# 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, Navigaiton*; 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.

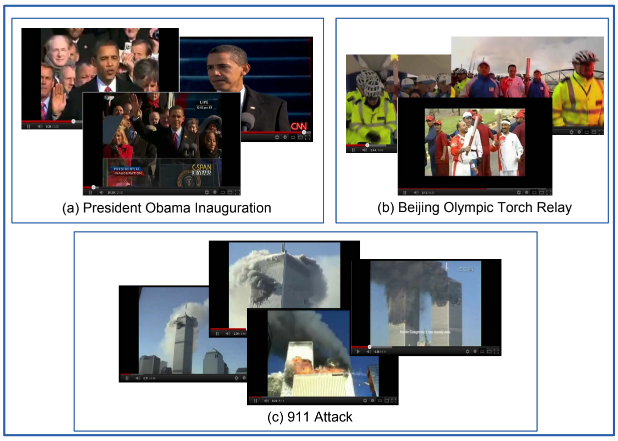

(a) President Obama Inauguration
(b) Beijing Olympic Torch Relay
(c) 911 Attack

**Figure 1: Examples of iconic clips.**

*While video hyperlinking (VH) is becoming a fundamental task for many multimedia applications, to the best of our knowledge, there is no open source available yet for automatic creation and visualization of hyperlinks among large collection of videos. In this paper, we introduce an open source, named VIREO-VH*[1]*, which provides libraries and tools for tracking of video near duplicates, creation of hyperlinks, and visualization of video linkages.* The functionalities are mostly implemented based on the state-of-the-art technologies.

## 2. SOFTWARE COMPONENTS

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

[1] Available at: http://vireo.cs.cityu.edu.hk/VIREO-VH/

*MM'12,* October 29–November 2, 2012, Nara, Japan.

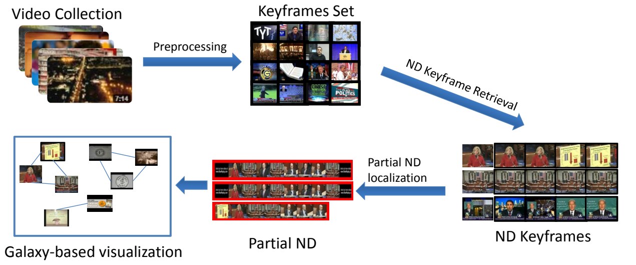


**Figure 2: High-level overview of VIREO-VH software architecture. The retrieval of near-duplicate (ND) keyframes is accomplished by indexing with bag-of-words inverted file and post-processing with geometric verification [2, 14]. The localization of partial ND segments is based on network flow optimization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

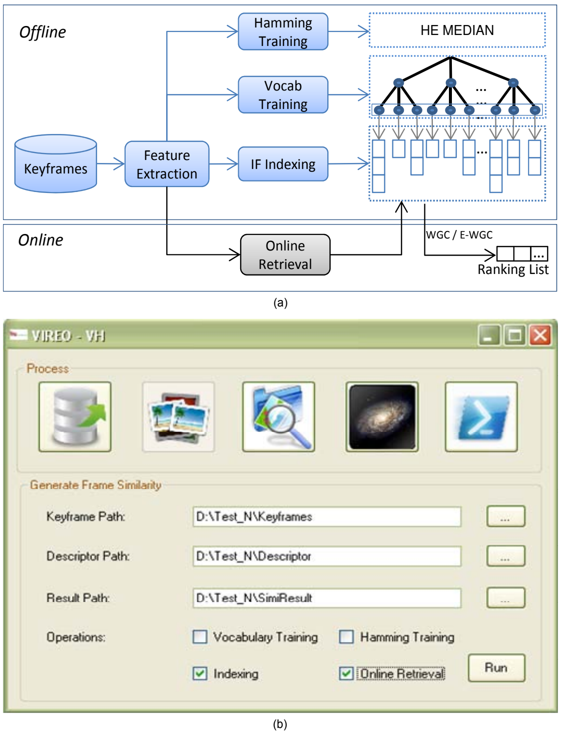
tial near-duplicate localization and galaxy-based visualization, as depicted in Figure 2. Basically, given a collection of videos, the visual content will be indexed based on bag-of-words (BoW) representation [9]. Near-duplicate keyframes will be retrieved and then temporally aligned in a pairwise manner 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 ffmpeg[2], 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 SIFT[3].

