Near-Lossless Video Summarization*

Lin-Xie Tang [†‡], Tao Mei [‡], Xian-Sheng Hua [‡]
[†] University of Science and Technology of China, Hefei 230027, P. R. China
[‡] Microsoft Research Asia, Beijing 100190, P. R. China
tlxxp@mail.ustc.edu.cn; {tmei,xshua}@microsoft.com

ABSTRACT

The daunting yet increasing volume of videos on the Internet brings the challenges of storage and indexing to existing online video services. Current techniques like video compression and summarization are still struggling to achieve the two often conflicting goals of low storage and high visual and semantic fidelity. In this work, we develop a new system for video summarization, called "Near-Lossless Video Summarization" (NLVS), which is able to summarize a video stream with the least information loss by using an extremely small piece of metadata. The summary consists of a set of synthesized mosaics and representative keyframes, a compressed audio stream, as well as the metadata about video structure and motion. Although at a very low compression ratio (i.e., 1/30 of H.264 baseline in average, where traditional compression techniques like H.264 fail to preserve the fidelity), the summary still can be used to reconstruct the original video (with the same duration) nearly without semantic information loss. We show that NLVS is a powerful tool for significantly reducing video storage through both objective and subjective comparisons with state-of-the-art video compression and summarization techniques.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*video*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based services*

General Terms

Algorithms, Experimentation, Human Factors.

Keywords

Video summarization, video storage, online video service.

* This work was performed when Lin-Xie Tang was visiting Microsoft Research Asia as a research intern.

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'09, October 19–24, 2009, Beijing, China.
Copyright 2009 ACM 978-1-60558-608-3/09/10 ...$10.00.

1. INTRODUCTION

Due to the proliferation of digital capture devices, as well as increasing video sites and community sharing behaviors, a tremendous amount of video streams are being collected from various sources and stored for a variety of purposes, ranging from surveillance, monitoring, broadcasting, to entertainment. According to the report in [29], the most popular video sharing site, YouTube [11], is now ingesting 15 hours of new videos each minute, indicating that the storage consumption increases at 5 TB every day if those videos are compressed with a bit rate of 500 kbps—a typical setting of the most popular codec on the Web, i.e., H.264 [5]. Those daunting yet increasing volumes of videos have brought sort of challenges: 1) *limited server storage*, it would be difficult for a single video site to host all the uploaded videos; 2) *considerable streaming latency*, streaming such large amount of videos over the limited bandwidth capabilities would lead to considerable latency; 3) *unsatisfying video quality*, the blocking artifacts and visual distortions are usually expected by traditional compression techniques, especially when extremely low bit rate is applied, which in turn degrades user experience of video browsing. Therefore, an effective video summarization technique which can represent or compress a video stream via extremely low storage is highly desirable. To deal with above problems, an effective video summary system should have the following capabilities: 1) low storage consumption which can benefit the backend of a typical video host by significantly reducing video storage and effective indexing, and 2) ability to be used for decoding so that the original video can be reconstructed or presented in the frontend without any semantic loss.

In general, there exist two solutions to the problem mentioned above, i.e., very-low bit rate video compression and video summarization. The first solution is designed from the perspective of signal processing, aiming to eliminate the spatio-temporal signal redundancy. For example, H.263 [6] and H.264 [5] are the most popular video codecs on the Internet. The second is derived from the perspective of computer vision, aiming to select a subset of the most informative frames or segments from the original video and represent the video in a static (i.e., a collection of keyframes or synthesized images) or dynamic (i.e., a new generated video sequence) form.

Video compression techniques are widely used in a majority of online video sites like YouTube [32], Hulu [16], Metacafe [27], Revver [28], and so on. For example, YouTube employed H.263 (at the bit rate of 200 $\sim$ 900 kbps) as the codec for "standard quality" video, and H.264 (at the bit rate

(a) baseline (590 kpbs) (b) 1/20 (29.5 kpbs) (c) 1/30 (19.7 kpbs) (d) NLVS (20.0 kpbs)

Figure 1: Keyframe examples compressed by different bit rate profiles in H.264 and reconstructed by the proposed NLVS. The visual fidelity in (d) is much better than (c) in terms of blocking artifacts and visual distortions with the similar bit rates.

of 2,000 kbps) for "High-Definition Quality" video [32]. However, compared with previous compression standards, these compression techniques only achieve the limited reduction of the original signal (e.g., H.264 usually achieves half or less the bit rate of MPEG-2 [3] [31]). Moreover, when given an extremely low bit rate (e.g., less than 1/20 of H.264 baseline profile), they usually introduce severe blocking artifacts and visual distortions which in turn degrade user experience. Figure 1 shows several keyframes with poor visual quality from a video stream compressed by H.264 reference software at a series of decreasing bit rate settings (H.264 baseline profile, and 1/20, 1/30 of baseline profile) [18]. It is observed that when the compression ratio decreases, both the amount and area of the blocking artifacts increase. When the bit rate reaches 1/30 of baseline profile, the visual quality from H.264 is degraded significantly.

Video summarization is an alternative approach to handling the storage of large-scale video data. Conventional video summarization mainly focuses on selecting a subset of video frames or segments that represent the important information of the original video. These frames or segments are then connected according to certain spatio-temporal orders to form a compact visual summary. Such summarization techniques can be used by any video search system to index the videos in the backend and present the search results in the front. For example, Microsoft's video search engine presents a short dynamic thumbnail for each searched video for fast preview—it returns a 15 or 30 seconds dynamic summary for each video [8]. However, as content understanding is still in its infancy, the video summary cannot guarantee the preservation of all informative content. In other word, such summary is a kind of lossy representation.

In this paper, we propose a new system to tackle with video storage problem. Our rescue comes from the use of mature techniques in computer vision and video processing. The proposed system, called "Near-Lossless Video Summarization" (NLVS), can achieve extremely low compression ratio (e.g., $1/30 \sim 1/40$ of H.264 baseline profile) without any semantic information loss. The near-lossless summary is achieved by using a set of compressed keyframes and mosaic images, as well as video structure and motion information, which can be in turn used for reconstructing a video with exactly the same duration as the original. By "near-lossless," we refer to: 1) we can reconstruct the video based on the summary so that human can understand the content without any semantic misunderstanding, and 2) we can do a wide variety of applications purely based on the compressed summary without performance degradation. Towards near-lossless summarization, we identify several principles for designing a NLVS system: 1) a video is processed in the granularity of subshot (which is a sub segment within a shot, indicating coherent camera motion of the same scene); 2) each subshot is then classified into four classes according to camera motion; and 3) a predefined set of mechanisms is performed to extracted metadata (i.e., representative keyframes and synthesized mosaic images, as well as structure and motion information) from these subshots, resulting in an extremely low compression ratio compared to H.264. Through the summary, a video (with audio track) with the same duration as the original video can be reconstructed without any semantic loss based on the predefined rendering schema. Figure 1 illustrates some frames reconstructed by NLVS, with the bit rate at 1/30 of H.264 baseline profile. NLVS greatly differs from traditional video compression in that it achieves much lower compression ratio without visual distortions, and video summarization in that it contains all the information without any semantic information loss.

