AVE - Automated Home Video Editing

Xian-Sheng HUA, Lie LU, Hong-Jiang ZHANG
Microsoft Research Asia
3F, Beijing Sigma Center, No.49 Zhichun Road, Beijing 100080, P.R.China
{ xshua; llu; hjzhang }@microsoft.com

ABSTRACT
In this paper, we present a system that automates home video editing. This system automatically extracts a set of highlight segments from a set of raw home videos and aligns them with user supplied incidental music based on the content of the video and incidental music. We developed an approach for extracting temporal structure and determining the importance of a video segment in order to facilitate the selection of highlight segments. Additionally we extract temporal structure, beats and tempos from the incidental music. In order to create more professional-looking results, the selected highlight segments satisfy a set of editing rules and are matched to the content of the incidental music. This task is formulated as a non-linear 0-1 programming problem and the rules are embedded as constraints. The output video is rendered by connecting the selected highlight video segments with transition effects and the incidental music.

Categories and Subject Descriptors
H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems — video; I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—video analysis.

General Terms
Algorithms, Experimentation.

Keywords
Video editing, video content analysis, video segmentation, audio segmentation, music analysis, video skimming, optimization.

1. INTRODUCTION
While camcorders have become a commodity home appliance, few watch the recorded videos or share them with friends and relatives. In contrast with sharing photographs and the stories behind them, watching a home video is often seen as a chore. Though many camcorders are becoming digital, the popularity of home videos has not changed. The key reasons behind this are low content quality of the recorded video and the difficulty of turning raw recorded video into a compelling video story. Existing video editing systems, such as *Abode Premiere*, are a great help for editing video, but the task is still a tedious and time consuming requiring significant editing skills and an aesthetic sense. In this paper, we present a system that automates home video editing, creating near-professional results using a set of video and music analysis algorithms.

MM'03, November 2-8, 2003, Berkeley, California, USA.

By automated video editing (AVE), we refer to a process which automatically selects suitable or desirable segments from an original video source and aligns them with a given piece of incidental music to create an edited video segment to a desired length. To ensure that the edited video is of satisfactory quality, two sets of rules derived from studying professional video editing are followed. The first deals with how to select suitable segments that are representative of the original video in content and of high visual quality. The other rule deals with how to align selected video segments with a chosen piece of incidental music to increase the impact of the edited video.

Generally watching a long unedited video requires a great deal of patience and time. An effective way to attract a viewer is to present a video that is as compact as possible, yet preserves the most critical features required to tell a story, relate an expression or chronicle an event. In other words, the editing process should select segments with greater relative "importance" or "excitement" value from the raw video. A formal definition of importance, however, is hard to make as it is a subjective concept. It is also difficult to quantify an importance measure, even though some qualitative importance measures can be obtained based on video editing rules. Furthermore, for a given video, the most "important" segments according to an importance measure could concentrate in one or in a few parts of the time line of the original video. This may obscure the storyline in the edited video. In other words, the distribution of the selected highlight video should be as uniform along the time line as possible so as to preserve the original storyline.

The second set of rules is related to the incidental music. To make the edited video more expressive and attractive, we try to have shot transitions occur exactly at music beats. We also try to match the motion intensities of selected video segments with the tempos of the corresponding music clips. Furthermore, if there is speech in the selected segment, it is better to keep the sentences whole and understandable in the output video. Accordingly, the volume of the music is turned down to make the utterance audible. To do all this requires audio and music analysis, such as beat tracking, tempo estimation and sentence detection. Additionally, the audio side of the problem must be taken into consideration when choosing video segments from the raw video.

1.1 Related Work
A research problem closely related to AVE is video summarization. Numerous contributions to this topic have been reported. One of the most straightforward approaches is to compress the original video by speeding up the playback[12]. However, the abstract factor in this approach is limited by the playback speed in order to keep the speech comprehensible. The InforMedia system [15] generates short synopsis of video by

integrating audio, video and textual information. By combining language understanding techniques with visual feature analysis, this system produces reasonable results. However, satisfactory results may not be achievable for home videos by such a text-driven approach since the speech signals in home videos are often quite noisy. Another approach to generating semantically meaningful summaries is event-oriented abstraction scheme, such as that presented in [5]. Recently, more sophisticated techniques have also been proposed. For example, the trajectories of moving objects were used in [16]. The linear dynamical system theory is applied in [13]. In [2], the authors use singular value decomposition to summarize video content.

Generally summarization requires semantic understanding of the video content. In [10], Y.F. Ma attempts to generate summaries by detecting "attractiveness" of or the possibility that the viewer may pay "attention" to each video frame, instead of understanding the semantic content. The automatic video editing system proposed in this paper uses this approach to select "important" video segments, as will be introduced in Section 3 along with some improvements required for automatic home video editing.

Although manually adding music to video is a common practice in the movie and video production, it is a difficult thing to ask a casual user to do. Automating this process to produce reasonable results, however, is a difficult task. One approach to this problem was reported in [1]. In that system, content selection is based on calculating video unsuitability, which is related only to camera motion and image contrast. In addition, video segments are merged together along the music timeline without taking motion-tempo matching into consideration. Our proposed system takes more sophisticated content features into account, such as *attention detection*, sentence detection, motion-tempo matching, content-based rendering, etc. Furthermore, it is an extendable framework and therefore flexible enough to add other features into the system. However, two of the three assumptions in [1] for creating music videos are also taken into account in our AVE system. The first is that *improved soundtrack quality improves perceived video image quality*. The other is that *synchronizing video and audio segments enhance the perception of both*.

