An Overview of Video Abstraction Techniques

Ying Li, Tong Zhang, Daniel Tretter
Imaging Systems Laboratory
HP Laboratories Palo Alto
HPL-2001-191
July 31st , 2001*

video abstraction, video skimming, video summarization

The fast evolution of digital video has brought many new applications. Consequently, research and development of new technologies are greatly needed which will lower the costs of video archiving, cataloging and indexing, as well as improve the efficiency and accessibility of stored videos. Among all possible research areas, video abstraction is one of the most important topics, which helps to enable a quick browsing of a large collection of video data and to achieve efficient content access and representation.

There are two fundamentally different types of video abstracts: still- and moving-image abstracts. The still-image abstract, also known as a static storyboard, is a small collection of salient images extracted or generated from the underlying video source. The moving-image abstract, also known as moving storyboard, consists of a collection of image sequences, as well as the corresponding audio abstract extracted from the original sequence and is thus itself a video clip but of considerably shorter length.

In this report, we present an overview of major technologies employed by each type of video abstraction, respectively. A list of important players in this research area, from both universities and the industry, is also provided.

* Internal Accession Date Only
Approved for External Publication

An Overview of Video Abstraction Techniques

Ying Li, Tong Zhang and Daniel Tretter

1 Introduction

Digital video is an emerging force in today's computer and telecommunication industries. The rapid growth of the Internet, in terms of both bandwidth and the number of users, has pushed all multimedia technology forward including video streaming. Continuous hardware developments have reached the point where personal computers are powerful enough to handle the high storage and computational demands of digital video applications. DVD, which delivers high quality digital video to consumers, is rapidly penetrating the market. Moreover, the advances in digital cameras and camcorders have made it quite easy to capture a video and then load it into a computer in digital form. Many companies, universities and even ordinary families already have large repositories of videos both in analog and digital formats [1][2], such as the broadcast news, training and education videos, advertising and commercials, monitoring, surveying and home videos. All of these trends are indicating a promising future for the world of digital video.

The fast evolution of digital video has brought many new applications and consequently, research and development of new technologies, which will lower the costs of video archiving, cataloging and indexing, as well as improve the efficiency, usability and accessibility of stored videos are greatly needed. Among all possible research areas, one important topic is how to enable a quick browse of a large collection of video data and how to achieve efficient content access and representation. To address these issues, video abstraction techniques have emerged and have been attracting more research interest in recent years.

Video abstraction, as the name implies, is a short summary of the content of a longer video document. Specifically, a video abstract is a sequence of still or moving images representing the content of a video in such a way that the target party is rapidly provided with concise information about the content while the essential message of the original is well preserved [3]. Theoretically a video abstract can be generated both manually and automatically, but due to the

huge volumes of video data and limited manpower, it's getting more and more important to develop fully automated video analysis and processing tools so as to reduce the human involvement in the video abstraction process. This paper will mainly focus on the technology of automatic video abstraction.

There are two fundamentally different kinds of abstracts: *still-* and *moving-image abstracts*. The still-image abstract, also known as a static storyboard, is a small collection of salient images extracted or generated from the underlying video source. In this report, we call this type of abstract a video summary. The moving-image abstract, also known as moving storyboard, or multimedia summary, consists of a collection of image sequences, as well as the corresponding audio abstract extracted from the original sequence and is thus itself a video clip but of considerably shorter length. In this report, we call this type of abstract a video skimming.

There are some significant differences between video summary and video skimming. A video summary can be built much faster, since generally only visual information is utilized and no handling of audio and textual information is needed. Therefore, once composed, it is displayed more easily since there are no timing or synchronization issues. Moreover, more salient images such as mosaics could be generated to better represent the underlying video content instead of directly sampling the video frames. Besides, the temporal order of all extracted representative frames can be displayed in a spatial order so that the users are able to grasp the video content more quickly. Finally, all extracted stills could be printed out very easily when needed.

There are also advantages using video skimming. Compared to a still-image abstract, it makes much more sense to use the original audio information since sometimes the audio track contains important information such as those in education and training videos. Besides, the possibly higher computational effort during the abstracting process pays off during the playback time: it's usually more natural and more interesting for users to watch a trailer than watching a slide show, and in many cases, the motion is also information-bearing. We'll look into more technical details of both types of video abstracts in the rest of this report.

2 Video Skimming

There are basically two types of video skimming: *summary sequence* and *highlight* [4]. A summary sequence is used to provide users an impression about the entire video content, while a highlight only contains the most interesting parts of the original video, like a movie trailer that only shows some of the most attractive scenes without revealing the story's end. In the VAbstract system developed by the *University of Mannheim, Germany* [3], the most characteristic movie segments are extracted for the purpose of automatically producing a movie trailer. Specifically, the scenes containing important objects/people are detected by finding the frames with high-contrast; the high-action scenes are extracted by picking up the frames having largest frame differences; also, the scenes that have a basic color composition similar to the average color composition of the entire movie, are included in the abstract with the hope that they may represent the basic mood of the original movie; moreover, the recognition of dialog scenes is performed by detecting the spectrum of a spoken "a" since "a" occurs frequently in most languages. Finally all selected scenes (except the last part of the movie), organized in their original temporal order, forms the movie trailer. There are some interesting ideas in this paper, but some parts of the algorithm are too simple to be effective and will need lots of improvement. The researchers also lack thorough user studies to support their conclusions. An improved version of VAbstract, called MoCA Abstracting, could be found in [16] where special events, such as closed-up shots of leading actors, explosions and gunfire, are detected to help determine the important scenes.