The rest of the paper is organized as follows. Section 2 reviews related work on video compression and summarization. Section 3 provides a system overview of NLVS. The details of summary generation and video reconstruction are described in Section 4 and 5, respectively. Section 6 gives the evaluations, followed by conclusions in Section 7.

2. RELATED WORK

2.1 Video Compression

Most commonly used video compression standards are provided by ITU-T or ISO/IEC. For example, the MPEG-x family standards from ISO/IEC Moving Picture Experts Group (MPEG) are widely adopted for high quality professional programs [2] [3] [4], while H.26x standards from ITU-T Video Coding Experts Group (VCEG) are predominantly applied for low bit rate videos such as conference and user-generated videos [5] [6]. H.264 (also known as MPEG-4 part 10) is the newest video compression standard jointly approved by MPEG and VCEG, which contains a rich family of "profiles," targeting at specific classes of applications [5]. H.264 has achieved a significant improvement in rate-distortion efficiency.

Although effective, the file size of the compressed signal by existing techniques is still far from practical requirements. Even the most popular video codec (i.e., H.264) can only achieve the best compression rate of $1/15 \sim 1/25$ (of its baseline) with the visually acceptable distortions. On the other hand, considering bandwidth limitation and transmission latency, researchers are now developing scalable video coding techniques which can achieve lower bit rates.

2.2 Video Summarization

Video summarization is a kind of technique that uses a subset of representative frames or segments from the original video to generate an abstraction [30]. The research on traditional video summarization has proceeded along two dimensions according to the metadata used for visualization, i.e., static summarization and dynamic skimming. Static summarization represents a video in a static image form. This

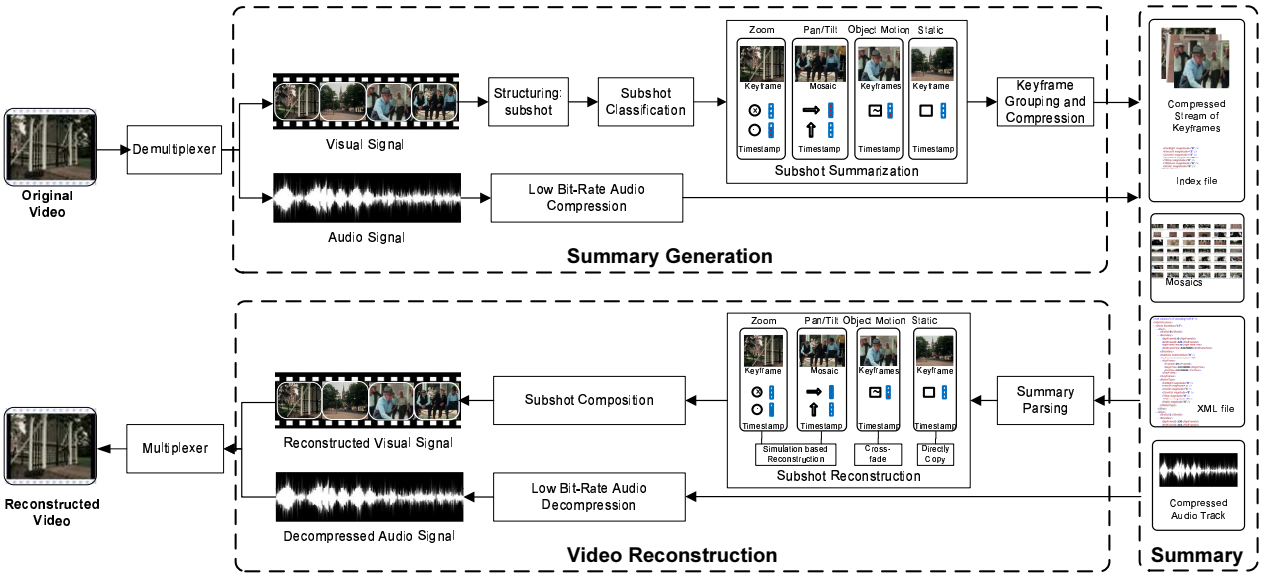


Figure 2: Framework of NLVS.

kind of summarization heavily depends on the keyframes extraction algorithm. According to different sampling mechanisms [15] [33], a set of keyframes are extracted from the shots of original video. Then, the selected keyframes are arranged or blended in a two-dimensional space. Storyboard is the most popular representation of video in the static form. For dynamic summarization, most mechanisms select video clips from the original video. Ma *et al.* proposed event-oriented scheme by using a user attention model [22]. Observing that users prefer the content with high visual quality rather than those detected as "attractive" while containing distortions, Mei *et al.* designed a video summarization system based on quality assessment [26].

Regardless static or dynamic, none of existing video summaries is able to losslessly preserve the semantic information, that is, the summaries belong to a kind of "lossy compression" of the original signal. Moreover, most of current summarization techniques highly depend on understanding video content, i.e., whether a static frame or segment corresponds to a highlight or important episode. Since semantic analysis of video content still remains a challenging problem, they cannot guarantee the preserve of informative content. As a result, it is difficult for a viewer to recall all the stories in the video while browsing. By "lossless summarization," we refer to the concept that the summary can be used for reconstructing the video with the same duration as original and without any semantic loss at the same time.

3. SYSTEM FRAMEWORK

Video is an information-intensive yet structured medium conveying time-evolving dynamics (i.e., camera and object motion). For designing an effective summarization system towards the two often conflicting goals of near-lossless "compression" and extremely low storage consumption, the following principles should be considered.

- To reduce the redundancy as much as possible, the temporal structure of video sequence such as shot and keyframe should be considered. As the contents in the entire video have large variations, a smaller segment with coherent content should be considered as the basic unit for summarization. Extracting metadata from this unit would significantly reduce the redundancy.

- To keep the information as much as possible, all the segments in the video should be included for extracting the metadata. As video differs from image in that it is a sequence of images containing dynamics, it is desirable to keep the dominant motion information (both camera and object motion) together with a set of representative images in the summary. In this way, the semantic could be preserved as much as possible.