As mentioned in [1], a commercial venture, *muvee.com*, offers an automatic system for producing music videos. Though no details of the algorithm are available, it is likely that editing is accomplished by a rule-based approach. Unlike *Muvee*, our system is an optimization-based system, in which an optimal set of video segments are extracted from the original video to produce the music video under certain adjustable and increasable constraints, thus it is easy to upgrade and refine. H. Sundaram has also proposed an optimization-based utility framework for automatic generation of audio-visual skims [17]. However, this framework is based on an assumption that the data is not raw stream (e.g., home video), but is the result of an editing process (e.g., film, news), as editing grammar or film syntax is one of the bases of the film reduction schemes proposed in [17].

1.2 System Overview

Our automated home video editing system has three stages, as illustrated in Figure 1. The first stage is content analysis, consisting of video temporal structure parsing, attention detection, sentence detection in the audio track of the original video, and beat/tempo detection in the music. The second stage is content selection (including boundary alignment), which selects a particular set of "important" and informative video segments that match motion with tempo, as well as shot boundaries with sentences in the audio track and music beats. The total length of the selected video segments may be determined either by the duration of the incidental music, which is what we assumed in this paper, or another desired value. This central stage is the primary and the most challenging one. The last stage is composition, which renders selected video segments with music by adding appropriate transitions between the selected video segments.

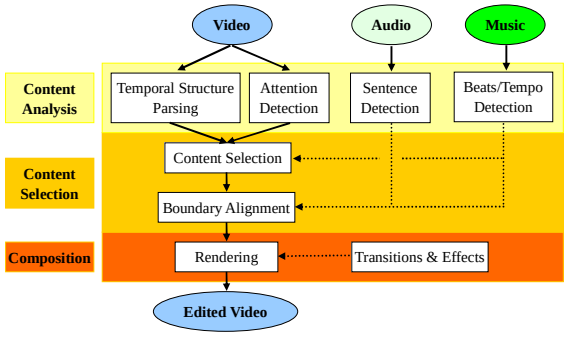

Figure 1. System overview.

Due to limited shooting skills, shots in original home videos are often long when compared with most professionally edited video programs. These original shots often contain redundant information as well as boring sequences and low quality frames. Consequently, our system segments video sequences during the temporal parsing process, and the "importance" index is computed for each sub-shots, as described in detail in Sections 2 and 3. Only the "best" sub-shots are used for the construction of the final video.

The rest of the paper is organized as follows. After presenting the problem formulation briefly in Section 2, video content analysis and music analysis are introduced in Section 3. Section 4 describes how to automate video editing, followed by experimental results in Section 5 and a number of ideas for future extensions of the system in Section 6. Conclusion and future work are presented in Section 7.

2. PROBLEM FORMULATION

There are three "input" data sequences, namely, music, audio and video, in our editing system, as illustrated in Figure 2. The objective is to excerpt particular segments from the video sequence that satisfy the requirements mentioned in Section 1. Our strategy parses the video sequence into hierarchical structures consisting of scenes, shots, and sub-shots. For music, we segment it into clips (we call them music sub-clips) by strong beats, and for each clip, tempo is estimated, which indicates the speed of the music sub-clips.

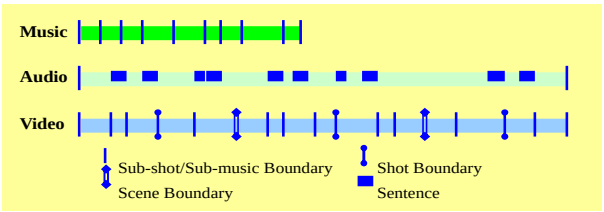

Figure 2. Three input data sequences.

Consequently, the problem is formulated as selecting a particular set of sub-shots from the video for the music sub-clips while satisfying the requirements (two sets of rules) presented in Section 1. To more clearly describe our algorithms, we define a series of symbols that will be employed in this paper.

A video *v* consists of a series of scenes, denoted by

$$Scene = \{Scene_i, 0 \le i < K^{(SC)}\} \quad (1)$$

Similarly, the video can also be represented as a series of shots and sub-shots, namely,

$$Shot = \{Shot_i, 0 \le i < K^{(SH)}\} \quad (2)$$

$$SubShot = \{SubShot_i, 0 \le i < K^{(SS)}\} \quad (3)$$

where $K^{(SC)}$, $K^{(SH)}$ and $K^{(SS)}$ are the total number of scenes, shots and sub-shots in the video, respectively. For simplicity, we often substitute N for $K^{(SS)}$, as is a very common term.

For a sub-shot, several features are extracted to represent the content and temporal location of the sub-shot, including *Importance (or Attention) Index*, *Motion Intensity*, and the *Scene/Shot ID* to which it belongs. All these features are denoted as follows:

$$Impt = \{impt_i, 0 \le i < N\} \quad (4)$$

$$Motion = \{motion_i, 0 \le i < N\} \quad (5)$$

$$SC = \{sc_i, 0 \le i < N\}, 0 \le sc_i < K^{(SC)} \quad (6)$$

$$SH = \{sh_i, 0 \le i < N\}, 0 \le sh_i < K^{(SH)} \quad (7)$$

The music sub-clip set for music *m* is denoted by

$$SubMusic = \{SubMusic_i, 0 \le i < M\} \quad (8)$$

where M indicates the total number of music sub-clips. The corresponding tempo of each sub-music clip is denoted by

$$Tempo = \{tempo_i, 0 \le i < M\} \quad (9)$$

The strength of the beat at the right boundary of each music sub-clip (except the last one) is indicated as

$$Beat = \{beat_i, 0 \le i < M - 1\} \quad (10)$$

Therefore, the problem of automated home video editing can be describe as, to select M elements from the N-element set *SubShot*, which satisfy the two sets of rules mentioned in Section 1, then to output a video by connecting sub-shots (shots to be exact, as viewed from the output video), with specific shot transitions, as well as alignment with the incidental music.