Defining which video segments are the highlights is actually a very subjective process, and it's also a very hard project to map human cognition into the automated abstraction process, thus most of existing video-skimming work focuses on the generation of a summary sequence. One of the most straightforward approaches in this case would be to compress the original video by speeding up the playback. As studied by Omoigui, et al. at *Microsoft Research* [11], the entire video could be watched in a shorter amount of time by fast playback with almost no pitch distortion using the time compression technology. Similar work is also reported by Amir, et al. in

IBM Almaden Research Center by using audio time scale modification technology [12]. These techniques, however, only allow a maximum time compression of 1.5-2.5 depending on the speech speed [13], beyond which the speech becomes incomprehensible.

The Informedia Project at *Carnegie Mellon University* (Department of Electrical and Computer Engineering) [5][6][7] aims to create a very short synopsis of the original video by extracting the significant audio and video information. Particularly, text keywords are first extracted from manual transcript and closed captioning by using the well-known TF-IDF (Term-Frequency-Inverse Document Frequency) technique, then the audio skimming is created by extracting the audio segments corresponding to the selected keywords as well as including some of their neighboring segments for better comprehension. Next, the image skimming is created by selecting the video frames which are: a) frames with faces or texts; b) static frames following camera motion; c) frames with camera motion and human faces or text, and d) frames at the beginning of a video scene, with a descending priority. As a result, a set of video frames, which may not align with the audio in time, but may be more appropriate for image skimming in visual aspect are extracted. Finally the video skimming is generated by analyzing the word relevance and the structure of the prioritized audio and image skimming. Experiments of this skimming approach have shown impressive results on limited types of documentary video that have very explicit speech or text contents. However, satisfying results may not be achievable using such a text-driven approach on other videos with a soundtrack containing more complex audio contents. Recently some improvements of their algorithms have been made and a subjective evaluation of this project is reported in [8].

In [9], Toklu and Liou from *Siemens Corporate Research* (Multimedia and Video Technology Department) reported their work on video skimming where multiple cues are employed including visual, audio and text information. Specifically, they first group detected shots into story units based on detected “change of speaker” and “change of subject” markers that are sometimes available from the closed captioning. Then audio segments corresponding to all generated story units are extracted and aligned with the summarized closed-caption texts. Representative images

are also extracted for each story unit from a set of keyframes consisting of all first frames of the underlying shots. Their final video skimming includes the audio and text information, but somehow the keyframe information is excluded from it. Finally, users are allowed to return their feedback to the system through an interactive interface so as to adjust the generated video skimming to their satisfaction. Similar to the Informedia project, this work also depends heavily on the text information. To avoid this situation, Nam and Tewfik from *University of Minnesota* (Department of Electrical and Computer Engineering) [10] propose to generate the skimming based on a dynamic sampling scheme. Specifically, they first decompose the continuous video source into a sequence of "sub-shots", where a motion intensity index is computed for each of them. Next, all indices are quantized into predefined bins, with each bin assigned a different sampling rate. Finally keyframes are sampled from each sub-shot based on the assigned rate. During the skimming playback, linear interpolation is performed to provide users a moving storyboard. While this work has avoided using the text information and the generated skimming is video content-adaptive, no discussions on how to handle the accompanying audio track is reported. Also, the work lacks a discussion on how to choose the number of predefined quantization levels, which is obviously a very important parameter in their approach. In the work reported by Hanjalic and Zhang (*Delft University of Technology*, Netherlands and *Microsoft Research*, China) [4], they first try to cluster all video frames into n clusters, with n varies from 1 to N. Then, a cluster-validity analysis is performed to determine the optimal number of clusters, i.e. the optimal value of n. One representative frame is then chosen from each of these clusters which forms the final keyframe sequence. Lastly, the skimming is generated by concatenating all video shots which contain at least one extracted keyframe. Although theoretically this method can be applied to a video sequence of an arbitrary length, the sequences of interest in this paper are rather constrained to specific events with a well-defined and reasonably structured content.

Some other work in this area tries to find solutions for domain-specific videos where special features can be employed. The VidSum project developed at *Xerox PARC* (Xerox Palo Alto Research Center) uses a presentation structure, which is particularly designed for their regular

weekly forum, to assist in mapping low-level signal events onto semantically meaningful events that can be used in the assembly of the summary [14]. In [15], He et al. from *Microsoft Research* reported their summarization work on audio-video presentations where informational talks are given with a set of slides. Some special knowledge about the presentation is utilized including the pitch and pause information, the slide transition points and the information about the access patterns of previous users. A detailed user study shows that although computer-generated summaries are less coherent than manually generated summaries, most of informative parts of the original presentation are well preserved. Another work reported by Lienhart from *Intel Corporation* mainly focuses on the summarization of home videos [16] where it is more usage model-based than content-based. First, the time and date information of the recordings are obtained by either extracting them from the S-VHS using text segmentation and recognition algorithms, or by directly accessing them from the digital video sequence. Then, all shots are clustered into 5 different levels based on the date and time they are taken, which include: the individual shots; a sequence of contiguous actions where the temporal distance between shots are within 5 minutes; a sequence of contiguous activities where temporal distance between shots are within 1 hour; individual days and individual multi-day events. In the next step, a shot shortening process is performed where longer shots are uniformly segmented into 2-minute-long clips. To choose the desired clips, the sound pressure level of the audio signal is calculated and employed in the selection process based on the observation that during important events, the sound is usually more clearly audible over a long period of time than is the case with less important content. Finally, all selected clips are assembled to form the final abstract using pre-designed video transition effects. This work also supports on-the-fly annotations with 2 microphones attached to the camcorder and voice transcribed using the Dragon Naturally Speaking package. There are certainly some interesting ideas for analyzing and processing home videos in this paper, yet the video content, which should be the focus of any summarization work, is unfortunately neglected by this paper.