- The summarization mechanism should be capable to maintain not only color, texture and shape in a single frame, but also the motion between successive frames.

Motivated by the above principles, we propose the framework of NLVS in Figure 2. The basic idea is to detect several representative frames and extract motion information for each basic segment. These frames and motion metadata, together with the compressed audio signal, will be used as the video summary and further for reconstructing the original video. As a *shot* is usually too long and contains diverse contents, a sub segment of shot, i.e., *subshot*, is selected as the basic unit for metadata extraction[1]. As shown in Figure 2, the framework consists of two main components, i.e., lossless video summarization and lossless video reconstruction. First, the visual and aural tracks are de-multiplexed from the original video. The video track is then decomposed to a series of subshot by a motion-based method [20], where each subshot is further classified into one of the four categories on the basis of camera motion, i.e., *zoom*, *translation*

[1] Shot is an uninterrupted temporal segment in a video, recorded by a single camera. Subshot is sub segment within a shot, say, each shot can be divided into one or more consecutive subshots. In the NLVS, subshot segmentation is equivalent to camera motion detection, indicating that one subshot corresponds to one unique camera motion within a shot [26].

Table 1: Key notations in the NLVS.

V	original video
V'	reconstructed video
N	number of subshots in a video
S_i	i-th subshot of video V
$S_i^{'}$	i-th subshot of video V'
N_i	number of frames in subshot S_i
M_i	number of keyframes in subshot S_i
$F_{i,j}$	j-th frame of subshot S_i
$F_{i,j}^{'}$	j-th frame of subshot S_i'
$KF_{i,k}$	k-th keyframe of subshot S_i
$I(KF_{i,k})$	frame index of keyframe $KF_{i,k}$
$C(F_{i,j})$	camera center of frame $F_{i,j}$ in subshot S_i
$Z^{acc}(S_i)$	accumulated zoom factor of subshot S_i
$Z(F_{i,j}^{'})$	zoom factor for rendering frame $F_{i,j}'$

(*pan/tilt*), *object* and *static*. An appropriate number of frames or synthesized mosaic images are extracted from each subshot [2]. To further reduce the storage, the selected frames are grouped according to color similarity and compressed by H.264, while the audio track is re-compressed by AMR audio codec at 6.7 kbps [7]. Finally, the summary consists of the mosaic images, the compressed frames and audio track, as well as the video structure and motion metadata (stored in a XML file [10]). Accordingly, the reconstruction module parses the summary, reconstructs each subshot on the basis of certain camera motion, concatenates all the reconstructed subshots, and finally multiplexes visual and aural track to reconstruct the original video. We will show the details of the lossless video summarization and reconstruction in Section 4 and 5, respectively. For the sake of mathematical tractability, a set of key notations is listed in Table 1.

4. SUMMARY GENERATION

This section describes the generation of video summary, targeting at an estimated 1/30 compression ratio of H.264 baseline at which traditional compression techniques fail to preserve the visual fidelity. As subshot is the basic unit for metadata extraction in the NLVS, we first describe how to decompose the original video into subshots and classify subshots into four dominant motion categories. Then, we show different summarization methods for different subshot categories. We will discuss how to further reduce the metadata storage by frame grouping and compression, as well as audio track compression via a very low bit rate audio codec.

4.1 Subshot Detection and Classification

The de-multiplexed video track is segmented into a series of shots based on a color-based algorithm [34]. Each shot is then decomposed into one or more subshots by a motion threshold-based approach [20], and each subshot is further classified into one of the six categories according to camera motion, i.e., *static*, *pan*, *tilt*, *rotation*, *zoom*, and *object motion*. The algorithm proposed by Konrad *et al.* is employed for estimating the following affine model parameters between two consecutive frames [21]

$$\begin{cases} v_x = a_0 + a_1 x + a_2 y \\ v_y = a_3 + a_4 x + a_5 y \end{cases} \quad (1)$$

where a_i $(i = 0, \ldots, 5)$ denote the motion parameters and

[2] Mosaic is a synthesized static image by stitching successive video frames in a large canvas [17].

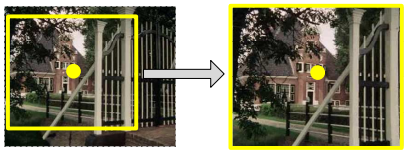


Figure 3: Examples of the distant view (left) and close-up view frame (right) in a *zoom-in* subshot.

(v_x, v_y) the flow vector at pixel (x, y). The motion parameters in equation (1) can be represented by a set of more meaningful parameters to illustrate the dominant motion in each subshot as follows [9]

$$\begin{cases} b_{pan} = a_0 \\ b_{tilt} = a_3 \\ b_{zoom} = \frac{a_1 + a_5}{2} \\ b_{rot} = \frac{a_4 - a_2}{2} \\ b_{hyp} = \left|\frac{a_1 - a_5}{2}\right| + \left|\frac{a_2 + a_4}{2}\right| \\ b_{err} = \frac{\sum_{j=1}^{N}\sum_{i=1}^{M}|p(i,j) - p'(i,j)|}{M \times N} \end{cases} \quad (2)$$

where $p(i, j)$ and $p^{'}(i, j)$ denote the pixel value of pixel (i, j) in the original and wrapped frame, respectively. M and N denote the width and height of the frame. Based on the parameters in equation (2), a qualitative thresholding method can be used to sequentially identify each of the camera motion categories in the order of *zoom*, *rotation*, *pan*, *tilt*, *object motion* and *static* [20]. In the NLVS, we treat *pan* and *tilt* in a single category of *translation*, as to be explained later, the mechanisms for extracting metadata from these two kinds of subshots are identical. As *rotation* motion seldom occurs, we take it as a special case and regard it as the *object* motion. As a result, each subshot belongs to one of the four classes, i.e., *zoom, translation* (*pan/tilt*), *object*, and *static*.

4.2 Subshot Summarization

For the sake of simplification, we define the following terms: a video V consisting of N subshots is denoted by $V = \{S_i\}_{i=1}^{N}$, and a subshot S_i can be represented by a set of successive frames $S_i = \{F_{i,j}\}_{j=1}^{N_i}$ or keyframes $S_i = \{KF_{i,k}\}_{k=1}^{M_i}$. Please see Table 1 for more details.

