 I_{CO} : 是视频数据 $v \in F$ 的一个二元组集合 $\{(co_1, v_1), (co_2, v_2), \dots, (co_i, v_i), \dots, (co_n, v_n)\}$, 其中 co_i 是视频集 F 的视频图像颜色特征集合中的一个颜色特征, v_i 是一段视频数据的抽象。

类似地, 我们可以定义形状、纹理和语义特征的辅助索引 I_{SH} 、 I_{TE} 和 I_{SE} 。

4. 基于 Hbase 的视频存储模式和实现

4.1. Hbase 数据库和存储优势

Hbase 是目前主流的分布式大数据库系统, 其原型来源于 Google 的 BigTable [8], 现已被广泛应用到各个行业[4] [7]。传统视频数据的存储常常采用文件和数据库相结合的方式, 但本文研究海底监测视频的存储, 视频中密集内容片段大多持续时间短、出现频率低, 分段后呈现为许多独立小视频的形态, 而对于稀疏空白段, 我们进行压缩存储, 用于完整视频的播放。因此, 一条视频流在实际存储时会被存放为两种类型、大小不一、数量巨大的视频碎片。如果仍选择用文件的方式去一一存储这些视频碎片, 会给视频数据的组织和管理造成巨大困难, 同时也会造成文件冗余和数据、元数据一致性维护难问题。为此本文不再采用传统存储方式, 而是选择用 Hbase 大数据库来存储和维护海底监测视频。

4.2. 视频存储实现方案

4.2.1. 视频分段方法

为了方便论述, 先对视频分段的目标进行描述, 对于一段视频 S , 可以将 S 分成两个子视频段集合 G 和 K , 其中 G 是 S 中的密集内容视频片段组成的集合, K 是 S 中稀疏空白视频片段组成的集合, 例如对于图 2 中的视频 S , 分段后 G 包含 $\{g_1, g_2, g_3, g_4, g_5\}$, K 包含 $\{k_1, k_2, k_3, k_4, k_5, k_6\}$ 。为了划分 g 和 k , 我们借助运动目标检测的方法, 在诸多视频运动目标检测方法中, 背景消减法对固定位置摄像源拍摄形

成的视频有很好的运动检测效果[9]，海底监测视频符合该特征，其思想是对视频背景进行建模得到背景帧，然后将后续帧与背景帧作差，用得到的差值衡量视频的运动变化量。本文的分段方法在背景减法的基础上进行，计算公式如下：

$$f_B(x, y) = \frac{1}{q} \sum_{k=1}^q f_k(x, y) \quad (1)$$

$$C(f_1, f_2) = \sum_{x=1}^m \sum_{y=1}^n |f_1(x, y) - f_2(x, y)| \quad (2)$$

假定视频帧大小为 $m \times n$ ，公式(1)中 $f_k(x, y)$ 和 $f_B(x, y)$ 分别表示视频帧 k 和背景帧 B 在像素点 (x, y) 处的灰度值，取值范围 $0 \sim 255$ ，这里采用了平均值法[9]计算背景，即选取连续 q 个帧在点 (x, y) 的灰度平均值作为背景帧在该点的灰度值。公式(2)中 $C(f_1, f_2)$ 指两帧 f_1, f_2 中所有像素点的灰度差之和，为了避免正负抵消添加了绝对值，例如 $C(f_i, f_B)$ 是当前帧 f_i 与背景帧 f_B 的灰度差， $C(f_i, f_{i-1})$ 是相邻两帧的灰度差，这里给灰度差一个阈值 T ，当当前帧与背景帧的灰度差大于 T 时，我们认为视频画面出现运动物体。 T 通过选取一段实际稀疏空白视频测试得到，具体参见文献[10]，本文取 $T = 6$ 。