3 Video Summary

Compared to video skimming, there is much more work going on in the video summary area, which is probably due to the reasons we discussed earlier. Because the video summary is basically a collection of still images that best represent the underlying content, the extraction or generation of these stills (called as keyframes in this report) becomes the main focus of all summary work. Based on the way the keyframes are constructed, we briefly categorize all related work into 4 classes: *sampling-based*, *shot-based*, *segment-based*, and *others*. We'll elaborate on them in the rest of this report.

3.1 Sampling-based Keyframe Extraction

Most of earlier work in video summarization chose to select keyframes by randomly or uniformly sampling the video frames from the original sequence at certain time intervals, which was applied in the Video Magnifier [18], MiniVideo system [19] and [20]. Although this is probably the simplest way to extract keyframes, the drawback is that such an arrangement may cause some short yet important segments to have no representative frames while longer segments could have multiple frames with similar content, thus failing to capture and represent the actual video content.

3.2 Shot-based keyframe Extraction

More sophisticated work tends to extract keyframes by adapting to the dynamic video content. Since a shot is defined as a video segment within a continuous capture period, a natural and straightforward way of keyframe extraction is to use the first frame of each shot as its keyframe [21][22][23][24][25][26]. However, while being sufficient for stationary shots, one keyframe per shot does not provide an acceptable representation of dynamic visual content, therefore multiple keyframes need to be extracted by adapting to the underlying semantic content. However, since computer vision still remains to be a very difficult research challenge, most of existing work chooses to interpret the content by employing some low-level visual features such as color and

motion, instead of performing a tough semantic understanding. In this report, based on the features that these works employ, we categorize them into following 4 different classes: color-based approach, motion-based approach, mosaic-based approach and others.

3.2.1 Color-based Approach

In the work reported in [27] and detailed in [28] by Zhang et al., the keyframes are extracted in a sequential fashion for each shot. Particularly, the first frame within the shot is always chosen as the first keyframe, then the color-histogram difference between the subsequent frames and the latest keyframe is computed. Once the difference exceeds a certain threshold, a new keyframe will be declared. A similar work is also reported by Yeung and Liu [29]. One possible problem with above extraction methods is that there is a probability that the first frame is a part of transition effect at the shot boundary, thus strongly reducing its representative quality. In [30], Huang et al. from *University of Illinois at Urbana-Champaign* propose to extract the keyframes using an unsupervised clustering scheme. Basically, all video frames within a shot are first clustered into certain number of clusters based on the color-histogram similarity comparison where a threshold is predefined to control the density of each cluster. Next, all the clusters that are big enough are considered as the key clusters, and a representative frame closest to the cluster centroid is extracted from each of them. Ferman and Tekalo reported a similar work in [32].

Because the color histogram is invariant to image orientations and robust to background noises, color-based keyframe extraction algorithms have been widely used. However, most of these works are heavily threshold-dependent, and cannot well capture the underlying dynamics when there is lots of camera or object motion.

3.2.2 Motion-based Approach

Motion-based approaches are relatively better suited for controlling the number of frames based on temporal dynamics in the scene. In general, pixel-based image differences [34] or optical flow computation [31][35] are commonly used in this approach. In Wolf's work [31] (*Princeton*

University, Department of Electrical Engineering), the optical flow for each frame is first computed, and then a simple motion metric is computed. Finally by analyzing the metric as a function of time, the frames at the local minima of motion are selected as the keyframes. A domain specific keyframe selection method is proposed in [36] where a summary is generated for video-taped presentations. Sophisticated global motion and gesture analysis algorithms are developed. Toklu and Liou from *Siemens Corporate Research* reported their work in [37] where 3 different operation levels are suggested based on the available machine resources: at the lowest level, pixel-based frame differences are computed to generate the "temporal activity curve" since it requires minimal resources; at level 2, color histogram-based frame differences are computed to extract "color activity segments", and at level 3 sophisticated camera motion analysis is carried out to estimate the camera parameters and detect the "motion activity segments". Keyframes are then selected from each segment and necessary elimination is applied to obtain the final result.

3.2.3 Mosaic-based Approach

A limitation of above approaches is that it is not always possible to select the keyframes that can represent the entire video content well. For example, given a camera panning/tilting sequence, even if multiple keyframes are selected, the underlying dynamics still couldn't be well captured. In this case, the mosaic-based approach can be employed to generate a synthesized panoramic image that can represent the entire content in an intuitive manner. Mosaics, also known as salient stills [39], video sprites [40] or video layers [41], are usually generated in the following 2 steps [42]: 1) Fitting a global motion model to the motion between each pair of successive frames; 2) Compositing the images into a single panoramic image by warping the images with the estimated camera parameters. The MPEG-7 MDS document [38] lists some commonly used motion models including translational model, rotation/scaling model, affine model, planar perspective model and quadratic model. Figure 1 shows the mosaic generated using an affine model with 183 frames taken from a video segment. Here as we can see, this one single still image gives us much more information than one or several regular keyframes could do.