Zoom subshot. Depending on the tracking direction, we label each *zoom* subshot as *zoom-in* or *zoom-out* based on b_{zoom}, which indicates the magnitude and direction of *zoom*. In the *zoom-in* subshot, successive frames describe gradual change of the same scene from a distant view to a close-up view, as shown in Figure 3. Therefore, the first frame is sufficient to represent the whole content in a *zoom-in* subshot. Likewise, the procedure of *zoom-out* is reverse—the last frame is sufficient to be representative. Thus, we can design summarization scheme for *zoom* subshot from two aspects, i.e., keyframe selection and motion metadata extraction. Here, we only take *zoom-in* subshot as an example.

We choose the first frame as the keyframe in a *zoom-in* subshot. In addition, camera motion is critical for recovering the whole subshot. The camera focus (i.e., the center point in Figure 3) and the accumulated zoom factors (i.e., zooming magnitude) of all frames with respect to the keyframe are recorded into a XML file. To obtain the camera center and accumulated zoom factor, we wrap all the frames to the keyframe based on the affine parameters in equation (1). For frame $F_{i,j}$ in the *zoom-in* subshot S_i, we calculate the

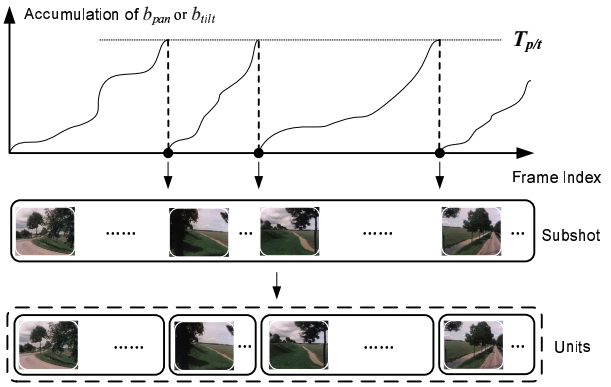


Figure 4: Segmentation of unit in the *translation* subshot.

center of the wrapped image (the center point in the left part of Figure 3) $C(F_{i,j}) = (C_x(F_{i,j}), C_y(F_{i,j}))$ by

$$C_x(F_{i,j}) = \frac{\sum_{m=1}^{H'_j}\sum_{n=1}^{W'_j} p_x(m,n)}{W'_j \times H'_j}, C_y(F_{i,j}) = \frac{\sum_{m=1}^{H'_j}\sum_{n=1}^{W'_j} p_y(m,n)}{W'_j \times H'_j} \quad (3)$$

where $p_x(m,n)$ and $p_y(m,n)$ denote the coordinate of the wrapped frame, while W'_j and H'_j denote the width and height of j-th wrapped frame. The accumulated zoom factor $Z^{acc}(S_i)$ can be computed by the area of the last frame wrapped in the global coordinates (i.e., the first keyframe)

$$Z^{acc}(S_i) = \sqrt{\frac{W'_{N_i} \times H'_{N_i}}{W \times H}} \quad (4)$$

where W'_{N_i} and H'_{N_i} denote the width and height of the last wrapped frame, while W and H denote those of the original.

Translation subshot. Compared to a *zoom* subshot, a *translation* subshot represents a scene through which camera is tracking horizontally or vertically. However, for *translation* subshot, a keyframe is far from enough to describe the whole story. For describing the wide field-of-view of the subshot in a compact form, the image mosaic is adopted in the summarization scheme. Existing algorithms for mosaic typically involve two steps [17] [23]: motion estimation and image wrapping. The first step builds the correspondence between two frames by estimating the parameters in equation (1), while the second uses the results in the first step to wrap the frames with respect to the global coordinates.

Before generating panoramas for each subshot, we first segment the subshot into units using b_{pan} and b_{tilt} to insure homogeneous motion and content in each unit. As a wide view derived from a large amount of successive frames probably results in distortions in the generated mosaic [25], each subshot is segmented into units using leaky bucket algorithm [19] [30]. As shown in Figure 4, if the accumulation of b_{pan} or b_{tilt} exceeds $T_{p/t}$, one unit is segmented from the subshot. For each unit, we generate a mosaic image to represent this unit [17]. Then, we save these mosaics and the focuses of camera (i.e., centroid of each frame in the mosaic image) obtained in equation (3) as the metadata. Figure 5 shows an example of mosaic generation in a subshot.

Object subshot. As there are usually considerable motions and appearance changes in a *object* subshot, we adopt

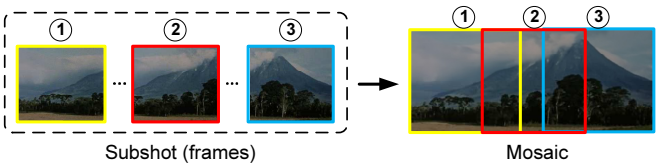


Figure 5: Mosaic generation. The three frames in a subshot are stitched together in the mosaic.

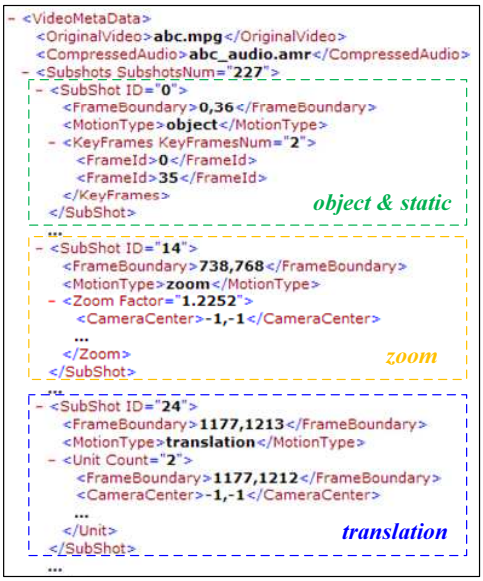


Figure 6: XML file format.

a frame sampling strategy to select the representative frames in the NLVS. As representative of content change between frames, we adopted b_{err} as the metric of object motion in *object* subshot. We also employ leaky bucket algorithm [19] [30] and threshold T_{om} for keyframe selection on the curve of accumulation of b_{err}. Moreover, we employ T_f to avoid successive selection in highly active subshot. That is, each selected keyframe $KF_{i,k}(k = 0, \ldots, M_i)$ satisfies:

$$I(KF_{i,k}) - I(KF_{i,k-1}) \geqslant T_f \quad (5)$$

where $I(KF_{i,k})$ is the frame index of $KF_{i,k}$. We can also take Figure 4 as the illustration for keyframe selection in an *object* subshot, where $T_{p/t}$ is replaced with T_{om} and b_{pan} or b_{tilt} is replaced by b_{err}. At each peak, a frame is selected as the keyframe. In addition, the first and last frames are also selected as the subshot keyframes. For each keyframe, we record its timestamp and image data as metadata.