The most significant step is content (sub-shot) selection. In this system, content selection is formulated as an optimization problem, which attempts to find an optimal set of segments while satisfying all the requirements. Let θ denote an M-element subset of {0, 1, 2, ⋯, N-1}, and Θ the set of all subsets of this form. Sub-shots whose subscripts are in θ are the selected sub-shots. Then the content selection problem can be rewritten as,

$$\begin{aligned} &Maximize/Minmize \;\; ObjectiveFuction\ (m, v, \theta) \\ &where\ \theta \in \Theta \end{aligned} \quad (11)$$

Accordingly, the content selection problem is converted into the design of the objective function and solving the optimization problem, detailed in Section 4. Prior to that, we detail video and music content analysis in the next section.

It should be mentioned here that although the duration of the output video is set equal to that of the user provided music in the above formulation, we may set it to any desired value. In this case, if the duration of the music is longer than the user provided duration, we fade out the music when the output video ends, while on the contrary, if the music duration is shorter than desired duration, the system will ask user to add another piece of music, or just repeat the music from the beginning and then fade out when the output video reaches its end.

3. VIDEO AND MUSIC ANALYSIS

Video content analysis consists of three components: temporal structure parsing, attention detection and sentence detection in the audio track of the video.

3.1 Temporal Structure Parsing

The first step of structure parsing, i.e., shot boundary detection, is performed using the algorithms proposed in [20]. For raw home videos, most of the shot boundaries are simple cuts, which are much easier to detect correctly in comparison with professionally edited videos. Once a transition is detected, video temporal structure is further analyzed using two approaches, described next.

One approach divides the shots into smaller segments, namely, sub-shots, whose lengths are in a certain range (defined in Section 4). This is accomplished by detecting the maximum of the frame difference curve (FDC). A shot is cut into two sub-shots at the local maximum, if the local maximum's distance from the two shot boundaries are both not less than the minimal length of a sub-shot. Then the above process is repeated until the lengths of all sub-shots are smaller than the maximal sub-shot length.

The other approach is to merge shots into groups of shots, i.e., scenes. There are many scene grouping methods presented in the literature [6][8]. In this paper, a hierarchical method that merges the most similar adjacent shots/scenes step-by-step into bigger ones is employed. The similarity measure is the intersection of averaged and quantized color histogram in HSV space [8]. The stop condition can be determined either by similarity threshold or the final scene numbers.

3.2 Attention Detection

As previously mentioned, most video summarization approaches require semantic understanding of the video content. Unfortunately, current computer vision and artificial intelligence technologies cannot accomplish it for unstructured home videos. However, if the objective is creating a compelling video, it may not be necessary to understand the semantic content completely. Alternatively, we need only determine those parts of the video more "important" or "attractive" than the others. Assuming that the most "important" video segments are those most likely to hold a viewer's interest, the task becomes how to find and model the elements that are most likely to attract a viewer's attention. This is the main idea of the work proposed by Ma et al. [10]. In our system, video segment selection is also based on this idea, but we refine the method by adding an "attention fusion" function, which generates improved results.

Attention is a neurobiological concept. Computational attention allows us to break down the problem of understanding a live video sequence into a series of computationally less demanding and localized visual, audio, and linguistic analytical problems. In [10], video summaries are based on modeling how a viewer's attention is attracted by object motion, camera motion, specific objects (such as faces), static attention regions, audio and language when viewing a video program. That system adopted a linear combination to implement the fusion scheme due to its effectiveness and simplicity. With such a scheme, each *attention component* is normalized to [0~1]. Let A denote combined attention index, it can be computed as

$$A = w_v \cdot \overline{M_v} + w_a \cdot \overline{M_a} + w_l \cdot \overline{M_l} \quad (12)$$

where w_v, w_a, w_l are the weights for linear combination, and $\overline{M_v}$, $\overline{M_a}$, $\overline{M_l}$ the normalized visual, audio, and linguistic attention indices, respectively.

Linear combination of all these attention components is a straightforward approach, but human attention response is more elusive. First, the viewer may react when a subset of the attention components are higher than the others. For example, video segments with high motion attention index but low audio attention and linguistic attention indices will often trigger a viewer's response. However, linear combination will average the attention indices into a much lower value. To describe this observation mathematically, if we denote the attention components as a feature vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, where $0 \le x_i \le 1$, $1 \le i \le n$, and the *attention fusion* function as $f(\mathbf{x})$ or $f(x_1, x_2, \ldots, x_n)$, then two feature vectors with equal mean but different distribution would have different attention fusion result. To be exact, the feature vector with higher variance will have a higher attention fusion result. Hence it is necessary that $f(\mathbf{x})$ satisfy

$$f(\mathbf{x}_1) < f(\mathbf{x}_2), \text{ if } E(\mathbf{x}_1) = E(\mathbf{x}_2),\ D(\mathbf{x}_1) < D(\mathbf{x}_2) \quad (13)$$

where $E(\mathbf{x})$ and $D(\mathbf{x})$ represent the mean and variance of $\mathbf{x}$, respectively. Apparently, the *maximum function* fits the above condition (13).