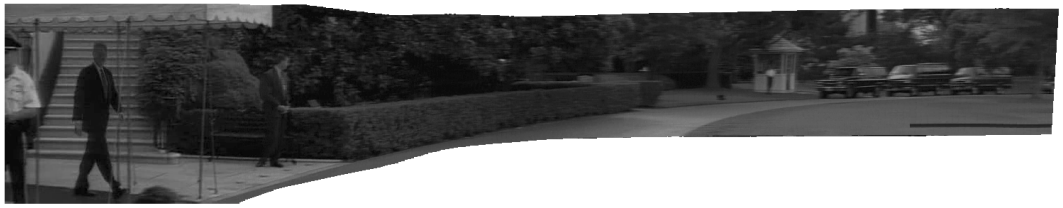

Figure 1: A mosaic image generated from 183 video frames

However, one drawback of above example is that it only provides an extended panoramic spatial view of the entire static background scene, but contains no information about the moving object in the foreground. In the above example, the camera actually panned from left to right to track the former president Bill Clinton as he walked across the yard; however, the generated mosaic does not actually provide us that information. To address this issue, Irani and Anandan proposed two types of mosaic in their work [43], the static background mosaic and the synopsis mosaic. While the static mosaic captures the background scene, the synopsis mosaic is constructed to provide a visual summary of the entire dynamic foreground event that occurred in the video clip by detecting the object trajectory. The final mosaic image is then obtained by simply combining both mosaics. Promising results are reported in this paper. In Vasconcelos and Lippman's work (*MIT Media Laboratory*) [42], a new global motion representation is proposed in both the spatial and temporal dimensions by augmenting the motion model with a generic temporal constraint.

However, despite the fact that the mosaic image is more informative and visually more pleasing than a regular keyframe, it has its own application limitation; it only works well when specific camera motions are detected, such as panning, tilting, etc. Generally they cannot be effectively applied to real world videos with complex camera effects and frequent background/foreground changes. One solution is to interchangeably use either regular keyframe or mosaic image, whichever is more suitable. Taniguchi et al. from *NTT Human Interface laboratory, Japan*, report their work towards this direction [44]. Specifically, given each shot, if a pan-tilt segment is detected, then a panoramic icon will be constructed, otherwise, the first frame of the shot will be used as the keyframe. Furthermore, to accommodate the irregular shape of the constructed mosaic image, a layout method for packing icons in a space-efficient manner is proposed.

3.2.4 Other Approaches

Some other work integrates certain mathematical methodologies into the summarization process based on low-level features. In the work reported by Doulamis et al. from *National Technical University of Athens* (Department of Electrical and Computer Engineering) [45], several descriptors are first extracted from each video frame by applying a segmentation algorithm to both color and motion domains, which forms the feature vector. Then all frames' feature vectors in the shot are gathered to form a curve in a high-dimensional feature space. Finally, keyframes are extracted by estimating appropriate curve points that characterize the feature trajectory, where the curvature is measured based on the magnitude of the second derivative of the feature vectors with respect to time. In [46], Stefanidis et al. from *University of Maine* (Department of Spatial Information Engineering) present an approach to summarize video datasets by analyzing the trajectories of contained objects. Basically, critical points on the trajectory that best describe the object behavior during that segment are identified and subsequently used to extract the keyframes. The Self-Organizing Map (SOM) technique is used to identify the trajectory nodes.

3.3 Segment-based Keyframe Extraction

One major drawback of using one or more keyframes for each shot is that it does not scale up for long videos since scrolling through hundreds of images is still time-consuming, tedious and ineffective. Therefore, recently more and more people begin to work on higher-level video unit, which we call a video segment in this report. A video segment could be a scene, an event, or even the entire sequence. In this context, the segment-based keyframe set will surely become more concise than the shot-based keyframe set.

In [47], Uchihashi et al from *FX Palo Alto Laboratory* first cluster all video frames into a predefined number of clusters, and then the entire video is segmented by determining to which cluster the frames of a contiguous segment belong. Next an importance measure is computed for each segment based on its length and rarity, and all segments with their importance lower than a

certain threshold will be discarded. The frame that is closest to the center of each qualified segment is then extracted as the representative keyframe, with the image size proportional to its importance index. Finally, a frame-packing algorithm is proposed to efficiently pack the extracted frames into a pictorial summary. Figure 2 shows one of their example summaries. In the work reported by Girgensohn and Boreczky also from *FX Palo Alto Laboratory* [33], *N* frames, which are mostly dissimilar from each other in terms of visual content, are first selected from all video frames, then they are classified into *M* clusters using a hierarchical clustering algorithm. Finally one representative frame is chosen from each cluster where temporal constraints are employed to help obtain a semi-uniform keyframe distribution. Yeung and Yeo from *IBM Thomas J. Watson Research Center* reported their work on summarizing video at the scene level [48]. In particular, based on the detected shot structure, they first classify all shots into a group of clusters using proposed "time-constrained clustering" algorithm. Then meaningful story units or scenes are extracted, from which 3 categories of temporal events are detected including dialogue, action and others. Next, for each story unit, a representative image (R-image) is selected to represent each of its component shot clusters, and a corresponding dominance value will be computed based on either the frequency count of those shots with similar visual content or the duration of the shots in the segment. All of the extracted R-images with respect to a certain story unit are then resized and organized into a single regular-sized image according to a predefined visual layout, which is called a video poster in their work. The size of each R-image is set such that the larger the image dominance, the larger its size. As a result, the video summary consists of a series of video posters with each summarizing one story unit and each containing a layout of sub-images that abstract the underlying shot clusters. Some interesting results are reported. However, there are two main drawbacks in their algorithms: 1) since the visual layouts are pre-designed and can't be adjusted to accommodate for the variable complexity of different story units, the shot clusters with low priority may not be assigned any sub-rectangles in the poster layout, thus losing their respective R-frames in the final summary; 2) the number of video posters is determined by the number of detected story units, thus the inaccuracy introduced in the scene detection algorithm will certainly affect the final

summarization result. Also, there is no way to obtain a scalable video summary, which may be desirable in certain cases.