Static subshot. A static subshot represents a scene in which the objects are static and background merely changes. Therefore, we can use one of the frames in the image sequence to represent the whole subshot. Here we simply select the middle frame in the subshot as the keyframe, and record its timestamp and image data as metadata.

4.3 Video Summary

The video summary in the NLVS consist of three components: 1) a XML file described the time and motion information, 2) images extracted from the original video, and 3) the compressed audio track.

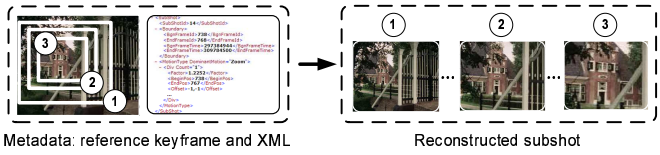


Figure 7: Reconstruction of a *zoom* subshot.

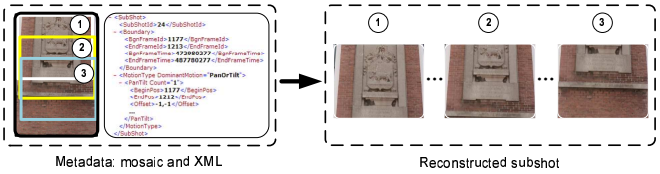


Figure 8: Reconstruction of a *translation* subshot.

XML file. Figure 6 illustrates a typical description of XML file. Subshots 0, 14, and 24 represent the time and motion information for the *object* (also for *static*), *zoom*, and *translation* subshots, respectively.

Images. There are two types of images in the metadata—mosaics and compressed keyframes. The mosaic images are stored in the JPEG format with *quality* = 95% [13] and resized to 1/2 of original scale. For the keyframes which contain much redundancy about the same scene, we employ a clustering based grouping and compression scheme to reduce the redundancy as much as possible. We only perform this process on the keyframes as mosaic is inherently a compact form and with different resolutions. The first keyframe from each subshot is chosen as the representative keyframe. Then, K-means clustering is performed in these representative keyframes using color moment feature with N_c clusters [24]. All the keyframes are arranged orderly in a sequence within each cluster. We employ H.264 baseline profile to compress the keyframe sequence [5].

Compressed low bit rate audio track. We employ a low bit rate audio compression standard, i.e., AMR [7], to compress the audio track for its scalability. We adopted 6.7 kbps profile in the NLVS for the sake of quality and storage consumption.

5. VIDEO RECONSTRUCTION

Towards reconstructing a video from the metadata, we need to parse the XML file by subshot index, as well as extract all the keyframes from the H.264 compressed file. Then we present how to reconstruct the video frame by frame in each subshot. Different mechanisms are proposed for different subshot types, i.e., *zoom*, *translation*, *object* and *static*.

5.1 Zoom Subshot

To reconstruct a subshot of zoom, we simulate the camera motion on the selected keyframes, taking *zoom-in* as an example below. We first simulate the subshot as a constant speed *zoom-in* procedure in which the zoom factor between successive frames is a constant $\sqrt[N_i-1]{Z^{acc}(S_i)}$ in one subshot. To reconstruct the j-th frame in the subshot S'_i, we calculate the zoom factor of the j-th frame referring to the first keyframe as

$$Z(F'_{i,j}) = \left(\sqrt[N_i-1]{Z^{acc}(S_i)}\right)^{j-1},\ (j = 2, \ldots, N_i) \qquad (6)$$

where N_i is the number of frames in S_i. Moreover, the camera focus of each frame with respect to the keyframe is calculated from the wrapping process. To construct a smooth wrapping path for frame reconstruction, a Gaussian filter is employed to eliminate the jitter of camera focus trajectory. For simplicity, we directly used a five-point Gaussian template $[\frac{1}{16}, \frac{4}{16}, \frac{6}{16}, \frac{4}{16}, \frac{1}{16}]$ to perform convolution over the trajectory parameters in the summary. When reconstructing the j-th frame in the subshot, as shown in Figure 7, we first shift the center of the keyframe with the smoothed camera focus and then resize the keyframe with zoom factor $Z(F'_{i,j})$. Finally, we carve the original frame from the resized keyframe with respect to the camera focus offset.

5.2 Translation Subshot

As mentioned in Section 4.2, a *translation* subshot consists of one or more units. Therefore, we reconstruct these units by simulating the camera focus trajectory along the mosaic, which include two steps, i.e., camera focus trajectory smoothing and frame reconstruction. As the generation of camera focus is the same in both *zoom* and *translation* subshot, we perform camera focus trajectory smoothing with the same mechanism for *zoom* subshot. When reconstructing the j-th frame in the *translation* subshot, we simulate the smoothed trajectory of camera focus along the mosaic and then carve the original frame from the mosaic. Figure 8 shows an example of reconstructing a *translation* subshot.

5.3 Object Subshot

To reconstruct the subshot, we simulate the object motion with gradual evolution of selected keyframes. Considering the efficiency and visually pleasure experience of the reconstruction video, we employ a fixed-length cross-fade transition between each keyframe to simulate the object motion. By modifying the fade-in and fade-out expression in [12], we define the following cross-fade expression to reconstruct j-th frame $F'_{i,j}$ in subshot S'_i

$$F'_{i,j} = \begin{cases} KF_{i,k} & 0 \leqslant j \leqslant l_i \\ (1-\alpha) \times KF_{i,k} + \alpha \times KF_{i,k+1} & l_i \leqslant j \leqslant l_i + L \\ KF_{i,k+1} & l_i + L \leqslant j \leqslant 2l_i + L \end{cases}$$

where $\alpha = \frac{j-l_i}{L}$, $2l_i + L = N_i$, and the length of the cross-fade L is set as $0.5 \times fps$ frames.

5.4 Static Subshot

For the *static* subshot, we choose one of the frames in the image sequence to represent the whole subshot. Here we reconstruct the frames in the subshot by simply copying the selected keyframe. After subshot reconstruction, all the frames in each subshot are reconstructed using the metadata. Finally, all the reconstructed frames are resized to original scale for video generation. We then integrate the reconstructed frames sequentially and the compressed audio track into a new video with the same duration as original.

6. EVALUATIONS

We evaluated the proposed NLVS from the following aspects: subshot classification, storage consumption, subjective visual effects, as well as the effectiveness of summary compared with state-of-the-art video summarization and compression techniques.

Table 2: Format of the data set.