下面给出视频分段算法，为了加快分段速度，以图像组 gop 中的 I 帧为单位进行运动检测：

算法 1: 视频分段算法 Video Segment Algorithm，简称为 VSA

输入：一段实际海底监测视频 S ，灰度差阈值 T ，提取背景的所需帧数 q ，连续度 c

输出：密集内容视频集合 G ，稀疏空白视频集合 K

VSA()

```

{   G_Gops ← ∅, K_Gops ← ∅;           //初始化
    fB ← build_background(S, q);      //用公式 1 构建视频背景的灰度矩阵
    P ← cut_gop(S);                   //将 S 划分为 gop 集合 P
    forall p ∈ P:
    {   fi ← extract_iframe(p);        //提取 p 中的 I 帧，并转为灰度矩阵
        d ← subtract(fi, fB);        //利用公式 2 计算灰度差
        if(d > T)
            G_Gops ← G_Gops + {p};    //将处于运动的 gop 加入 G_Gops
        else
            K_Gops ← K_Gops + {p};    //将相对静止的 gop 加入 K_Gops
    }
    G ← merge_gop(G_Gops, c);         //将 G_Gops 中连续个数超过 c 的 gop 合并为视频 g 并放入 G
    K_Gops ← drop_gop(K_Gops);        //将 K_Gops 中剩余的 gop 放入 K_Gops
    K ← merge_gop(K_Gops);
    return G, K;
}

```

算法 1 中的 gop 连续是指两个 gop 在原始视频中相邻。为了减少因噪音或个别物体短暂出现产生的错误分段，设置了连续度 c ，即至少连续 c 个 gop 都是属于运动画面时，我们才将它视为一段密集内容视频段。 c 的值可以根据实际需要动态调整。

4.2.2. 密集内容视频的底层特征提取

3.1.1 节讲到在关键视频段 g 存储之前，需要对其进行底层特征提取，以满足基于视频底层特征检索

的需要。由于关键视频段 g 的持续时间一般较短, 因此可以从 g 中抽取一个代表帧来反映视频中出现的监控目标, 然后从该帧图像中提取到特征即可作为 g 的特征。对于持续时间过长的 g , 不妨先将时长阈值设为 T' , 若 g 长度大于 T' 则对其切分处理。在代表帧的选择和图像特征提取方面, 目前图像领域已经有许多成熟的方法, 有兴趣的读者可以在该方面进行扩展, 这里为了方便讨论后续视频的特征存储和检索, 特意选择 g 中和背景帧灰度差最大的一帧作为代表帧, 例如以颜色特征的图像特征提取方法如下: 首先将整个颜色值域划分为 10 个区间, 分别对应黑、灰、白、赤、橙、黄、绿、青、蓝、紫, 然后对于一帧图像 I , 采用文献[11]的方式计算该帧图像的主色值, 将该值映射到一个颜色区间, 如“蓝”色, 那么字符“蓝”即可作为该帧图像的颜色特征, 并关联到 g 的 co 属性。对于其它底层视频特征, 方法类同, 不再赘述。

4.2.3. 稀疏空白视频的压缩

目前通用的视频编码器, 如 H.264/AVC 编码器中采用的是固定长度 Gop 的视频编码结构[12], 由于本文在视频分段时并没有破坏视频的原始编码结构, 所以一段稀疏空白视频 k 是由 n 个长度相同的 gop 顺序组合而成, 并且这些 gop 之间有大量时间上的冗余。为了压缩冗余, 本文仅保留其中第一个 gop (记为 p) 来记录视频背景信息, 用 n 来记录 k 中 gop 的个数。在解压缩时, 采用将 p 重复复制 n 次的方法得到 k 。对于视频编码采用的是可变长度的 gop 编码结构, 可以设:

$$n = \left\lceil \frac{k \text{的持续时长}}{p \text{的持续时长}} \right\rceil \quad (3)$$

同样将 p 重复复制 n 次得到 k 。虽然损失了时长的精确性, 但可以通过播放程序进行调整, 这里不再赘述。

4.2.4. Hbase 表设计和存储

前面讨论了密集内容视频 g 和稀疏空白视频 k 的数据准备工作, 本节阐述如何将其映射到 Hbase 数据表中。下面是设计的 Hbase 视频表, 考虑到视频数据不需要版本控制, 所以表中没有画出时间戳。

Table 1. Hbase video storage table

表 1. Hbase 视频存储表

Rowkey (I_R)	Data	Column family						
		Property						
	video	le	lo	n	(I)sh	(I)te	(I)se	(I)co
$r_1 (id_1 + t_1 + 1)$	v_{1g}	le_1	lo_1		sh_1	te_1	se_1	co_1
$r_2 (id_2 + t_2 + 1)$	v_{2g}	le_2	lo_2		sh_2	te_2	se_2	co_2
$r_3 (id_n + t_n + 0)$	$v_{3k(p)}$	le_3	lo_3	n_3				