## 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:



Keyframe Path: D:\Test_N\Keyframes

Descriptor Path: D:\Test_N\Descriptor

Result Path: D:\Test_N\SimiResult

Operations: □ Vocabulary Training □ Hamming Training

☑ Indexing ☑ Online Retrieval

**Figure 3: Near duplicate keyframe retrieval: (a) flow chart of different building blocks; (b) screenshot showing how different blocks could be conveniently and independently used by ticking the check boxes.**

*Vocabulary Training* generates a hierarchical vocabulary of 20k leaf nodes by default. The training samples (SIFT descriptors) are randomly drawn from the video collection. Standard k-means algorithm is implemented for top-down hierarchical clustering of samples. With the vocabulary tree, each word is a Voronoi cell where SIFT descriptors can be efficiently quantized.

*Hamming Training* generates a tiny signature for each visual word to alleviate the quantization error [2]. The training is performed by partitioning a Voronoi cell into $2^{32}$ sub-regions, resulting in a 32-bits binary signature attached to each descriptor quantized to the cell. Note that Hamming distance weighting [2] is also implemented for better scoring of similarity.

*Inverted file structure* is constructed by mapping the descriptors of every keyframe to words. Basically, each word in the structure points to a posting list that keeps all the features quantized to that word. To facilitate fast pruning of words and geometric verification during online retrieval, each feature in the posting list also keeps the following information: spatial location, local geometric information (dominant scale and orientation) and Hamming signature.

*Online retrieval* consists of the steps for processing every keyframe in a collection as a query, searching of near-duplicate (ND) keyframes, and post-processing by fast geometric verification. To alleviate the adverse effect due to quantization, multiple assignments (MA) of nearest visual words to local features of query keyframes based on soft weighting [8] is implemented. The candidate ND keyframes of a query are retrieved by traversing the inverted file with the aid of Hamming signature for pruning false matches of words. The candidate keyframes are further verified by

[2] http://ffmpeg.org/

[3] http://www.cs.ubc.ca/~lowe/keypoints/

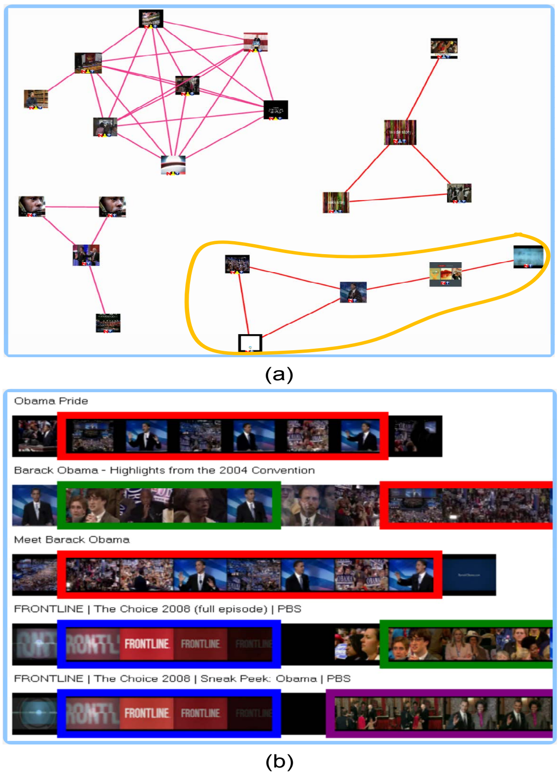

(a)
Obama Pride
Barack Obama - Highlights from the 2004 Convention
Meet Barack Obama
FRONTLINE | The Choice 2008 (full episode) | PBS
FRONTLINE | The Choice 2008 | Sneak Peek: Obama | PBS
(b)

**Figure 4: Galaxy based visualization: (a) visual snippets of various structures, (b) a close view of the snippet circled in yellow shows how videos are hyperlinked. Boxes of different colors are used to highlight different links.**

checking the geometric consistency of matched words. We provide two versions of checking based on WGC (weak geometric consistency) [2] and E-WGC (enhanced WGC) [14].