	File format	Resolution (pixel)	Video		Audio	
			Codec	Bit rate (kbps)	Codec	Bit rate (kbps)
TVS videos	mpg	352×288	MPEG-1	1,150	MPEG-1	192
HDTV videos	dvr-ms	720×480	MPEG-2	6,800	MP2	384
Online videos	flv	320×240	H.263	284 ∼ 398 (347 in average)	MP3	64

Table 3: Performance of Subshot Classification.

	Zoom	Translation	Object	Static
Precision	0.85	0.95	0.98	0.92
Recall	0.92	0.94	0.96	0.97

Table 4: Keyframe compression ratio (CR) with different N_c.

N_c	1	2	4	10	20	30
$\frac{1}{2}$ Scale	0.2357	0.2358	0.2367	0.2371	0.2379	0.2380
Org. Scale	0.5772	0.5776	0.5793	0.5803	0.5812	0.5812

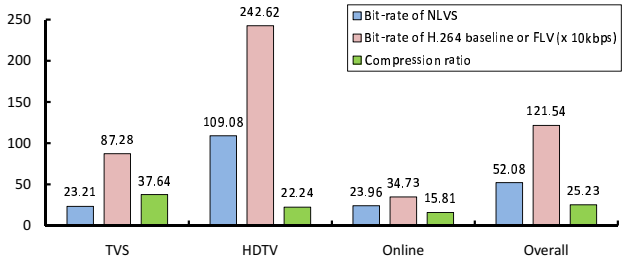


Figure 9: Results of storage consumption (in kbps).

6.1 Experimental Settings

We collected 35 representative videos with 682 minutes and 6,543 subshots in total, including 25 videos from BBC rush data set from TRECVID 2007 [1] (TVS videos), 3 HDTV programs (HDTV videos), and 7 online videos. Table 2 lists the formats of these videos. The thresholds for NLVS were set as $T_{p/t} = 200$, $T_{om} = 800$, $T_f = 10$.

6.2 Evaluation of Subshot Classification

To evaluate the motion threshold-based approach for subshot classification [20], we randomly selected 18 videos from the data set, and invited a volunteer to manually label the subshot types (i.e., *zoom*, *translation*, *object*, and *static*). The performance is listed in Table 3. It is observed that the classification achieves satisfying performance in terms of precision and recall.

6.3 Evaluation of Storage Consumption

We first evaluate the clustering with different cluster numbers N_c and image scales, as well as compare the compressed H.264 file size with original JPEG keyframe in terms of compression ratio (CR). Table 4 lists the results. We can see that the best compression ratio is obtained when the cluster number N_c=1 in both scales. The storage consumption of video summary in the NLVS includes the mosaic images, the keyframe sequence compressed by H.264, the XML file, and the compressed audio track. As it is difficult to evaluate the storage among different videos by using file size, we adopted bit rate as the metric for the storage consumption. Since H.264 becomes the most popular codec on the Internet, its baseline profile was adopted to compress the TVS and HDTV videos, while MP3 was adopted to compress the audio track at 128 kbps. Then, we compared NLVS with H.264 baseline profile (named "H.264.baseline") in terms of storage. For the online videos, we compared NLVS with FLV. The results of all 35 videos are shown in Figure 9. We can see that with NLVS, the storage can be significantly reduced compared with H.264 and FLV.

6.4 Evaluation of Video Summarization

As evaluating video summarization is highly subjective, we carried out a user study to perform the evaluation. We invited 30 volunteers including 15 males and 15 females with diverse backgrounds (i.e., education, literature, architecture, and so on.) and varies degrees of video browsing experience to participate the user studies over 35 videos. We generated six different forms for each video according to the following compression and summarization techniques.

(1) **Original video (Ori.).** We directly used the original TVS and online videos, without any codec conversion.

(2) **NLVS reconstructed video (NLVS).** We summarized the 35 videos using NLVS and then generated reconstruction video with the same duration.

(3) **H.264 compressed video with rate control (H.264.rc).** We compressed the videos by H.264 to make the compressed signals be with the same storage consumption with NLVS (i.e., the same bit rate setting and audio track setting).

(4) **Static Summarization (Static).** For each subshot, we selected one keyframe and arranged them on a one-dimensional storyboard.

(5) **Dynamic Summarization 1 (Dyn.1).** We used the dynamic skimming technique proposed in [22] to summarize the original videos, so that the summaries are with the same filesize as those of NLVS [3].

(6) **Dynamic Summarization 2 (Dyn.2).** We adopted the video skimming scheme proposed by CMU in TRECVID 2007 [14] to summarize the videos. The sample rate is set as the same as (5).

Note that we only evaluated the reconstructed video by NLVS, as well as other five forms of video summaries which are derived from the original video. We did not evaluated the NLVS metadata. An analogy can be drawn with the video coding standard: the metadata of NLVS is like the compressed signal, while the NLVS reconstructed video can

[3] Sample rate is the skimming ratio in video skimming scheme. For example, if we perform video skimming on a video with duration of 10 minutes, and get a video summary with duration of 1 minute, then the sample rate is 10%.

(a) Original (585kbps) (b) NLVS (13kbps) (c) H.264 (13kbps)

Figure 10: Example frames of Ori., NLVS, and H.264.rc. (Note: bit rates are computed without considering audio track, as only visual information is shown in this figure.)

be taken as the decoded video stream. As a result, we evaluated the dynamic summarizations, H.264, and NLVS with different forms of decoded/reconstructed video streams under the same setting (i.e., they are with the same file size). We only assigned one form to each user to guarantee none would watch more than two forms from the same video.

Figure 10 shows some example frames reconstructed by NLVS, as well as the corresponding frames of original video and H.264.rc. As we have mentioned in Section 5 that the simulation of camera motion and frame transition will introduce displacement in *zoom* and *translation* subshots, as well as fold-over in *object* subshot. However, the reconstructed frames by NLVS still achieved the comparable visual quality with the original. Through with the same bit rate as NLVS, frames of H.264.rc contain large amount of blocking artifacts, which is mainly due to the extremely low bit rate.

In addition to the above comparisons in terms of visual quality, we employed a signal-based quantitative performance metric, i.e., *PSNR/shot* of YUV color space, to compare the NLVS with H.264.rc at the same bit rate setting. We obtained *PSNR/shot* by averaging the $PSNR$ over all the frames in one shot and showed the results of the same video in Figure 11. We can observed that NLVS obtains comparable capability to H.264.rc in signal maintenance of channel U and V, while in channel Y, *PSNR/shot* is degraded lower than 40db in both NLVS and H.264.rc due to their extremely low bit rates. For NLVS, the degradation is mainly caused by the displacement in *zoom* and *translation* subshots, as well as fold-over in *object* subshot. Though the influence of such degradation is slight as illustrated in Figure 10, it will affect the signal perception. For H.264.rc, the setting of quantization step and block-based coding scheme in extremely low bit rate degraded the signal precision.

