Video Hyperlinking: Libraries and Tools for Threading and Visualizing Large Video Collection

Lei Pang, Wei Zhang, Hung-Khoon Tan, Chong-Wah Ngo

SUBMITTED TO ACM MULTIMEDIA 2012 OPEN SOURCE SOFTWARE COMPETITION

Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong
Faculty of Information & Communication Technology, University Tunku Abdul Rahman, Perak, Malaysia
{leipang3, wzhang34}@student.cityu.edu.hk, thkhoon@utar.edu.my, cscwngo@cs.cityu.edu.hk

ABSTRACT

While HTML documents could be effortlessly hyperlinked by markup tags, creation of the hyperlinks for multimedia objects is by no means easy due to the involvement of various visual processing units and intensive computational overhead. This paper introduces an open source, named VIREO-VH, which provides end-to-end support for creating hyperlinks to thread and visualize collections of videos. The software components include video pre-processing, bag-of-words based inverted file indexing for scalable near-duplicate keyframe search, localization of partial near-duplicate segments, and galaxy visualization of video collection. The open source has been internally used by VIREO research team since 2007, and was evolved over years based on experiences through developing various multimedia applications.

Categories and Subject Descriptors

H.5.4 [INFORMATION INTERFACES AND PRESENTATION]: Hypertext/Hypermedia—Architectures, Navigation; D.2.8 [DOCUMENT AND TEXT PROCESSING]: Document Preparation—Index generation

General Terms

Design, Documentation

Keywords

Video hyperlinking, partial near-duplicates, large-scale video browsing

1. INTRODUCTION

The growing proliferation of social media has accelerated the spread of professional and user generated videos on the Internet. Iconic clips of hot topics are often edited and then inserted into new videos, serving either as a reminder of topics when new events arrive, or as a support of viewpoints, or changes of perspectives with additional information being added to the original clips. Figure 1 shows some examples of iconic clips for several hot news events. In the literature, iconic clips have been found to be particularly useful for exploring different applications, including the threading of evolving news events [13], novelty reranking [11], multimedia-based question-answering [6] and social video monitoring [12]. In general, the key to these applications is to track different versions of near-duplicate videos, either fully or partially. We refer to this task as “video hyperlinking”, where the aim is to bridge videos such that the relationship among them could be easily uncovered, read and visualized.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'12, October 29–November 2, 2012, Nara, Japan.
Copyright 2012 ACM 978-1-4503-1089-5/12/10 ...$15.00.

2. SOFTWARE COMPONENTS

We modularize VIREO-VH into four basic components: video pre-processing, near-duplicate keyframe retrieval, par-

Figure 1: Examples of iconic clips.
Figure 2: High-level overview of VIREO-VH software architecture. The retrieval of near-duplicate (ND) keyframes is accomplished by indexing with bag-of-words inverted file and post-processing with geometric verification [2, 14]. The localization of partial ND segments is based on network flow optimization algorithm [10]. The display of hyperlinked videos on a 2D visualization window is implemented by force-directed algorithm [5].

Partial near-duplicate localization and galaxy-based visualization, as depicted in Figure 2. Basically, given a collection of videos, the visual content will be indexed based on bag-of-words (BoW) representation [9]. Near-duplicate keyframes will be retrieved and then temporally aligned in a pairwise manner among videos. Segments of a video which are near-duplicate to other videos in the collection will then be hyperlinked with the start and end times of segments being explicitly logged. The end product is a galaxy browser, where the videos are visualized as a galaxy of clusters on Web browser, with each cluster being a group of videos that are hyperlinked directly or indirectly through transitivity propagation. User friendly interaction is provided such that end user can zoom in and out to have a glance as well as close inspection of video relationship.

2.1 Video Pre-processing

The color histogram based algorithm in [1], which is known as one of the most reliable algorithm for cut detection, is adopted for the decomposition of videos into shots. The basic idea is that the color histogram does not change rapidly within but across shots. In VH, the frames are first extracted from a video with ffmpeg2, and then the cosine similarities between the color histograms of two adjacent frames are calculated. A shot boundary is detected if the cosine similarity exceeds a threshold $H$. In addition, for reducing excessive detection of shots due to gradual transition, lighting change and noise, the length of a shot is restricted to less than $L$ frames. The middle frame of each shot is extracted as keyframe. Further from each keyframe, local points based on Difference of Gaussian (DoG) are detected and then described with SIFT3.

2.2 Near-duplicate Keyframe Retrieval

This component is built upon the BoW representation, consisting of two major parts: offline processing and online retrieval. Figure 3(a) shows the flow of different building blocks in the component. Offline processing includes the training of visual vocabulary and Hamming medians, together with the construction of inverted file for indexing, which are briefly described as following:

2http://ffmpeg.org/
3http://www.cs.ubc.ca/~lowe/keypoints/
query frame in $Q$ old for suppressing the frames from $R$ND segment. The default values for these parameters are successively aligned frames. The parameter $\min$ specifies the maximum difference between the time stamp values of two parameters $\min$ and the set of ND keyframes from a reference video. Temporal constraints specified by three libraries: ANN\(^4\), LP\_SOLVE\(^5\), and MCF\(^6\).

By progressively aligning the videos in a pairwise manner, videos with partial NDs are transitively threaded through hyperlinking. The set of partial NDs, common to a video thread, could then be viewed as iconic clips.