Second, the more attention components with higher indices there are, the more likely the content will draw a viewer's attention (the maximum function does not reflect this characteristic, although it fits the first one). That is to say, $f(\mathbf{x})$ is a *monotone increasing function*, which can be described by the following formula,

$$f(x_1, \ldots, x_i, \ldots, x_n) < f(x_1, \ldots, x_i + \varepsilon, \ldots, x_n), \text{ if } \varepsilon > 0 \quad (14)$$

To satisfy inequalities (13) and (14), we construct an *attention fusion* function defined by (proof omitted)

$$AFF_n^{(\gamma)}(\mathbf{x}) = E(\mathbf{x}) + \frac{1}{2(n-1) + n\lambda} \sum_{k=1}^{n} |x_k - E(\mathbf{x})| \quad (15)$$

where parameter $\lambda > 0$ is a predefined constant, which controls the amount of differences between the left and right sides of inequalities (13) and (14) when $\mathbf{x}_1$, $\mathbf{x}_2$ and ε are fixed. The greater the parameter λ is, the smaller the differences are. In our implementation, λ is set to 0.2.

In calculating overall attention of a video segment, we consider separately camera motion, object motion and other basic attention components and use the attention fusion function (15). Weights can also be added to adjust the relative importance of the different attention components.

Based on attention detection, an *attention curve* is produced by calculating the attention index of each video frame. Importance index for each sub-shot is obtained by averaging the attention indices of all video fames within this sub-shot. The normalized importance of the selected sub-shot list is measured by

$$I(m, v, \theta) = \frac{1}{M} \sum_{i=1}^{M-1} impt_i^{(\theta)} \quad (16)$$

where the superscript θ indicates that the corresponding "importance" sequence is the selected sub-sequence of the original one, M is number of sub-music clips defined in Section 2.

As a byproduct, motion intensity, and camera motion (type and speed) for each sub-shot is also obtained. All these provide information for content selection, as detailed in Section 4.

3.3 Sentence Detection

In general, we want to keep conversation segments from the video in the final edited video. In particular, we do not want to break a sentence when selecting the sub-shot. Thus, it is necessary to detect each sentence boundary for further alignment.

In our system, the audio track is first segmented into pause and non-pause, then, each non-pause segment is further classified into speech and non-speech, using the approaches proposed in [9]. Non-speech segments are classified as pause segments. Finally, the audio track is segmented into sentences based on the duration of pause. If a pause segment is longer than a threshold, most likely it is the boundary of a sentence. In our implementation, the threshold is set as 300ms. At the same time, if a speech segment is less than 400ms, it is not regarded as a speech segment.

3.4 Music Analysis

In the proposed system, a selected piece of incidental music is segmented into music sub-clips by detecting strong beats. A strong beat is taken as the boundary of a music sub-clip. These music sub-clips are then used as the basic timeline for automated editing. Based on the editing requirements outlined in Section 1, shot transitions should occur at the music beats, meaning sub-shot boundaries and music sub-clip boundaries should be aligned. Music tempo of a music clip is also calculated to represent how fast or slow it is. In general, when the music is fast, the length of music sub-clips is short to generate a naturally flowing video. Furthermore, the motion intensity of a selected sub-shot should be well *matched* to the tempo of the corresponding music sub-clips. That is, when the motion in video is strong, the tempo of the corresponding music sub-clip should be also strong, and vice versa.

Instead of beat detection using a complex algorithm [14], a much simpler scheme is applied in our system. We do not detect exact the beat series, but only the onsets. The strongest onset in a time window is assumed as a beat. This is reasonable because there are many beat positions in a time window (for example, 3 second); thus, the most possible position of a beat is the position of the strongest onset. To give a more pleasant perception, the music sub-clips should not be too short or too long. In our implementation, the length of music sub-clips is limited to 3-5 seconds. Then, music sub-clips can be extracted in the following way: given the previous boundary, the next boundary is selected as the strongest onset in the current window which is 3-5 seconds away from the previous boundary.

The tempo of each music sub-clip is calculated by the onset frequency in the clip. The tempo is then normalized to [0, 1]. The higher the value is, the faster the tempo is.

4. AUTOMATED VIDEO EDITING

In this section, we introduce how we select appropriate content from a given set of raw home videos and match it with the incidental music. Firstly, we filter out low-quality segments or frames. Then, based on a number of editing rules, we select appropriate sub-shots from the original videos, and align the sub-shot boundaries with music beats and sentence boundaries in the audio track. Finally, selected video and music segments are composed into a whole, using transition effects to bridge the gaps.

4.1 Low-Quality Filtering

Since most home videos are recorded by unprofessional home users using camcorders, there are often low quality segments in the recordings. Some of those low quality segments result from incorrect exposure, shaking, poor focus during shooting, or from the fact that the users often forget to turn off the recording button so floors or walls are unintentionally recorded. Most of these low quality segments that are not caused by camera motion can be detected by examining their color and texture entropy. However, sometimes, good quality video frames also have low entropies, such as in videos of ski events. Therefore, we combine both motion analysis with the entropy approach so as to reduce false detection. That is, segments are considered possibly low quality only when both entropy and motion intensity are low. Alternatively, the approach proposed in [19] can be adopted to detect incorrectly exposed segments, as well as low quality segments caused by camera shaking. Very fast panning segments caused by rapidly changing viewpoints, and fast zooming segments are detected by checking camera motion speed (refer to Section 3.2). These are filtered out from the selection since these segments are not only blurred, but also lack appeal.