6.4.1 Content Maintenance

In traditional video summarization schemes, despite the keyframe-based or the video skimming-based approach, the sample rate and content maintenance are always two conflicting goals. However, with NLVS, we can reconstruct the metadata to a video with the same duration as original video. Through the summary, users can recover the whole story of the original video without any semantic information loss. We evaluated content maintenance along two dimensions, i.e., *content comprehension* and *content narrativeness*.

- **Content comprehension.** We design questionnaires for each video, including 10 questions for each video, covering the content of original video from the beginning to the end. Such as "Who is the baby's father," "How many people appear in this clip of video," etc. The content comprehension score was added 1 if current user came out with the correct answer. In this way, we can get an average score from 0 to 10 for each video per question. Then, the scores for all the questions are averaged as the final content comprehension.

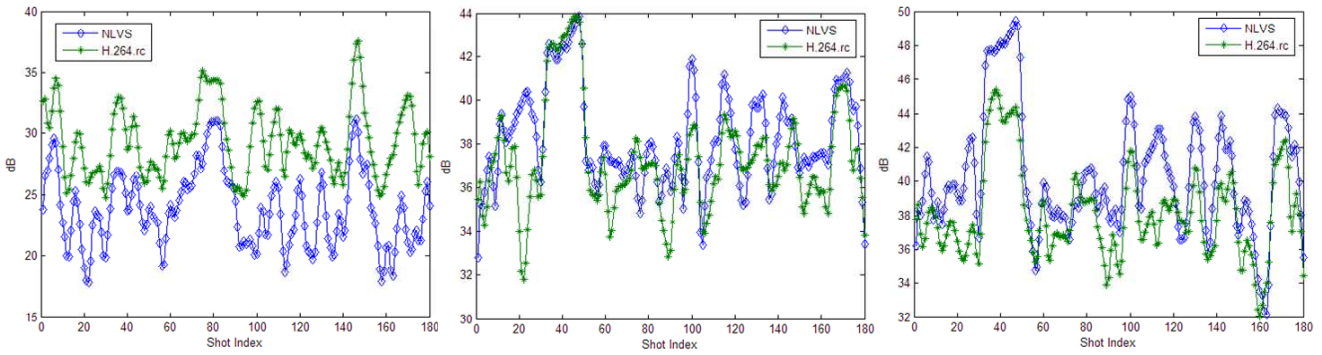
- **Content narrativeness.** The volunteers were asked to write down what they had seen in the videos or image sequences in time order. Then, a score between 1 and 10 was assigned to each video by assessing the users' descriptions according to a pre-generated ground-truth.

Figure 12 shows the results of content comprehension and content narrativeness. As the volunteers had different experiences and the difficulty of questionnaires varied with video data, we normalized the score of content comprehension and content narrativeness. We set both scores for the original video as 10, and scaled the scores of other five videos with respect to the score of the original video. From the results, original video has advantage in content coverage, while NLVS also achieves a comparable performance due to its capability of information maintaining. On the other hand, due to the extremely low bit rates, H.264.rc has lower performance, but still is superior to the static and dynamic summarizations. For traditional video summarization (i.e., Dyn.1, Dyn.2, and Static), we can find that static summarization outperforms the other two in terms of both content comprehension and content narrativeness. The main reason is the extremely low sample rate. In our experiment, the sample rate was set as about $0.01 \sim 0.05$ for TVS videos and $0.05 \sim 0.13$ for online videos. With the same sample rate setting, video skims generated by Dyn.1 and Dyn.2 only covered very small proportions of the original videos, thus lost most of the information. This leaded to the degradation in both content comprehension and content narrativeness. Since static summarization kept one keyframe for each subshot, volunteers can infer the time evolving information from the sequence of static keyframes. On the other hand, only one keyframe for a subshot is far from enough to describe the Five Ws (i.e., who, what, when, where, why, and how) in each scene. Therefore, static summarization also degraded sharply in content narrativeness.

6.4.2 User Impression

In this section, we evaluate user impression in terms of three criteria: *visual smoothness*, *visual sharpness*, and *satisfaction*.

- **Visual Smoothness.** Visual smoothness not only measures the content consistence through the video



(a) Y-channel (b) U-channel (c) V-channel

Figure 11: *PSNR/shot* in YUV color space.

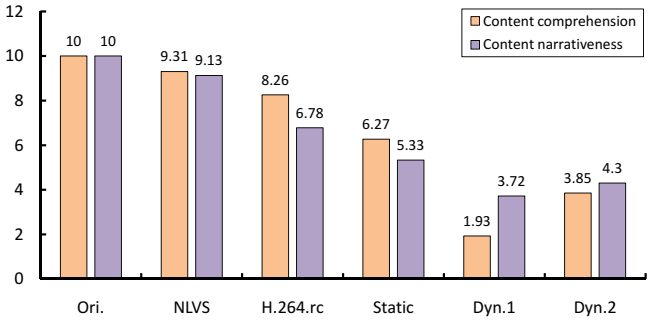


Figure 12: Results of content maintenance.

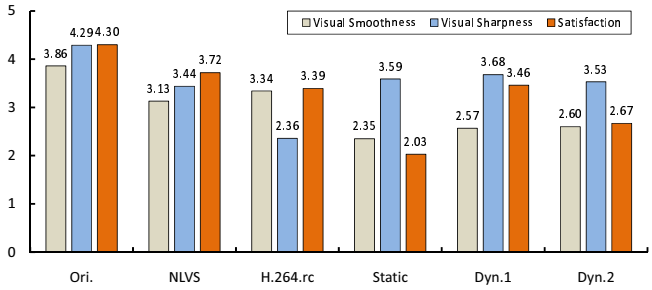


Figure 13: Results of user impression.

stream, but also reflects the smoothness of camera motion and continuity of object movement.

- **Visual Sharpness.** Visual sharpness measures a video from the following aspects: shape, color and texture of object; the direction and intensity of object movement; distinguishing of foreground and background; the small change of color and illumination.

- **Satisfaction.** Satisfaction measures the user's enjoyability of the video.