每一段视频 v 会被存储到 Hbase 的一行记录中, 行键结构根据视频标识定义 r , 具体实现为“ $id + t + f$ ”, “+”代表字符串的连接, 当 $f = 0$ 代表当前视频段 $v = k$, $f = 1$ 代表 $v = g$, 这样在查询时可以利用键尾 f 快速过滤稀疏视频段。上表 1 以三行数据为例, 第 1、2 为密集内容数据行, 第 3 行为稀疏空白数据行。由于 Hbase 行键按字典有序, 因此这三行数据属于同一条视频流且时间上连续。每行的第一个列族为 $data$ 列族, 里面只有一个列 $video$, 在该列中存储原始视频, 例如第 1、2 行存储密集内容视频段 g , 第 3 行存储稀疏空白视频段 k , 这里实际存储的是 k 压缩后的 p 。正如 4.2.2 节讲到 g 可能因时间过长需要切分的问题, 这里也有一个要切分 g 的理由: 对于 Hbase 的一个字段($cell$), 虽然可以最高存储 2 GB

的数据量,但这样会使 Hbase 的效率很低,推荐的 cell 大小为 10 MB,所以本文将任何大小超过 10 MB 的 g 进行了切分并分行存储。第二个列族为视频属性列族 property,它包含 2 个固定的列: le、lo,因为两类视频记录都有时长和拍摄地点的属性。剩下的列根据需求动态增加,对于密集内容视频记录,在列族 property 下增加 co、sh、te、se 列用来存储视频图像特征,而对稀疏空白视频记录增加解压参数列 n。这体现了 Hbase 列的灵活性和允许表稀疏存储的特点。

由于 Hbase 是列式数据库,它会把同一列族下的数据存储在一起,使得基于某一列的查询速度更快,所以本文选择将数据和属性分到两个不同的列族,使之满足在海量监测视频中基于属性列的快速检索。当然也可以采用更快的方式,即建立二级索引,如 H-index 或 solr + Hbase 方案等,本文采用了 H-index 对特征列建立了二级索引,限于篇幅这里就不再展开。

4.3. Hbase 存储容量分析

将视频按照本文方法存入 Hbase 数据库,占用的存储空间可以用如下方法来计算。假设对一段码率为 r Mbps,持续时长为 le 秒的视频 S , S 中密集内容部分所占得比例为 c ,稀疏空白视频段的个数为 m ,一个 gop 长度约为 1 秒,则利用 Hbase 存储方式存储 S 占用的空间为: $(r \times le \times c + m \times r)/8\text{MB}$ 。然而,用传统文件方式存储 S 所占用的空间为 $r \times le/8\text{MB}$,两者相比得到的视频压缩率 z 为:

$$z = c + \frac{m}{le} \quad (4)$$

我们利用若干持续时长约为 1 小时(3600 s)的海底监测视频,通过回滚播放并求平均值的方式得到 $c = 0.25$, $m = 5$,代入公式(3),求得 $z = 0.251$,即本方法可以节省近 75%的存储空间。

5. 基于 Hbase 的视频数据检索

5.1. 自然回放某段时间的视频

自然回放某段视频就是选择一条视频流和起止时间进行播放,这是最常用的视频分析方法,下面给出自然回放的算法:

算法 2: 自然回放视频算法 Normal Play,简称为 NP

输入: 视频流 id, 视频起始时间 st, 视频结束时间 et, 视频数据库 My_Hbase

输出: 目标视频文件 v

NP()

```
{
  K ← ∅; G ← ∅; v ← ∅; //初始化
  use(My_Hbase); //打开数据库
  K ← Select_blank_video(My_Hbase, id, st, et); //找到所有稀疏空白视频
  G ← Select_key_video(My_Hbase, id, st, et); //找到所有密集内容视频
  for each k ∈ K
  {
    for each g ∈ G
    {
      if (k.t < g.t) //比较两段视频的起始时间
      {
        k ← unzip(k); //按照 4.3.3 解压缩稀疏空白视频
        v ← v + k; //将视频段 k 追加到 v 中
        break;
      }
    }
  }
  else
```



```

        v←v+g;           //将视频段 g 追加到 v 中
    }
}
return v;
}