4.2 Sub-shot Selection

In Section 2, we formulated content selection as an optimization problem. The next issue is how to design the objective function. According to the two sets of rules mentioned in Section 1, there are three computable objectives as listed below:

(1) Selecting "important" sub-shots.
(2) Motion should match well with music tempo.
(3) Selected sub-shots should be nearly uniformly distributed.

Objective (1) and (3) reveal the first set of rules which deals with how to select suitable segments that are representative of the original video in content and of high visual quality. Objective (2) reveal the set of rules which deals with how to align video segments with the incidental music to increase the impact of the edited video. Of course, other computable objectives that may assist content selecting can be adopted here too.

The first objective is achieved by examining the average attention value of each sub-shot as described in Section 3.2. For the second objective, we calculate a *Correlation Coefficient* of the music tempo sequence and the motion intensity of the selected sub-shot series. That is,

$$\rho(m,v,\theta)=\rho\left(Tempo, Motion^{(\theta)}\right) \tag{17}$$

where the superscript θ indicates that the corresponding motion intensity sequence is the selected sub-sequence of the original one.

Distribution Uniformity is represented by normalized entropy. At the scene level, we define

$$H^{(SC)}(m,v,\theta)=-\frac{1}{\log K^{(SC)}}\sum_{i=0}^{K^{(SC)}-1}p_i\log p_i \tag{18}$$

where p_i = (# *of selected sub-shot in Scene*$_i$) / *M*. At the shot level, we define $H^{(SH)}(m,v,\theta)$ in a similar way. Thus the overall measure for distribution uniformity is

$$H(m,v,\theta)=k_1H^{(SC)}(m,v,\theta)+k_2H^{(SH)}(m,v,\theta) \tag{19}$$

where $k_1, k_2 \geq 0$, $k_1+k_2=1$. It is easy to see that $0\leq H(m,v,\theta)\leq 1$.

Consequently, our problem is formulated as finding θ^* which satisfies

$$\theta^*=\arg\max_{\theta}\{F(m,v,\theta),\theta\in\Theta\} \tag{20}$$

$$F(m,v,\theta)=\alpha\frac{1+\rho}{2}+\beta I+\gamma H \tag{21}$$

where $\alpha, \beta, \gamma \geq 0$, $\alpha+\beta+\gamma=1$. This is a mathematical programming problem. As explained below, the problem is more clearly re-written as a nonlinear 0-1 programming problem.

The subset $\theta\in\Theta$ can be represented by an *N*-dimensional 0-1 sequence $x=\{x_i, 0\leq i<N\}$, where $x_i=1$ if $i\in\theta$; otherwise $x_i=0$.

Then the *importance index* $I(m,v,\theta)$ is rewritten as

$$I(m,v,x)=\sum_{i=0}^{N-1}x_i\cdot impt_i \tag{22}$$

The *distribution uniformity* measure can be rewritten as

$$\begin{aligned}H(m,v,x)=k_1&\left[-\frac{1}{\log K^{(SC)}}\sum_{i=0}^{K^{(SC)}-1}p_i\log p_i\right]\\&+k_2\left[-\frac{1}{\log K^{(SH)}}\sum_{j=0}^{K^{(SH)}-1}q_j\log q_j\right]\end{aligned} \tag{23}$$

where

$$p_i=\frac{\sum_{M_i^{(SC)}}^{M_{i+1}^{(SC)}-1}x_i}{M},\ M_i^{(SC)}=\left|\{s\in SC, s<i\}\right| \tag{24}$$

$$q_j=\frac{\sum_{M_j^{(SH)}}^{M_{j+1}^{(SH)}-1}x_j}{M},\ M_j^{(SH)}=\left|\{s\in SH, s<j\}\right| \tag{25}$$

and where the $|\cdot|$ operator calculates the number of elements in a finite set. This measure is nonlinear. The *motion-tempo* matching measure can be rewritten in a similar way, and it is also nonlinear.

Consequently, the programming problem is re-written as the following nonlinear 0-1 integer-programming problem:

$$\begin{aligned}&\max F(m,v,x)=\alpha\frac{1+\rho}{2}+\beta I+\gamma H\\&\text{subject to: }\sum_{i=0}^{N-1}x_i=M,\ x_i\in\{0,1\}\end{aligned} \tag{26}$$

In the experiments presented in Section 5, $k_1=k_2=1/2$, $\alpha=\beta=\gamma=1/3$. It is obvious that this problem is not a simple linear programming problem, so it is very difficult to find an analytical

solution. When M and N are large, the optimization search space increases dramatically, and we cannot solve it using an exhaustive search. Therefore, we use a heuristic searching algorithm, the *Genetic Algorithm* (GA) [18], to find solutions approaching the global optimum. This optimization algorithm is good at finding reasonable (near optimal) solutions for search spaces which are neither continuous nor differentiable.

4.3 Boundary Alignment

As previously mentioned, two types of alignments are required in the automatic editing system, as listed below,

(1) **Sub-shot boundary and music beat alignment**. Transitions between selected sub-shots (these sub-shots are edited shots in the final output video) should occur at the beats of the music, i.e., at the boundaries between the music sub-clips.