Each user was required to assign a score of 1 to 5 (higher score indicating better experience of the above criteria) to the three criteria, respectively. Figure 13 shows the results of user impression. We can see the original video clearly outperforms the other method with the varying grades with videos. NLVS also got considerable grades. However, due to the sustained resized process, the visual sharpness of NLVS was degraded, but still acceptable. Due to the extremely low bit rate, H.264 was degraded sharply in visual sharpness, while this degradation also affected its visual smoothness and satisfaction. For static and dynamic summarization, the visual smoothness was low, mainly due to their low sample rates. Their satisfaction grades were also influenced by the difficulty of content understanding.

7. CONCLUSIONS AND FUTURE WORK

We have presented a novel system to tackle with large-scale video storage which greatly differs from traditional video compression and summarization techniques. The proposed near-lossless video summarization (NLVS) is able to achieve extremely low storage consumption and can be used to reconstruct the original video without least semantic loss (i.e., high visual fidelity yet the same duration). Compared with existing popular video coding standards such as H.264, NLVS achieves much lower compression ratio at which H.264 fails to preserve satisfying visual fidelity; while compared with conventional video summarization techniques, NLVS can keep much more information.

However, we are aware of some limitations in NLVS. For example, the efficiency and accuracy of subshot classification highly depend on the computationally intensive estimation of affine model. Hence it remains a challenging problem to speed up the motion computation while keep information as much as possible. In addition, the mosaic for the *translation* subshot cannot well preserve object motion. Therefore, a more effective method for compact representation is desirable. Our future work includes applying NLVS to more applications in video search tasks, such as near-duplicate video detection and video annotation based on the summary. Moreover, we aim at investigating human factor for scalable summarization to further reduce the storage consumption with keeping information loss as least as possible.

8. REFERENCES

[1] TREC video retrieval evaluation. *http://www-nlpir.nist.gov/projects/trecvid/.*

[2] MPEG-1 VideoGroup, Information technology—coding of moving pictures and associated

audio for digital storage media up to about 1.5 Mbit/s: Part 2—Video. *ISO/IEC 11172-2*, 1993.

[3] MPEG-2 Video Group, Information technology—generic coding of moving pictures and associated audio: Part 2—Video. *ISO/IEC 13818-2*, 1995.

[4] MPEG-4 Video Group, Generic coding of audio-visual objects: Part 2—Visual. *ISO/IEC JTC1/SC29/WG11 N1902, FDIS of ISO/IEC 14 496-2*, 1998.

[5] ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification. March 2003.

[6] ITU-T Rec. H.263, Video coding for low bit rate communication. version 1, Nov. 1995; version 2, Jan. 1998; version 3, Nov. 2000.

[7] 3rd Generation Partnership Project. AMR speech codec: General description. *TS 26.071 version 5.0.0*, June 2002.

[8] Bing. http://www.bing.com/.

[9] P. Bouthemy, M. Gelgon, and F. Ganansia. A unified approach to shot change detection and camera motion characterization. *IEEE Trans. on Circuit and Syst. for Video Tech.*, 9(7):1030–1044, 1999.

[10] T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler. Extensible Markup Language (XML) 1.0 (Second Edition). *Available at http://www.w3.org/TR/REC-xml*, 2000.

[11] L. Carter. Web could collapse as video demand soars. *Daily Telegraph, http://www.telegraph.co.uk/news/uknews/1584230/Web-could-collapse-as-video-demand-soars.html*.

[12] W. A. C.Fernando, C. N. Canagarajah, and D. R. Bull. Automatic detection of fade-in and fade-out in video sequences. *Proceedings of IEEE International Symposium on Circuits and Systems*, pages 255–258, 1999.

[13] Digital Compression and Coding of Continuoustone Still Images, Part 1. Requirements and guidelines. *ISO/IEC JTC1 Draft International Standard 10918-1*, Nov. 1991.

[14] A. G. Hauptmann, M. G. Christel, W.-H. Lin, and etc. Clever clustering vs. simple speed-up for summarizing rushes. *Proceedings of the International Workshop on TRECVID Video Summarization*, pages 20–24, 2007.

[15] X.-S. Hua, L. Lu, and H.-J. Zhang. Optimization-based automated home video editing system. *IEEE Trans. on Circuit and Syst. for Video Tech.*, 14(5):572–583, 2004.

[16] Hulu. http://www.hulu.com/.

[17] M. Irani and P. Anandan. Video indexing based on mosaic representations. *Proceedings of the IEEE*, 86(5):905–921, 1998.

[18] JVT Reference Software version 15.1A. http://bs.hhi.de/suehring/tml/.

[19] C. Kim and J.-N. Hwang. Object-based video abstraction for video surveillance systems. *IEEE Trans. on Circuit and Syst. for Video Tech.*, 12(12):1128–1138, 2002.

[20] J. G. Kim, H. S. Chang, J. Kim, and H. M. Kim. Efficient camera motion characterization for MPEG video indexing. *In Proceedings of ICME*, pages 1171–1174, 2000.

[21] J. Konrad and F. Dufaux. Improved global motion estimation for N3. *ISO/IEC JTC1/SC29/WG11 M3096*, 1998.

[22] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li. A user attention model for video summarization. *Proceedings of ACM Multimedia*, 2002.

[23] R. Marzotto, A. Fusiello, and V. Murino. High resolution video mosaicing with global alignment. *Proceeding of CVPR*, pages 692–698, 2004.

[24] T. Mei, X.-S. Hua, W. Lai, L. Yang, and et al. MSRA-USTC-SJTU at TRECVID 2007: High-level feature extraction and search. In *TREC Video Retrieval Evaluation Online Proceedings*, 2007.

[25] T. Mei, X.-S. Hua, H.-Q. Zhou, S. Li, and H.-J. Zhang. Efficient video mosaicing based on motion analysis. In *Proceedings of IEEE International Conference on Image Processing*, pages 861–864, 2005.

[26] T. Mei, X.-S. Hua, C.-Z. Zhu, H.-Q. Zhou, and S. Li. Home video visual quality assessment with spatiotemporal factors. *IEEE Trans. on Circuit and Syst. for Video Tech.*, 17(6):699–706, June 2007.

[27] Metacafe. http://www.metacafe.com/.

[28] Revver. http://www.revver.com/.

[29] E. Schonfeld. YouTube's Chad Hurley: "We Have The Largest Library of HD Video On The Internet.". *TechCrunch. http://www.techcrunch.com/2009/01/30/youtubes-chad-hurley-we-have-the-largest-library-of-hd-video-on-the-internet/.*

[30] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 3(1):692–698, 2007.

[31] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):560–576, July 2003.

[32] YouTube. http://www.youtube.com/.

[33] H.-J. Zhang. *Content-Based Video Analysis, Retrieval and Browsing*. Academic Press, 2002.

[34] H.-J. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, June 1993.