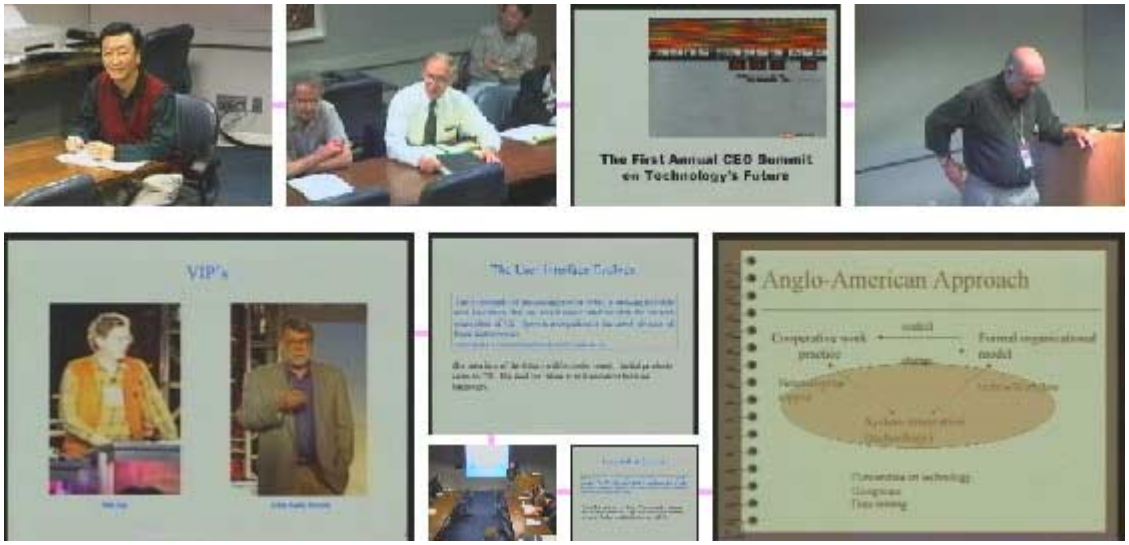

Figure 2: A video summary containing variable-sized keyframes

Sun and Kankanhalli from *National University of Singapore* (School of Computing) reported their work in [49] where no shot detection is needed. On the contrary, the entire video sequence is first uniformly segmented into *L*-frame long units, and then a unit change value is computed for each unit, which equals to the distance between the first and last frame of the unit. Next, all the changes are sorted and classified into *2* clusters, the small-change cluster and the large-change cluster, based on a predefined ratio *r*. Then for the units within the small-change cluster, the first and last frames are extracted as the R-frames, while for those in the large-change cluster, all frames are kept as the R-frames. Finally, if the desired number of keyframes has been obtained, the algorithm will stop, otherwise, the retained R-frames will be regrouped as a new video, and a new round of keyframe selection will be initialed. This work showed some interesting ideas, yet the uniform segmentation and subsequent two-class clustering may be too coarse. A simple color histogram-based distance between the first and last frame of a segment cannot truthfully reflect the variability of the underlying content, and if these two frames happen to have similar color composition, even if this segment is quite complex, it will still be classified

into the small-change cluster. Therefore, the final summarization result may miss significant parts of the video information while at the same time retaining all the redundancies of other video parts.

Based on Lagendijk et al.'s work [50][51], Ratakonda et al. from *University of Illinois at Urbana-Champa*ign (Department of Electrical and Computer Engineering) reported their work on generating a hierarchical video summarization [52] since a multilevel video summarization will facilitate quick discovery of the video content and enable browsing interesting segments at various levels of details. Specifically, given a quota of total number of desired keyframes, each shot is first assigned a budget of allowable keyframes based on the total cumulative actions in that shot, which forms their finest-level summary. To achieve coarser-level summary, a pair-wise K-means algorithm is applied to cluster temporally adjacent keyframes based on a predetermined compaction ratio *r*, where the number of iterations is controlled by certain stopping criterion, for instance, the amount of decrease in distortion, or a predetermined number of iteration steps. While this algorithm does produce a hierarchical summary, the temporal-constrained K-means clustering will not be able to merge two frames when they are visually similar but temporally apart. In some cases, a standard K-means will work better when preserving original temporal order is not required.

Doulamis et al. from *National Technical University of Athens* (Department of Electrical and Computer Engineering) introduced the fuzzy scheme into their summarization work [53]. Specifically, for each video frame, they first apply a recursive shortest spanning tree (RSST) algorithm to perform the color and motion segmentation, then a fuzzy classification is carried out to cluster all extracted color and motion features to predetermined classes, which then forms a fixed-dimensional feature vector. Finally keyframes are extracted from the video sequence by minimizing a cross-correlation criterion using a genetic algorithm (GA). The major drawback of this work is the high computational complexity required for extracting the fuzzy feature vector. Auephanwiriyakul et al. also employ a fuzzy scheme in their work [54] by fuzzily clustering

similar shots together since membership of a frame in some particular scene is not binary. The final summary consists of all the median frames of the clusters.