(2) **Sub-shot boundary and sentence alignment**. A sentence should not be broken by a sub-shot boundary.

These two alignment requirements are met by the following alignment strategy.

(1) The minimal duration of sub-shots is made greater than maximal duration of music sub-clips. For example, we may set music sub-clip duration in the range between 3 and 5 seconds, while sub-shots duration in 5 to 7 seconds.

(2) Since sub-shot durations are generally greater than music sub-clips, we can shorten the sub-shots to match their duration to that of the corresponding music sub-clips.

(3) For sentence alignment, the sub-shot boundaries are shifted to ensure the sentences are contained in sub-shots. If a sentence is longer than a music sub-clip, we fade out the sentence or merge two music sub-clips.

4.4 Rendering

We use fifteen common transition effects such as cross-fade, wipe and dissolve to connect all sub-shots into one video in the rendering process. The type of the transition used for two consecutive sub-shots is determined by the similarity of the two sub-shots. This checks if they are in the same scene. The transition duration is determined by beat strength. They are described by equation (28) and (29) as below,

$$TransitionType_i = \begin{cases} Cross\ Fade, & if\ SceneID_i = SceneID_{i+1} \\ Randomly\ chose\ from\ other\ types, & otherwise \end{cases} \quad (27)$$

$$TransitionDuration_i = 1 - beat_i \quad (28)$$

where $0 \le i < M - 1$. More complex transition selection methods could be designed to take more video and music features into account, in addition to factoring in the user's preferences.

5. EXPERIMENTS AND RESULTS

Video content analysis in AVE system is processed in about 1/6 real time for MPEG1 video on a Dell 1.2GHz computer (including decoding time), while content selection and boundary alignment only take less than 10 seconds for editing 5 minutes video from a one-hour source video. Music analysis is also very fast. Five-minute music only takes about 10 seconds for analyzing. Final rendering or encoding to a video file is processed in real time. Therefore, for a one-hour video and 5-minutes music, after less than 11-minutes processing on content analysis and editing, we are able to view the final edited results, or obtain a video file after another 5 minutes for encoding or file saving.

Although it is difficult to objectively evaluate the AVE results, in the following sub-sections, we present some objective experimental results for content selection, show a number of examples of edited videos, and compare the results with randomly edited videos and manually edited videos.

5.1 Objective Evaluation of Content Selection

Table 1 shows the detailed experimental results for *GA* solutions on five videos of different types (scenery, festival, etc.), and five pieces of music of different genres (light music, pop music, etc.), labeled by Video #1 to #5 and Music #1 to 5. The five source home videos are about Hawaii (scenery, 19 minutes), Christmas (festival, 40 minutes), China (travel/scenery, 57 minutes), a wedding (event, 47 minutes) and fishing (event, 23 minutes), respectively. In Table 2, we have also compared the value of the importance index with the average value of the most "important" M sub-shots (labeled by "MAX" in the table) in the video. On average, the system kept 86% of the most "important" sub-shots while the music tempos and motion intensities matched quite well (ρ = 0.64 on average). Also the selected sub-shots are well distributed within the original input videos (H = 0.81 on average).

Table 1. Evaluation of GA solutions.

Video #	1	2	3	4	5	Average
Video Length	19m	40m	57m	47m	23m	37.2m
# of Scene	12	36	42	39	6	24.2
# of Shot	108	324	471	513	57	255.4
# of Sub-shot	258	620	1273	892	277	544.0
Music Length	4m4s	2m37s	11m56s	3m59s	1m30s	3m13s
# of Sub-music	49	33	95	51	20	41.6
ρ	0.80	0.53	0.55	0.60	0.72	**0.64**
I	0.50	0.47	0.48	0.50	0.49	**0.49**
H	0.93	0.82	0.81	0.89	0.62	**0.81**
F	0.74	0.61	0.61	0.66	0.61	**0.65**

Table 2. Evaluation of GA solutions.

Video #		1	2	3	4	5	Average
I	GA	0.50	0.47	0.48	0.50	0.49	**0.49**
	MAX	0.58	0.53	0.57	0.61	0.55	**0.57**
	GA/MAX	86%	87%	84%	84%	89%	**86%**

Figure 3 shows an example curve, which illustrates the matching index of the motion intensity and music tempo of the optimal solution for Video #1 and Music #1.

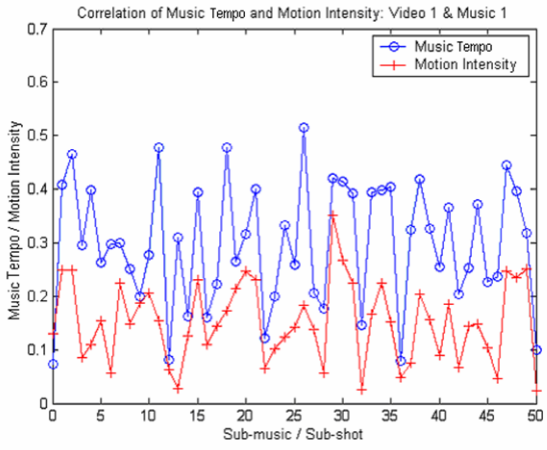

Figure 3. Matching curve of music tempo and video motion.

