BROADCAST VIDEO PROGRAM SUMMARIZATION USING FACE TRACKS

Kadir A. Peker[1], Isao Otsuka[2], Ajay Divakaran[1]

[1]Mitsubishi Electric Research Laboratory, Cambridge, MA
[2]Mitsubishi Electric Corporation, Advanced Technology R&D Center

ABSTRACT

We present a novel video summarization and skimming technique using face detection on broadcast video programs. We take the faces in video as our primary target as they constitute the focus of most consumer video programs. We detect face tracks in video and define face-scene fragments based on start and end of face tracks. We define a fast-forward skimming method using frames selected from fragments, thus covering all the faces and their interactions in the video program. We also define novel constraints for a smooth and visually representative summary, and construct longer but smoother summaries.

1. INTRODUCTION

Personal video recorders (PVR) enable digital recording of several days' worth of broadcast video on a hard disk device. Several user and market studies confirm that this technology has the potential to profoundly change the TV viewing habits. Effective browsing and summarization technologies are deemed crucial to realize the full potential of these systems.

Video summarization can be used for a number of different purposes or tasks, in a wide variety of applications on various platforms. The ideal case for application integration is for the summarization technology to be perfectly aligned and seamlessly integrated with the application such that the user may not even know that video summarization technology is used in the background. In this regard, video summarization technology can manifest in many different forms depending on the application environment it is used in.

Quite recently, domain specific content-segmentation such as news video story segmentation [1], or home video summarization [2] has been studied and has produced impressive results.

Our focus in this work is driven by the constraints of the consumer electronics platforms, and the requirements and flexibilities of TV viewing application. A number of application characteristics define our emphasis points:

1. Flexibility of user interaction in TV viewing: A responsive algorithm and an effective user interface can accommodate for less stringent accuracy requirements.
2. Limited processing power of the consumer electronics platforms: The application is run on a consumer electronics platform, as opposed to a general purpose PC. The resources are much more limited. The types of video are very varied as well, and not specialized. The most return on limited investment is desired.

We have developed audio classification based summarization solutions for sports video in our past work [3]. We used face detection in mostly static scenes such as news, for video browsing and summarization in [4]. In this work, we use face tracks to extend our work beyond static face scenes.

The platform limitations and the generality of broadcast video on PVRs (personal video recorders) suggest using features that provide maximal application range with the minimum cost. We find faces as the most important visual class that will enable analysis of a wide array of video types, as the humans are mostly the primary subject of video programs. We use the Viola-Jones face detector, which provides high accuracy and high speed [5]. It can also easily accommodate detection of other objects by changing the parameter file used. Thus, the same detection engine can be used to detect several classes of objects. The parameter files on the consumer device can even be updated remotely.

We focus on the typical broadcast program, which usually range from 30 minutes to an hour. A summarization technique that can summarize an hour program in a reasonable time would generate a similarly reasonable length summary for a 30 minute program also. Hence, we take 1 hour as our guide for judging summary lengths and content coverage.

We envision two ways of skimming of video programs: fast-forwarding, and, playback of segments (video summary) from the whole video. Video analysis and summarization techniques are used for selecting the frames for fast-

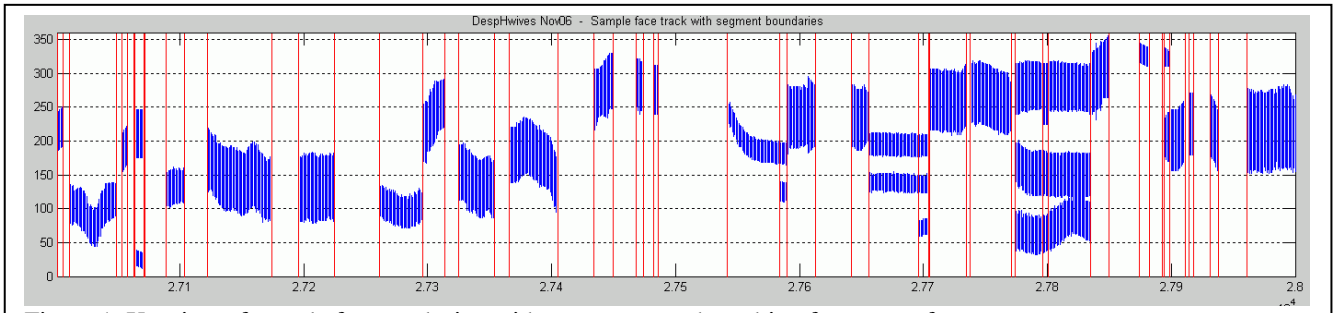

Figure 1. X-t view of sample face tracks in a video segment, and resulting face-scene fragments.

forwarding at an average speed of 20x. A hierarchical key-frame based browsing of video can also be added as a third method. We assume a consumer electronics platform such as a TV or a PVR, but the techniques can be easily extended to other platforms.