```

5.2. 按图像底层特征检索播放视频

为满足基于图像底层特征的视频检索，我们设计了检索算法，下面以视频的颜色特征为例进行阐述，其它特征的检索算法与之类似。

算法 3: 基于颜色特征的视频检索算法 Video Retrieval Based On Color Features, 简称 VRCF

输入: 视频的颜色特征 a , 例如 $a='灰'$, 视频流 id , 视频数据库 My_Hbase

输出: 符合特征的视频文件集合 $VideoSet$

VRCF()

```

{
    R←∅; G←∅; VideoSet←∅;           //初始化
    use(My_Hbase);                   //打开数据库
    R←Query_Color_Index(hbase, a, id); //查询颜色特征索引
    for each r∈R
    {
        g←Find_video(hbase, r);      //找到 r 对应的视频记录
        G←G+{g};
    }
    v←∅; //初始化视频文件
    previous_t←0;
    for each g ∈ G
    {
        if (previous_t=0 | isNext(previous_t, g.t) //判断两个视频记录是否连续
        {
            previous_t←g.t;           //记录上一条视频段的起始时间
            v←v+g;
        }else
        {
            VdieoSet←VideoSet+{v};
            v←∅;                       //重新组织下一个视频文件
            previous_t←g.t;
            v←v+g;
        }
    }
    return VideoSet;
}

```

5.3. 按语义特征播放视频

为了提供视频语义特征检索，本文设计了一种云计算检索、语义写回的检索策略，主要思想是：当需要按语义特征查询时，首先判断要检索的语义特征是否已经存在于视频表 se 字段中。若该字段非空，

即有某个值, 则立即返回与 se 字段相符合的视频记录, 即要求的视频段; 否则, 利用检索要求, 分析出相关的底层特征, 例如, 例如搜索语义为“海底热液”时, 其对应的视频颜色特征更偏向于黑色、灰色等, 这个过程可以用人工或相关的关系库实现, 利用这些底层特征缩小要检索的视频记录范围; 然后利用云计算平台, 对所有视频逐个进行语义特征分析和识别; 最后将符合要求的视频记录返回, 同时将该语义特征写回该视频对应的 se 字段中, 不断完善视频语义特征库。值得说明一下, 上述方法和实现, 不是本文讨论的重点, 将在后续论文详细阐述。

6. 实验评价

6.1. 实验环境搭建及数据准备

本实验环境采用 Dell 机群, 运行的操作系统为 Linux, 其中一台计算节点安装了 Ffmpeg 及 Opencv 视频处理软件, 其它节点安装了 Hbase-2.0 数据库, 总存储容量约 2 TB。计算节点间通过高速网络互联。

实验的视频数据来自加拿大维多利亚海底实验网络的深海摄像系统[13], 我们选取了 20 段时长约 60 分钟的海底监测视频。将视频数据以 15 分钟为单位送入视频处理节点, 经分段、压缩和特征提取之后, 发送到 Hbase 数据库服务器, 由 Hbase 负责全部视频数据的存储和管理。

6.2. 视频播放实验

为了验证视频经过分段和压缩存储之后, 系统能够正常的将 Hbase 中的视频记录查找出来, 并且恢复为原始的视频流, 我们按照算法 2 进行视频自然回放实验, 播放时的的帧序列截图如下图 3。



Figure 3. Video normal playback experiment
图 3. 视频自然回放实验

从上述图像序列可以看出, 恢复后的视频基本可以流畅播放, 仅在分段点处有些许失帧, 由此可以认为本文设计的 Hbase 存储系统满足一般的视频回放需求。

6.3. 视频数据存储容量分析

为分析利用本文 Hbase 方式和传统文件方式存储视频占用的实际磁盘存储空间大小, 分别将时长为 1, 2, 3, 4, 5, 6, 7, 8, 9 小时的监测视频存储到 Hbase 数据库, 采用 `hdfs dfs -du -h/usr/hbase/hbase_db/data/default/` 命令查询 Hbase 数据表的磁盘占用量(本文 Hbase 运行与 Hdfs 之上)。另一方面, 在 linux 文件系统中可以利用 `ls -l` 命令, 查询原始视频文件的磁盘占用量, 记录两者数值大小, 在使用不同视频多次实验求均值后, 得到实验结果如图 4 所示。