### 2.4 Galaxy-based Visualization

To provide a glance of how videos are hyperlinked among each other, this component provides tools for visualizing the video relationship as a galaxy of video snippets. A snippet is defined as a cluster of videos that are threaded by partial NDs. Specifically, these videos are reachable from any video in the cluster by traversing the hyperlinks, and hence are likely to be somewhat related. Figure 4(a) shows a galaxy view of video collection, where the constellation of visual snippets clearly indicates the video relationship as a result of hyperlinking. The structure of visual snippets could carry semantic meanings. For example, a fully connected snippet could indicate high redundancies among videos; a sparsely connected structure may expose the happening of an evolving event\(^7\) or manipulation history of media\(^8\), whereas a highly centralized video (hub) with excessive hyperlinks to other videos may be a summary video\(^9\).

The galaxy-based visualization is programed by JavaScript InfoVis Toolkit\(^7\). Force directed algorithm\(^5\) is adopted to determine the layout of visual snippets in the galaxy view. To provide easy navigation of the galaxy, the implementation allows user-friendly interaction by zooming in and out of galaxy and visual snippets. By clicking a snippet, an interface as in Figure 4(b) will be shown. In this view, each video in a snippet is presented as a series of keyframe thumbnails. Partial NDs are highlighted with different colors so that the hyperlinks among them can be easily read.

### 3. PERFORMANCE

We first briefly present the performance of two key components: ND keyframe retrieval and partial ND localization. The overall performance of the open source in terms of efficiency is then followed.

#### 3.1 ND Keyframe Retrieval

For comparison, we use a public dataset “Holiday”\(^8\) to evaluate the performance of ND keyframe retrieval. The comparison is made against the state-of-the-arts results reported in\([2]\). Using the same configuration (e.g., same vocabulary size, Hamming embedding, etc.) as\([2]\), Table 1 shows the retrieval performance based on 500 queries. Generally, our result is slightly better than that reported in\([2]\).

#### 3.2 Partial ND localization

We show the performance on a collection of 220 videos (totally 31.2 hours) crawled from YouTube using the key-

---

\(^4\)http://www.cs.umd.edu/~mount/ANN/

\(^5\)http://lp.solve.sourceforge.net/

\(^6\)http://typo.zib.de/opt-long_projects/Software/Mcf/

\(^7\)http://thejit.org/

\(^8\)http://lear.inrialpes.fr/~jegou/data.php
Table 1: Comparison of ND retrieval in terms of mean average precision between our implementation and [2].

<table>
<thead>
<tr>
<th>version</th>
<th>BoW</th>
<th>HE + MA</th>
<th>HE + MA + WGC</th>
</tr>
</thead>
<tbody>
<tr>
<td>[2]</td>
<td>0.466</td>
<td>0.735</td>
<td>0.815</td>
</tr>
<tr>
<td>ours</td>
<td>0.480</td>
<td>0.744</td>
<td>0.818</td>
</tr>
</tbody>
</table>

Table 2: Runtime of each step

<table>
<thead>
<tr>
<th>Stage</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-processing</td>
<td></td>
</tr>
<tr>
<td>Keyframe Extraction</td>
<td>54min</td>
</tr>
<tr>
<td>Feature Extraction</td>
<td>21min</td>
</tr>
<tr>
<td>ND Retrieval</td>
<td></td>
</tr>
<tr>
<td>Vocabulary Training</td>
<td>12min</td>
</tr>
<tr>
<td>Hamming Training</td>
<td>6min</td>
</tr>
<tr>
<td>Indexing</td>
<td>4min</td>
</tr>
<tr>
<td>Online Retrieval</td>
<td>7min</td>
</tr>
<tr>
<td>Partial ND Localization</td>
<td>8min</td>
</tr>
<tr>
<td>Galaxy Visualization</td>
<td>55sec</td>
</tr>
</tbody>
</table>

Word “economic collapse”. Using our open source and default parameter settings, a total of 35 partial ND segments are located, resulting in 10 visual snippets. Compared to the ground-truth manually created on this dataset, the precision of ND localization is as high as 0.95 and the recall is 0.66. The generated visual snippets are also meaningful, aligning well to events such as the bankruptcy of Lehman Brothers and European debt crisis.

3.3 Efficiency

Using the collection of 220 videos as examples, Table 2 lists the running time for each step. The experiment was conducted on a standard PC with dual core 3.16 GHz CPU and 3 GB of RAM. In total, creating a galaxy view for 31.2 hours of videos (more than 4,000 keyframes) could be completed within 2.5 hours using our open source.

4. SOFTWARE USABILITY AND TARGET AUDIENCE

VIREO-VH could be either used as an end-to-end system with video collection as input and visual hyperlinks as output, or called as functions independently for development of different applications. For content owners interested in the content-wise analysis of a video collection, VIREO-VH can be used as an end-to-end system by simply inputting the location of video collection and the output paths. The resulting output can then be viewed with the provided interactive interface for showing the glimpse of video relationship in the collection.

VIREO-VH also provides libraries to grant researchers for programmatic access. The libraries consist of various classes (e.g., Vocab, HE, Index, SearchEngine and CNetwork), providing different functions for vocabulary/hamming training, keyframe indexing, near-duplicate keyframe searching and video alignment. Users can refer to the manual or website9 for details. Furthermore, the components of VIREO-VH are independently developed for providing flexibility that users can substitute any of the components with their own versions of implementation. This capability is particular useful for benchmarking the users’ own choice of algorithms. As an example, users can choose their own visual vocabulary and Hamming median but use the open source for building index and retrieving near-duplicate keyframes, by ticking appropriate check boxes as shown in Figure 3(b).

5. ACKNOWLEDGEMENTS

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 119610).

6. REFERENCES