The purpose of skimming through video can be to review an already viewed program (decide to keep or delete, remember what it was about, locate a certain segment, etc), or a new program (decide to watch or not, find out what it is about, go to desired segments, etc). With the availability of large volumes of PVR storage, we believe users will have the luxury of recording liberally and deciding later, when they have the content along with a set of skimming tools, rather than a few lines of descriptive text only.

We take people, or faces, to be the focus of our interest noting that humans are the primary subjects of most consumer video programs.

2. FACE TRACKS IN VIDEO

We use the Viola-Jones face detector which is based on boosted rectangular image features [1]. We first sub-sample the video stream to 360x240, and perform the detection at 1 pixel shifts. The speed is about 15 fps at these settings on a Pentium IV 3.0 GHz PC, including decoding and display overheads. Using DC images in MPEG encoded streams increases the speed dramatically, both through the detector (speed is proportional to the number of pixels), and through savings in decoding. The minimum detected face size increases in this case, but the targeted significant faces are mostly within range. The detector can be run on only the I-frames, or at a temporally sub-sampled rate to further improve processing cost.

About 1 false detection per 30-60 frames occurs with the frontal face detector. Missed faces are more common because of occlusions, highly non-frontal faces, etc. Face tracks are formed by matching each new face to the existing "live" tracks. A track is declared "dead" if less than 3 faces in the last 10 frames are detected for that track. Currently, matching of faces is performed through computing the ratio of the area of the overlapped region to both of the faces, and deciding according to a threshold (at least %50 of both). A new face that doesn't match any current live track starts a new track. Gaps in tracks are filled through interpolation of x, y, and size values of faces. Tracks that are shorter than 3 frames are eliminated as false alarms.

We declare a "fragment boundary" in the video when a new track starts or a track ends. Thus, fragments are temporal units of video that contain the same tracks (zero or more) throughout. Each track, in turn, spans one or more video fragments. Figure 1 illustrates the face tracks and video fragments. The x-axis corresponds to time (frames) and the y-axis corresponds to the width of the video frame. Thus, it can be thought of as a 'top view' of the video after face detection. Whenever a track starts or ends, a fragment boundary is declared, and marked by vertical lines. Some fragments are caused by a few frames of misses within a longer face track, resulting in split tracks with gaps in between.

3. FAST-FORWARD BASED VIDEO SKIMMING

3.1. Duration and Visual Coverage

The most straightforward way to browse through a recorded video program is by fast forwarding. A user who wants to get a rough idea of what is there in a given 1 hour program can, at a fast forward rate of 20x, go through a 60 min program in 3 minutes. We determine this to be a reasonable length for a user to sit through and get a good idea of the contents of a program. At 20x speed, assuming speedup is achieved by temporal sub-sampling, more than 1 frame per second is taken from the video. In terms of visual coverage of the contents of the program, this should be sufficient in almost all cases, since a very critical visual scene being missed between two key-frames is very unlikely. (We loosely call the frames selected for fast-forward playback as key-frames. Later, we will use non-uniform sub-sampling to select these frames).

There are two parameters for fast-forward playback: the sub-sampling rate, and the playback rate. Only small speed up rates (2-4x) can be achieved through playing all the frames but at a higher rate, because of processing limitations. At higher rates, sub-sampling is used. The actual playback rates in that case are usually lower than 30 frames/sec, because of the seeking overhead. If we assume a 1 frame / sec sub-sampling, then the summary (or the super fast-forward) is about 2-4 minutes long.

We view the set of 3600 frames (at 1 frame/second sub-sampling) as a sufficient visual coverage of the 60 min content, as noted before. Furthermore, we view the 2-4 min playback time as a reasonable length for a quick view (summary) of a 60 min program, also noted before. This 1 frame / sec uniform sub-sampling of the video is our baseline summary that satisfies visual content coverage and length criteria for a quick view of a 1 hour video program. This quick view is useful in many places as indicated by its wide use in PVRs etc. already. For example, it is an effective way of reviewing content, e.g. those recorded automatically by the PVR based on inferred user preference.

3.2. Fast-Forward Skimming Using Non-uniform Sub-sampling Based on Face Fragments

One aspect we want to improve on the uniform sub-sampling is to further assure that significant key points are captured visually. This is the basis for key-frame selection approaches; however, we describe the problem in the context of super fast-forwarding (at around 20x speed), rather than story-boarding. Problem parameters such as average frame rate (equivalently, number of key-frames) are determined accordingly; or new constraints such as minimum frame rate, maximum deviation from uniform sampling, etc are introduced.

We select a key-frame from each fragment, as defined in section 2. Thus, we have representation of all face compositions (each face track, and every combined or single appearance of those faces). Each different fragment constitutes a different face scene composition.