从图 4 中可以看出, 在存储相同时长的视频要求下, 本文 Hbase 方式比传统文件方式占用更少的存储资源, 且随之视频的持续增长, 两者差距越来越大。视频时长为 9 小时, 节省了近 75.6% 的存储空间。由此可见, 本文针对海底监测视频设计的存储策略比文件方式节省更多的存储资源。

6.4. 视频检索实验

为了测试特征检索能力, 在开始实验前, 先利用 4.2.2 节提出的特征提取方法, 对密集内容视频进行特征提取, 然后将提取好的特征存放到 Hbase 视频表对应视频记录的属性列中。另一方面, 将原始视频

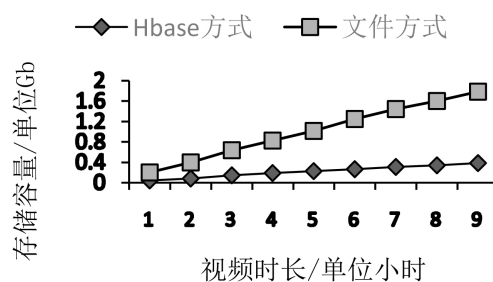


Figure 4. Analysis of video storage capacity
图 4. 视频存储容量分析

文件保存在 linux 文件系统中, 利用文件名来标定视频的摄像源 id 和拍摄起止时间等信息。下面是对上述两种视频存储方式进行检索实验的过程描述: 1) 将检索条件设定为要查询的视频的摄像源 id、起止时间和颜色特征 a。2) 对本文 Hbase 方式, 首先连接 Hbase 数据库, 调用 Hql 语句执行上述检索, Hbase shell 会返回查询结果并提供本次查询耗时。3) 对于文件方式, 调用编写 java 程序搜索视频文件目录树, 利用摄像源 id、起止时间信息来定位到符合要求的视频文件, 然后将视频输入到借助文献[14]方法实现的视频特征检索软件, 经过软件提取得到符合要求视频的起止时间信息, 记录程序执行总时间作为文件方式的检索时间。4) 将视频的时间范围依次设定为 1, 10, 20, 30, 40, 50, 60, 70, 80, 90 分钟, 分别记录两种方式对应的检索时间。通过变换检索特征, 重复进行上述实验, 对检索时间求平均值, 实验结果如图 5 所示。

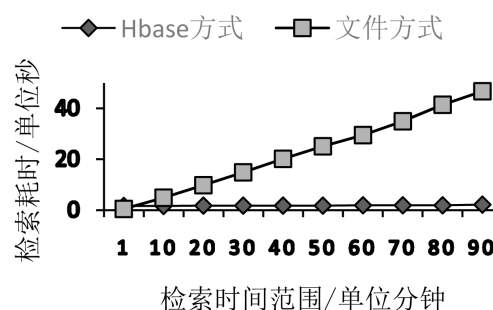


Figure 5. Video retrieval experiment
图 5. 视频检索实验

从上图 5 可以看出, 传统文件方式的检索时间随着检索范围的扩大, 呈现高斜率线性增长。而 Hbase 方式的检索时间变化幅度很小, 仅随着检索范围的增大稍有上升。当检索小视频时, 如 1 分钟的视频, 采用软件检测视频文件的方式要比 Hbase 方式更快一些, 这是因为连接 Hbase 数据库耗时比较多, 从 Hbase 主节点向从节点发送查询指令也有一定延迟。但是, 在检索 10 分钟及以上的视频时, Hbase 方式的检索速度要远高于文件方式, 这是由于 Hbase 可以直接通过查询视频特征列来完成。通过上述比较可以得出, 在应对较长视频的特征检索时, 本文 Hbase 方式的检索速度远高于传统文件检索方式。