Compared with all above work, Dementhon et al. from *University of Maryland* (Language and Media Processing Lab) treat the video summarization task in a more mathematical way where a video sequence is represented as a curve in a high-dimensional feature space [55]. First, a 13-dimensional feature space is formed by the time coordinate and three coordinates of the largest "blobs" (image regions) using four intervals (bins) for each luminance and chrominance channel. Then the authors try to simplify the curve by using the multidimensional curve splitting algorithm, which results in a linearized curve, characterized by "perceptually significant" points that are connected by straight lines. The keyframe sequence is finally obtained by collecting frames found at those significant points. With a splitting condition that checks the dimensionality of the curve segment being split, the curve can also be recursively simplified at different levels of detail, where the final level is determined by a pre-specified threshold that evaluates the distance between the curve and its linear approximation. A major potential problem of this approach is the difficulty in evaluating the applicability of obtained keyframes, since there is no comprehensive user study to prove that the frames lying at "perceptually significant" points do capture all important instances of the video. Gong and Liu from *C&C Research Laboratory* (NEC USA) reported their work on using the SVD (Singular Value Decomposition) technique [56]. Specifically, given an input video sequence, they first create a frame matrix A with each column containing a feature vector with respect to one particular frame. Then an SVD is performed, which not only reduces the feature space dimensions, but also provides a metric that could be used to measure the amount of visual content contained in each frame cluster using its degree of visual changes. Next, the most static frame cluster is located, and its content value computed, which, together with the distance between frames, is used as the threshold to cluster the rest of the frames. Finally, the frames whose feature vectors are at the center of the clusters are chosen as the keyframes. One potential problem is that basically this approach will generate a set of visually dissimilar keyframes while their temporal order information is totally lost. A similar idea using Principle Component Analysis (PCA) technology could also be found in [57].

3.4 Other Keyframe Extraction Work

Other keyframe extraction work integrates some other technologies into their summarization framework, such as wavelet transform, face detection, etc. In Dufaux's work [58] (*Compaq Computer Corp*. Cambridge Research Lab), he integrates the motion and spatial activity analysis with skin-color and face detection technologies, so that the selected keyframe will have high likeliness of containing people or portraits, which certainly makes more sense than a landscape image. However, this work mainly focuses on choosing one single keyframe for the entire video sequence, which makes it quite difficult to evaluate the final summarization result. In Campisi et al.'s work [59], a progressive multi-resolution keyframe extraction technique based on wavelet decomposition is proposed. One of the main advantages of this approach is the possibility of controlling the coarseness of the details' variations which are taken into account in the selection of a keyframe by properly choosing the particular sub-band to analyze and the level of the pyramid decomposition. However, the overall computation complexity is relatively too high. Another work reported by Kim and Hwang from *University of Washington* (Department of Electrical Engineering) uses the segmented video object as the building block [60]. Particularly, the video objects in each frame are first segmented and identified using the Moving Edge (ME) map or the mathematical morphology technique. Then within each shot they set the first frame to be the first keyframe, and continuously compare the number of objects contained in each subsequent frame with that in the last extracted keyframe. If the object number changes, the current frame is declared as a new keyframe, otherwise, their region-based distance will be compared with a preset threshold, and if it exceeds the threshold, it will still be selected as a new keyframe. Interesting results have been reported; however, this object-based summarization scheme may only work well for videos which have relatively simple content and contain a small number of video objects, such as the surveillance videos. More complex video sources will reduce its effectiveness.

4. Conclusion

Video abstraction is practically an inseparable research area for many video applications including video indexing, browsing and retrieval. A concisely and intelligently generated video abstraction will not only enable a more informative interaction between human and computer during the video browsing, but also help to build more meaningful and quicker video indexing and retrieval systems. Recently, video abstraction has been attracting considerable research interest, and it is gradually coming to play an important role in the multimedia database area.

5. Major Players

Major players in video abstraction field are:

1. **Companies:**
    - Fuji Xerox Research Lab (FX Palo Alto Lab), mainly focus on the weekly staff meeting video
    - Sharp Labs of America
    - Intel Corporation, generic and home videos
    - Compaq Computer Corp. Cambridge Research Lab
    - IBM Almaden Research Center
    - C & C Research Lab, NEC USA
    - AT&T BELL Lab, Machine Perception Research Department
    - Microsoft Research, mainly focus on information talks (presentations), and Microsoft Research, China
    - IBM Watson Research Center
    - Siemens Corporate Research Lab, Multimedia & Video Technology Department
    - Hewlett-Packard Research Lab
    - NTT Human Interface laboratory, Japan

2. **Universities:**
    - University of Mannheim, Germany.
    - Carnegie Mellon University, Department of Electrical and Computer Engineering
    - University of Minnesota, Department of Electrical and Computer Engineering
    - Delft University of Technology, the Netherlands
    - University of Illinois at Urbana-Champaign
    - Princeton University
    - MIT Media Lab
    - National Technical University of Athens, Department of Electrical and Computer Engineering
    - University of Maine, Department of Spatial Information Engineering
    - National University of Singapore, School of Computing
    - University of Maryland, Language and Media Processing Lab
    - University of Washington, Department of Electrical Engineering

6. Reference

[1] A. Elmagarmid et al., “Video database systems”, *Kluwer Academic Publishing*, Boston, 1997.

[2] H. D. Wactlar, M. G. Christel, Y. Gong and A. G. Hauptmann, “Lessons learned from building a Terabyte digital video library”, *IEEE Computer*, pp. 66-73, Feb. 1999.