5.2 Examples

Three video clips demonstrate three edited videos produced by the system presented in this paper are available for downloading at http://research.microsoft.com/users/xshua/AVE. We summarize the results as following.

Hawaii: A 4-minute video created from a 19-minute scenery video (Video #1) and a piece of 4-minute-long light music (Music #1). The raw home video is segmented into 258 sub-shots, 49 of them are selected in the final result. This video sample is also submitted as a video figure.

Xmas: The original video is a 40-minute home video shoot during Christmas (Video #2). A 2.5-minute-long Christmas song (Music #2) accompanies. 33 sub-shots are selected from the raw video which is segmented into 620 sub-shots.

China: A 12-minute video generated from a 57-minute video (Video #3) shoot in China, accompanied by classical Chinese music (Music #3). The original video is segmented into 1273 sub-shots, 95 of them are selected in the final result.

5.3 Subjective User Study

To subjectively evaluate AVE results, we compare the auto-edited videos with the videos produced by connecting randomly selected video segments (sub-shots) and the videos manually edited by a *unprofessional* user (using the professional video editing tool, *Adobe Premiere*), who is fond of and familiar with editing home videos but knew nothing about how AVE works.

Ten evaluators majored in arts are invited to do the user study. The three sets of videos and music, which we used for producing the three aforementioned video examples, are employed in the user study. To obtain more reasonable result, three different sets of randomly edited videos are taken in this evaluation, while only one set of manually edited video are applied due to huge labors are required for manually editing. Two sets of edited videos that produced by AVE without enabling attention detection or music matching/alignment are also put in the evaluation. Accordingly, there are 21 videos in total. Each set of edited videos that generated from the same video/music source are randomly ordered and renamed, thus both the authors and evaluators don't know the producers of the videos just by looking at the names of the videos (i.e., we may say the evaluation is double blinded). All users are required to give a satisfaction score (0~1) to each edited video, which reflects "informativeness" and "enjoyablilty" [10]. The scores of the first set of videos generated by random content selection are fixed to 0.50 thus the users can take them as examples to giving scores for other results. Detail evaluation results are listed in Table 3, including average satisfaction values, average value of the objective function (Equation 26) (F_a) and the average editing time (T_a, in minutes). The results show AVE has much higher satisfaction than random results. The main reason for this evaluation result is, random editing loses more important segments than AVE (sometimes low quality segments are even selected), as well as does not align the music with the shot boundaries. From Table 3, we can also see AVE has very close satisfaction to manually edited results, but AVE only takes about 11% of time as manually editing. Although manually editing could choose relatively more "important" or representative segments from the video, it is not easy for an unprofessional user to manually synchronize music beats with the shot boundaries, as well as align motions with music tempos. This may be the main reason for AVE results are close or even a little bit better than manually edited results. And, as listed in the table, the comparison results between full AVE and AVE without enabling attention detection or music matching show that the criteria we proposed have significant contributions to the editing results. A better evaluation would be to compare AVE results with professionally edited videos, as to be discussed in Section 7.

Additionally, F_a has the similar trend as the subjective evaluation values, which supports that the computable objectives we used are reasonable.

Table 3. AVE subjective evaluation.

Methods	*Hawaii*	*Xmas*	*China*	*Average*	F_a	T_a
***Random* 1**	0.50	0.50	0.50	**0.50**	0.37	8 m
***Random* 2**	0.70	0.70	0.43	**0.61**	0.35	8 m
***Random* 3**	0.49	0.54	0.57	**0.53**	0.26	8 m
Manual	0.79	0.87	0.83	**0.83**	0.43	210 m
AVE*	0.70	0.81	0.74	**0.75**	0.41	10m
AVE **	0.65	0.73	0.69	**0.69**	0.34	9m
AVE	0.83	0.90	0.83	**0.85**	0.65	12 m

Note: AVE* and AVE** are the results of AVE without enabling attention detection and music matching/alignment, respectively.

6. EXTENSIONS

We have extended the automated video editing system to a number of other functions, including Best Incidental Music Recommendation and Editing Styles.

6.1 Best Incidental Music Recommendation

By comparing the objective function values, we have implemented an experiment to pick the most suitable music from Music #1 to Music #5 for Video #1. The objective function values of optimal solutions are shown in Table 4.

Table 4. Best matching music selection.

Music No.	#1	#2	#3	#4	#5	Average
ρ	0.80	0.71	0.69	0.62	0.67	0.70
I	0.50	0.50	0.49	0.48	0.49	0.49
H	0.93	0.90	0.93	0.88	0.88	0.90
F	0.74	0.71	0.70	0.66	0.68	0.70

From the table, it is seen that Music #1 is the most suitable one for Video #1. The example video clip used for "Hawaii" mentioned in Section 5.2 is the one created from Video #1 and Music #1.

6.2 Editing Styles

Individuals have different preferences on what is interesting or "important". By adjusting the parameters of the attention fusion function and the objective function of optimization processes, we can obtain different sets of video segments. On the other hand, different transition effects or frame effects, such as *grayscale*, *sepia tone*, *old movie*, and so on, also make the output videos have different appearances.

Based on the above ideas, the following editing styles have been designed. More editing styles can be designed based on users' preferences.

Music Video: adjust the durations of music sub-clips based on the average tempo of the music. That is, a fast music clip will result in fast shot changes in the output video, and vice versa. In addition, the weight for matching motion with tempo is increased to 1/2 from the default setting of 1/3.