7. 结束语

海底监控就像一双“窥探”海底的眼睛, 透过它人们可以发现海底未知的秘密, 为了存储海量视频数据, 本文设计了一种基于 Hbase 的监测视频存储方法, 该方法针对海底监测视频的稀疏性, 设计了基于运动矢量的视频分段方法, 着重保留分段后的密集内容部分, 对空白稀疏部分进行了压缩, 大大减少了视频数据量。为了能更好的存储这些数据, 设计了基于 Hbase 的视频存放策略, 该方法打破了传统数

据库加文件的视频存储模式限制,将视频稀疏的特性融入到存储设计思想中去,简化了视频的存储结构。在视频数据使用方面,设计了两种视频播放方式并提供了多种基于条件和特征的视频检索机制,相比传统的视频检索方式,速度可以大大提高。但是,在特征检索方面,特别是基于语义特征的视频检索,还不完善,下一步将充分利用深度学习和模式识别技术增强视频大数据的检索能力。

基金项目

国家社科基金项目(17BQT086);国家海底科学观测系统子项目(2970000001/001/016);CCF 信息系统开放课题项目(CCFIS2018-01-03);实验教改项目(1380104112)。

参考文献

- [1] 李建如,许惠平.加拿大“海王星”海底观测网[J].地球科学进展,2011,26(6):656-661.
- [2] 李小三,杨灿军,陈燕虎,卢汉良,李德骏,金波.海底观测网络摄像系统设计与实现[J].船舶工程,2011,33(1):65-69.
- [3] 郑红霞,张训华,赵铁虎.海底监测技术之海底观测网络[J].海洋地质前沿,2015,31(5):51-56.
- [4] 刘高辉.基于Hadoop的海量视频数据分布式处理研究[D]:[硕士学位论文].武汉:华中师范大学,2017.
- [5] 江冕,牛中盈,张淑萍.视频监控存储系统的设计与实现[J].计算机工程与设计,2014,35(12):4195-4201.
- [6] 操顺德.面向海量高清视频的高性能分布式存储系统[D]:[硕士学位论文].武汉:华中科技大学,2017.
- [7] 刘晓静.基于HBase的海量小视频存储与检索系统的研究与实现[D]:[硕士学位论文].西安:西安电子科技大学,2014.
- [8] Chang, F., Dean, J., Ghemawat, S., et al. (2008) Bigtable: A Distributed Storage System for Structured Data. *ACM Transactions on Computer Systems*, 26, 1-26. <https://doi.org/10.1145/1365815.1365816>
- [9] 张志付.基于背景减法的运动检测算法研究[D]:[硕士学位论文].上海:上海交通大学,2008.
- [10] Ding, S., Li, G., Li, Y., et al. (2017) SurvSurf: Human Retrieval on Large Surveillance Video Data. *Multimedia Tools and Applications*, 76, 6521-6549. <https://doi.org/10.1007/s11042-016-3307-4>
- [11] 杨慧炯.面向自然图像的自适应主色提取方法研究[J].微电子学与计算机,2018,35(9):99-102.
- [12] 余成伟,陆建华,郑君里.基于场景切换的H.264码率控制技术[J].清华大学学报(自然科学版),2007,47(7):1133-1136.
- [13] Taylor, S.M. (2008) Strategic and Business Planning for the VENUS and NEPTUNE Canada Ocean Observing Systems. *OCEANS*, Quebec, 15-18 September 2008, 1-7. <https://doi.org/10.1109/OCEANS.2008.5151873>
- [14] 李伟,王蓉.基于颜色和SIFT特征的视频目标检索方法研究[J].科学技术与工程,2013,13(29):8810-8814.

知网检索的两种方式:

1. 打开知网首页: <http://cnki.net/>, 点击页面中“外文资源总库 CNKI SCHOLAR”, 跳转至: <http://scholar.cnki.net/new>, 搜索框内直接输入文章标题, 即可查询;
或点击“高级检索”, 下拉列表框选择: [ISSN], 输入期刊 ISSN: 2161-8801, 即可查询。
2. 通过知网首页 <http://cnki.net/> 顶部“旧版入口”进入知网旧版: <http://www.cnki.net/old/>, 左侧选择“国际文献总库”进入, 搜索框直接输入文章标题, 即可查询。

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: csa@hanspub.org