[3] S. Pfeiffer, R. Lienhart, S. Fischer and W. Effelsberg, “Abstracting digital movies automatically”, *Journal of Visual Communication and Image Representation*, vol. 7, no. 4, pp. 345-353, Dec. 1996.

[4] A. Hanjalic and H. J. Zhang, “An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, Dec. 1999.

[5] M. Smith and T. Kanade, “Video skimming for quick browsing based on audio and image characterization”, *Technical Report CMU-CS-95-186*, School of Computer Science, Carnegie Mellon University, July 1995.

[6] A. G. Hauptmann and M. A. Smith, “Text, speech, and vision for video segmentation: The Informedia Project”, *Proc. of the AAAI Fall Symposium on Computer Models for Integrating Language and Vision*, 1995.

[7] M. A. Smith and T. Kanade, “Video skimming and characterization through the combination of image and language understanding techniques”, *Proc. of the IEEE Computer Vision and Pattern Recognition*, pp. 775-781, 1997.

[8] M. G. Christel, M. A. Smith, C. R. Taylor and D. B. Winkler, “Evolving video skims into useful multimedia abstractions”, *Proc. of Conference on Human Factors in Computing Systems (CHI98)*, pp. 171-178, April 1998.

[9] C. Toklu, A. P. liou and M. Das, “Videoabstract: A hybrid approach to generate semantically meaningful video summaries”, *ICME2000*, New York, 2000.

[10] J. Nam and A. H. Tewfik, “Video abstract of video”, *IEEE Third Workshop on Multimedia Signal Processing*, pp. 117-122, Sep. 1999.

[11] N. Omoigui, L. He, A. Gupta, J. Grudin and E. Sanocki, “Time-compression: System concerns, usage, and benefits”, *Proc. of ACM Conference on Computer Human Interaction*, 1999.

[12] A. Amir, D. Ponceleon, B. Blanchard, D. Petkovic, S. Srinivasan and G. Cohen, “Using audio time scale modification for video browsing”, *Proc. of the 33rd Hawaii International Conference on System Sciences*, vol. 1, Jan. 2000.

[13] G. W. Heiman, R. J. Leo, G. Leighbody and K. Bowler, “Word intelligibility decrements and the comprehension of time-compressed speech”, *Perception and Psychophysics*, vol. 40, no. 6, pp. 407-411, 1986.

[14] D. M. Russell, “A design pattern-based video summarization technique: moving from low-level signals to high-level structure”, *Proc. of the 33rd Hawaii International Conference on System Sciences*, vol. 1, Jan. 2000.

[15] L. He, E. Sanocki, A. Gupta and J. Grudin, “Audo-summarization of audio-video presentations”, *Proc. of ACM Multimedia*, pp. 489-498, 1999.

[16] R. Lienhart, “Dynamic video summarization of home video”, *Proc. of IS&T/SPIE*, vol. 3972, pp. 378-389, Jan. 2000.

[17] R. Lienhart, S. Pfeiffer and W. Effelsberg, “Video abstracting”, *Communications of the ACM*, pp. 55-62, Dec. 1997.

[18] M. Mills, “A magnifier tool for video data”, *Proc. of ACM Human Computer Interface*, pp. 93-98, May 1992.

[19] Y. Taniguchi, “An intuitive and efficient access interface to real-time incoming video based on automatic indexing”, *Proc. of ACM Multimedia*, pp. 25-33, Nov. 1995.

[20] K. Otsuji, Y. Tonomura and Y. ohba, “Video browsing using brightness data”, *Proc. of SPIE*, vol. 1606, pp. 980-985, 1991.

[21] B. Shahraray and D. C. Gibbon, “Automatic generation of pictorial transcriptions of video programs”, *Proc. of IS&T/SPIE*, vol. 2417, pp. 512-519, San Jose, CA, 1995.

[22] P. England, R. B. Allen, M. Sullivan and A. heybey, “I/Browse: The bellcore video library toolkit”, *Proc. of SPIE*, vol. 2670, pp. 254-264, Feb. 1996.

[23] H. Ueda, T. Miyatake, S. Sumino and A. Nagasaka, “Automatic structure visualization for video editing”, *Proc. of INTERCHI’93*, pp. 137-141, 1993.

[24] S. W. Smoliar and H. J. Zhang, “Content-based video indexing and retrieval”, *IEEE Multimedia*, pp. 62-72, 1994.

[25] F. Arman, R. Depommier, A. Hsu and M. Y. Chiu, “Content-based browsing of video sequences”, *ACM Multimedia’94*, pp. 97-103, Aug. 1994.

[26] B. Falchuk and K. Karmouch, “A multimedia news delivery system over an ATM network”, *International Conference on Multimedia Computing and Systems*, pp. 56-63, 1995.

[27] H. J. Zhang, C. Y. Low and S. W. Smoliar, “Video parsing and browsing using compressed data”, *Multimedia Tools and Applications*, vol. 1, pp. 89-111, 1995.

[28] H. J. Zhang, J. Wu, D. Zhong and S. W. Smoliar, “An integrated system for content-based video retrieval and browsing”, *pattern Recognition*, vol. 30, no. 4, pp. 643-658, 1997.

[29] M. M. Yeung and B. Liu, “Efficient matching and clustering of video shots”, *Proc. of IEEE ICIP’95*, vol. I, pp. 338-341, 1995.

[30] Y. Zhuang, Y. Rui, T. S. Huang and S. Mehrotra, “Adaptive key frame extraction using unsupervised clustering”, *ICIP’98*, 1998.