Highlights: This is achieved by increasing the weight of sub-shot importance index in selecting sub-shots to 1/2 from the default setting of 1/3.

Old Movie: This is generated by adding "old movie" noises on each video frame to simulate old-age film.

Day by Day: Group sub-shots into individual days and insert a short manually-made shot which contains a caption indicating the corresponding date.

7. CONCLUSION AND DISCUSSION

In this paper, an effective system that automates home video editing was presented. Given a piece of incidental music, a series of video segment highlights that satisfy certain editing rules are automatically extracted from an input raw home video based on the content of the video and music. The final output video is rendered by connecting the selected video segments with specific transition effects and aligning then with the incidental music. In addition, under this framework, we can choose the best-matched music for a given video and support different output styles.

There are a number of possible improvements for this system. For example, as mentioned in [11], professional video editors use a set of film and TV editing patterns as rules, a so-called *video grammar*. Matsuo et al. proposed an approach based on data mining to discover editing patterns [11]. This technology, as well as TV and film editing patterns may be adopted into this system to make the editing result more professional looking.

In this paper, motion intensities of the video segments are matched with the tempos of the corresponding music sub-clips, as well as shot changes occurring at the music beats. However, further analysis on this scheme is necessary, such as to study how and to what extent it affects the perception of the video content. Additionally, as observed from typical music TV, there are more matching aspects which could be used such as video playback speed and music speed. We may design better matching scheme by taking these kinds of features into account.

Furthermore, how to obtain better semantic story-telling in the output video remains a challenging task. Generally professionally edited videos have more semantic meanings, as human understanding of the stories in the source video are taken into consideration. However, current technologies on computer vision and artificial intelligence are far from this shape. Nevertheless, our next step work will also try to explore how to better automatically or semi-automatically expose semantics in the output videos. For example, face detection and tracking may assist to create music videos that have a "central character" or "leading actor". In addition, semantic classification of video shots, such as indoor vs. outdoor, cityscape vs. landscape, beach, sun rising/falling, moon night, etc., may also facilitate semantic content selection.

8. REFERENCES

[1] Foote, J., Cooper, M., and Girgensohn, A. Creating Music Videos Using Automatic Media Analysis. *ACM Multimedia 2002*.

[2] Gong, Y.H., and Liu, X. Video Summarization Using Singular Value Decomposition**.** *Proc. of CVPR*, June, 2000.

[3] Hanjalic, A., Lagendijk, R. L., and J. Biemond. Automated Highlevel Movie Segmentation for Advanced Video-Retrieval Systems. *IEEE Trans on Circuits and Systems For Video Technology*, Vol. 9, No. 4, June 1999, 580-588.

[4] Itti, L. Real-Time High-Performance Attention Focusing in Outdoors Color Video Streams. *Proc. SPIE Human Vision and Electronic Imaging IV (HVEI'02)*, Jan 2002.

[5] Jeho, N., and Ahmed, H.T. Dynamic Video Summarization and Visualization. *Proc. of ACM Multimedia*, October 1999.

[6] Kender, J. R., and Yeo, B. L. Video Scene Segmentation via Continuous Video Coherence. *Proc IEEE Intl Conf on Computer Vision and Pattern Recognition 1998*, 367-373.

[7] Li, S.Z. et al. Statistical Learning of Multi-View Face Detection. *Proc. of ECCV 2002*.

[8] Lin, T. and Zhang, H.J. Video Scene Extraction by Force Competition. *ICME*, 2001.

[9] Lu, L., Jiang, H. and Zhang, H. J. A Robust Audio Classification and Segmentation Method. *ACM Multimedia 2001*.

[10] Ma, Y. F., Lu, L., Zhang, H.J., and Li, M.J. A User Attention Model for Video Summarization. *ACM Multimedia 2002*, 533-542.

[11] Matsuo, Y., Amano, M., and Uehara K. Mining Video Editing Rules in Video Streams. *ACM Multimedia 2002*, 255-258.

[12] Omoigui, N., He, L., Gupta, A., Grudin, J., and Sanoki, E. Time-compression: System Concerns, Usage, and Benefits. *Proc. of ACM ICH 1999*.

[13] Orriols, X., and Binefa, X. An EM Algorithm for Video Summarization, Generative Model Approach. *ICCV* 2001.

[14] Scheirer, E. Tempo and Beat Aalysis of Acoustic Musical Signals. *Journal of the Acoustical Society of America*, 103(1), 588-601, 1998.

[15] Smith, M.A., and Kanade, T. Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques. *Proc. of* Computer Vision and Pattern Recognition 1997.

[16] Stefanidis, A., Partsinevelos, P., Agouris, P., and Doucette, P. Summarizing Video Datasets in the Spatiotemporal Domain. *Proc. of 11th Intl. Workshop on Database and Expert Systems Applications, 2000.*

[17] Sundaram, H., Xie, L., and Chang, S.F. A Utility Framework for the Automatic Generation of Audio-Visual Skims. *ACM Multimedia 2002*.

[18] Whitley, D. A Genetic Algorithm Tutorial. *Statistics and Computing*, Vol. 4, 64-85, 1994.

[19] Yan, W.Q., Kankanhalli, M. Detection and Removal of Lighting & Shaking Artifacts in Home Videos. *ACM Multimedia 2002*, 107-116.

[20] Zhang, H.J., Kankanhalli, A., and Smoliar, S.W. Automatic Partitioning of Full-Motion Video. *Multimedia Systems*, 1, 10-2, 1993.