We treat no-face fragments differently in two respects:

1. No-face fragments shorter than 15 frames are eliminated. In our experiments, such fragments can constitute up to half of all no-face segments, and about 15% of all fragments. Hence the filtering results in considerable reduction in number of key-frames. These are usually misses that occur in the middle of a face track. Note that detection of a face track, even if it is of short duration, is significant as it captures an object (a face) that we want to capture visually. But there is no such need for no-face fragments.
2. No-face fragments that are longer than 300 frames are sub-sampled at 1/150 frames. Face fragments indicate constancy of an observed quality. No-face fragments, however, are not necessarily uniform in content. Thus, we want to ensure we don't miss anything significant, by taking a frame at least every 5-10 seconds. This in fact introduces a very small increase in the number of key-frames (less than 3% in our experiments)

3.3. Smoothness and Coverage Constraints

At the second step, we want to construct a video skim that is less jumpy then a sequence of discrete key-frames, but has the same visual coverage with the minimal number of frames possible. Given the set of fragments Fi;

1. The summary should include at least k frames from each fragment,
2. The summary should constitute of segments, each of which are contiguous set of frames and at least of length M.

To simplify this optimization problem, we set k=M/2. In this case a trivial, near-optimum solution is to select the last k frames from a fragment and the first k frames from the next fragment, thus forming a 2k=M frames segment. Then we continue in the same way with the following fragments. If a fragment is shorter than k frames, then, a) if k frames selected from the previous fragment are adjacent (i.e. selected from the end of the fragment), then we select the first k frames from the next segment, b) if the current segment starts with the current (short) fragment, then we select more than k frames from the next fragment, to make up to 2k=M.

3.4. Application

We generated 3 minute skims of an episode of Desperate Housewives (60 minutes). We detected about 60000 faces in 8006 tracks. After filtering using length and score criteria, 2166 tracks were left. There were 3940 fragments, 1236 were no-face fragments. 586 of no-face fragments were eliminated due to length, leaving about 3400 fragments. We selected 2 frames from each fragment (to simulate 15 fps on-the-fly playback on a consumer device) and constructed a 3 min 47 sec skim. Subjectively, the skim had a better coverage of people and interactions in the episode, compared to a skim generated with uniform sampling. One useful side effect we observed was that, the proportion of the commercials in the skim was less than in the original program, since commercials contained less faces.

In the second step, we generated a smoother but longer (9.5 minutes) summary by using the method of section 4.3, with k=5 and M=10. Note that the length is less than 5 times of

the 1-frame-per-fragment summary, because many fragments are shorter than 5 frames. This longer summary was much smoother.

We believe both types of skims (3 min and 10 min) have their application in the consumer electronics application we envision. A 3 min summary visually covers a whole program in a very short time. This can be useful to review programs that are already seen, or completely new programs such as documentaries etc. to see what type of a program it is and what kind of themes are included. The longer summary is useful when the user wants to watch a portion of the video with better understanding than the 3 min skim. For example, the user may watch the first few minutes of the 10 min skim (which will cover first 10-20 minutes of the original program), and decide whether s/he wants to watch/keep the whole episode or not.

4. CONCLUSIONS

We have presented a video skimming method based on face detection and can be used for different tasks and goals on a consumer electronics platform for watching broadcast video programs. We presented methods for segmenting the video using face track information, and criteria for smoothness and visual coverage of the skim.

In the future, we want to combine the visual and audio analysis to generate a more 'pleasant' summary. For instance, we have observed that the alignment of summary boundaries with the speech or sentence boundaries in the video affects the perceived quality of the video summary largely.

5. REFERENCES

[1] T.S. Chua, S.F. Chang, L. Chaisorn, W. Hsu, "Story Boundary Detection in Large Broadcast Video Archives – Techniques, Experience and Trends," ACM Multimedia Conference, 2004.

[2] Chong-Wah Ngo, Yu-Fei Ma, Hong-Jiang Zhang. "Automatic Video Summarization by Graph Modeling,", Ninth IEEE International Conference on Computer Vision (ICCV'03) - Volume 1, 2003.

[3] Divakaran, A.; Peker, K.A.; Radharkishnan, R.; Xiong, Z.; Cabasson, R., "Video Summarization Using MPEG-7 Motion Activity and Audio Descriptors", Video Mining, Rosenfeld, A.; Doermann, D.; DeMenthon, D., October 2003 Kluwer Academic Publishers.

[4] K. A. Peker, A. Divakaran, T. Lanning, "Browsing news and talk video on a consumer electronics platform using face detection," Proc. Of SPIE 6015 Multimedia Systems and Applications VIII, Oct 2005, Boston.

[5] Viola P., Jones M., "Robust real-time object detection", IEEE Workshop on Statistical and Computational Theories of Vision, 2001.