[31] W. Wolf, “Key frame selection by motion analysis”, *ICASSP’96*, vol. 2, pp. 1228-1231, 1996.

[32] A. M. Ferman and A. M. Tekalp, “Multiscale content extraction and representation for video indexing”, *Proc. of SPIE*, vol. 3229, pp. 23-31, 1997.

[33] A. Girgensohn and J. Boreczky, “Time-constrained keyframe selection technique”, *Proc. of ICMCS’99*, pp. 756-761, 1999.

[34] R. L. Lagendijk, A. Hanjalic, M. Ceccarelli, M. Soletic and E. Persoon, “Visual search in a smash system”, *Proc. of ICIP’96*, pp. 671-674, Sep. 1996.

[35] B. Shahraray, “Scene change detection and content-based sampling of video sequences”, *Proc. of SPIE*, vol. 2419, pp. 2-13, Feb. 1995.

[36] S. X. Ju, M. J. Black, S. Minneman and D. Kimber, “Summarization of video-taped presentations: automatic analysis of motion and gestures”, *IEEE Transactions on CSVT*, 1998.

[37] C. Toklu and S. P. Liou, “Automatic keyframe selection for content-based video indexing and access”, *Proc. of SPIE*, vol. 3972, pp. 554-563, 2000.

[38] Multimedia Description Schemes group, “Text of 15938-5 FCD Information Technology – Multimedia Content Description Interface – Part 5 Multimedia Description Schemes”, *ISO/IEC JTC 1/SC29/WG11/N3966*, Singapore, March 2001.

[39] M. Massey and W. Bender, “Salient stills: process and practice”, *IBM Systems Journal*, vol. 35, 1996.

[40] M. Lee, W. Chen, C. Lin, C. Gu, T. Markoc, S. Zabinsky and R. Szeliski, “A layered video object coding system using sprite and affine motion model”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, Feb. 1997.

[41] J. Wang and E. Adelson, “Representing moving images with layers”, *IEEE Transactions on Image Processing*, vol. 3, Sep. 1994.

[42] N. Vasconcelos, A. Lippman, “A spatiotemporal motion model for video summarization”, *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1998.

[43] M. Iran and P. Anandan, “Video indexing based on mosaic representation”, *Proceedings of The IEEE*, vol. 86, no. 5, May 1998.

[44] Y. Taniguchi, A. Akutsu and Y. Tonomura, “PanoramaExcerpts: extracting and packing panoramas for video browsing”, *Proc. of the 5th ACM International Multimedia Conference*, pp. 427-436, Nov. 1997.

[45] A. D. Doulamis, N. Doulamis and S. Kollias, “Non-sequential video content representation using temporal variation of feature vectors”, *IEEE Transactions on Consumer Electronics*, vol. 46, no. 3, August 2000.

[46] A. Stefanidis, P. Partsinevelos, P. Agouris and P. Doucette, “Summarizing video datasets in the spatiotemporal domain”, *Proc. of 11th International Workshop on Database and Expert Systems Applications*, pp. 906-912, Sep. 2000.

[47] S. Uchihashi, J. Foote, A. Girgensohn and J. Boreczky, “Video manga: generating semantically meaningful video summaries”, *ACM Multimedia’99*, 1999.

[48] M. M. Yeung and B. L. Yeo, “Video visualization for compact presentation and fast browsing of pictorial content”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 5, Oct. 1997.

[49] X. D. Sun and M. S. Kankanhalli, “Video summarization using R-sequences”, *Real-time Imaging*, pp. 449-459, Dec. 2000.

[50] R. L. Lagendijk, A. Hanjalic, M. Ceccarelli, M. Soletic and E. Persoon, “Visual search in a SMASH system”, *Proc. of ICIP’97*, pp. 671-674, 1997.

[51] A. Hanjalic, M. Ceccarelli, R. L. Lagendijk, and J. Biemond, “Automation of systems enabling search on stored video data”, *Proc. of SPIE*, vol. 3022, pp. 427-438, 1997.

[52] K. Ratakonda, M. I. Sezan and R. Crinon, “Hierarchical video summarization”, *Proc. of SPIE*, vol. 3653, pp. 1531-1541, Jan. 1999.

[53] A. D. Doulamis, N. D. Doulamis and S. D. Kollias, “A fuzzy video content representation for video summarization and content-based retrieval”, *Signal Processing*, vol. 80, no. 6, June 2000.

[54] S. Auephanwiriyakul, A. Jushi and R. Krishnapuram, “Fuzzy shot clustering to support networked video databases”, *IEEE International Conference on Fuzzy Systems Proceedings*, pp. 1338-1343, 1998.

[55] D. Dementhon, V. Kobla and D. Doermann, “Video summarization by curve simplification”, *ACM Multimedia 98*, pp. 211-218, 1998.

[56] Y. Gong and X. Liu, “Generating optimal video summaries”, *ICME2000*, New York, 2000.

[57] E. Sahouria and A. Zakhor, “Content analysis of video using principal components”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, Dec. 1999.

[58] F. Dufaux, “Key frame selection to represent a video”, ICME2000.

[59] P. Campisi, A. Longari and A. Neri, “Automatic key frame selection using a wavelet based approach”, *Proc. of SPIE*, vol. 3813, pp. 861-872, July 1999.

[60] C. Kim and J. N. Hwang, “An integrated scheme for object-based video abstraction”, *ACM Multimedia 2000*, Los Angeles, CA, 2000.