Considering that a collection could contain hundreds or even thousands of videos, ending up to tens of thousands of keyframes as queries to be processed. To reduce processing time, multi-threading (pthread) using C++ is implemented for this component. The speed of processing will benefit greatly if running on a multi-core processor.

### 2.3 Partial Near-duplicate Localization

This component consolidates the set of retrieved ND keyframes by temporally aligning them at the video level and hence creating hyperlinks. More specifically, partial ND segments of videos are located and explicitly linked. The implementation is based on [10] which converts the localization problem into a network flow optimization problem that efficiently resolves the alignment by joint utilization of visual cues and temporal constraints. Given an anchor video $Q$ and the set of ND keyframes from a reference video $R$, a temporal network is constructed by chronologically linking the ND keyframes. Temporal constraints specified by three parameters $wnd$, $S_{min}$, $L_{min}$ are imposed on the network for finding a maximal path corresponding the best possible partial ND alignment. The parameter $wnd$ specifies the maximum difference between the time stamp values of two successively aligned frames. The parameter $S_{min}$ is a threshold for suppressing the frames from $R$ with low similarity to query frame in $Q$. $L_{min}$ states the minimum length of a ND segment. The default values for these parameters are set based on empirical experience, while users are allowed to adjust the values to fit their applications. For fast optimization, early partitioning of the temporal network into multiple sub-networks for parallel alignments is also implemented. The implementation of this component relies on 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 [4], whereas a highly centralized video (hub) with excessive hyperlinks to other videos may be a summary video [3].

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://lpsolve.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].**

| version | BoW | HE + MA | HE + MA + WGC |
|---|---|---|---|
| [2] | 0.469 | 0.735 | 0.813 |
| ours | 0.480 | 0.744 | 0.818 |

**Table 2: Runtime of each step**

| *Stage* | | *Time* |
|---|---|---|
| Pre-processing | Keyframe Extraction | 54min |
| | Feature Extraction | 21min |
| ND Retrieval | Vocabulary Training | 42min |
| | Hamming Training | 6min |
| | Indexing | 4min |
| | Online Retrieval | 7min |
| Partial ND Localization | | 8min |
| Galaxy Visualization | | 55sec |

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 website[9] 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

[1] J. S. Boreczky and L. A. Rowe. Comparison of video shot boundary detection techniques. In *Storage and retrieval for skill image and video databases IV*, pages 170–179, 1996.

[2] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *IJCV*, 87:192–212, 2010.

[3] J. R. Kender, M. L. Hill, A. P. Natsev, J. R. Smith, and L. Xie. Video genetics: a case study from youtube. In *ACM Multimedia*, pages 1253–1258, 2010.

[4] L. Kennedy and S.-F. Chang. Internet image archaeology: automatically tracing the manipulation history of photographs on the web. In *ACM Multimedia*, pages 349–358, 2008.

[5] S. G. Kobourov. *Handbook of Graph Drawing and visualization.* CRC Press, to appear in 2012.

[6] G. Li, Z.-Y. Ming, R. Hong, T.-S. Chua, H. Li, and S. TangChua. Question answering over community-contributed web videos. *IEEE Multimedia*, 17:46–57, 2010.

[7] L. Pang, S. Tan, H. K. Tan, and C. W. Ngo. Galaxy browser: exploratory search of web videos. In *ACM Multimedia*, pages 803–804, 2011.

[8] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

[9] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, volume 2, pages 1470–1477, 2003.

[10] H.-K. Tan, C.-W. Ngo, R. Hong, and T.-S. Chua. Scalable detection of partial near-duplicate videos by visual-temporal consistency. In *ACM Multimedia*, pages 145–154, 2009.

[11] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Practical elimination of near-duplicates from web video search. In *ACM Multimedia*, pages 218–227, 2007.

[12] L. Xie, A. Natsev, J. R. Kender, M. Hill, and J. R. Smith. Visual memes in social media: tracking real-world news in youtube videos. In *ACM Multimedia*, pages 53–62, 2011.

[13] D.-Q. Zhang and S.-F. Chang. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In *ACM Multimedia*, pages 877–884, 2004.

[14] W. Zhao, X. Wu, and C. W. Ngo. On the annotation of web videos by efficient near-duplicate search. *IEEE Trans. on Multimedia*, 12(5):448–461, 2010.

[9] http://vireo.cs.cityu.edu.hk/VIREO-VH/Doc.html