Creating Music Videos using Automatic Media Analysis

Jonathan Foote, Matthew Cooper, and Andreas Girgensohn
FX Palo Alto Laboratory, Inc.
3400 Hillview Avenue, Bldg. 4
Palo Alto, CA 94304, USA
{foote, cooper, andreasg}@fxpal.com

ABSTRACT

We present methods for automatic and semi-automatic creation of music videos, given an arbitrary audio soundtrack and source video. Significant audio changes are automatically detected; similarly, the source video is automatically segmented and analyzed for suitability based on camera motion and exposure. Video with excessive camera motion or poor contrast is penalized with a high unsuitability score, and is more likely to be discarded in the final edit. High quality video clips are then automatically selected and aligned in time with significant audio changes. Video clips are adjusted to match the audio segments by selecting the most suitable region of the desired length. Besides a fully automated solution, our system can also start with clips manually selected and ordered using a graphical interface. The video is then created by truncating the selected clips (preserving the high quality portions) to produce a video digest that is synchronized with the soundtrack music, thus enhancing the impact of both.

Keywords

Video editing, video analysis, audio analysis, music video.

1. INTRODUCTION

The widespread proliferation of personal video cameras has resulted in huge a data management problem. Virtually all camcorder owners own a dusty pile of home-recorded videotapes that are too precious to throw away, but too tedious to actually watch in their entirety. The situation is aggravated by the poor sound quality of informal video. Without professional sound recording and post-production, even otherwise well-produced video appears amateurish. In fact, studies have shown that poor sound quality degrades the perceived video image quality [17].

We present a solution: a fully- or semi-automatic video summarizer that can condense a lengthy home movie into a compelling 3 minute music video, set to the music of one's choice, and suitable for sharing with friends and family. In operation, the user selects a favorite musical work for the soundtrack, and a longer source video. Typically this will be a home video, though any source video may be used. The music is automatically segmented and analyzed for tempo, while the video is automatically segmented and analyzed for unsuitability. The results of this analysis include a time-indexed audio novelty score, in which peaks correspond to points of significant change in the soundtrack audio. The video analysis produces a time-indexed unsuitability score which is based on estimated camera motion and brightness. These two time series have peaks at segment boundaries in the music and clip boundaries in the video, respectively. To align the video and audio tracks, video clips are truncated, combined, and/or discarded such that the final set of clip boundaries align exactly with significant audio changes to produce a professional-appearing music video. Raw video with unsuitable camera motion or exposure is preferentially discarded by the algorithm. By selecting the soundtrack music, the user can tailor the tempo and mood of the resulting high-quality video, making it into a more personal statement.

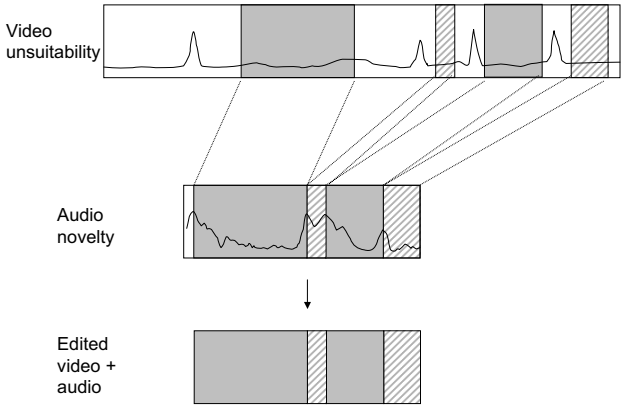


Figure 1. Automatically editing video to synchronize with a shorter audio soundtrack.

Music videos can also be created semi-automatically using a graphical interface. In this mode, automatic video analysis is used to segment the source video into clips. The user then selects and orders the video clips on a timeline. The user may optionally lock a video clip start or endpoint. The system determines the lengths and portions of the remaining clips by using automatic analysis to preserve video with low camera motion and good exposure.

In developing this system, we have relied on several key assumptions. The first is that *improved soundtrack quality improves perceived video image quality*, and thus using professional-quality audio for home video will improve the perception of the latter. This fact is a truism in the film industry, and has been affirmed in a number of studies. One study at MIT showed that listeners judge the identical video image to be higher quality when accompanied by higher-fidelity audio [17]. A second assumption is that *synchronizing video and audio segments enhance the perception of both*. Again, this is a common practice of cinematic sound editors world-

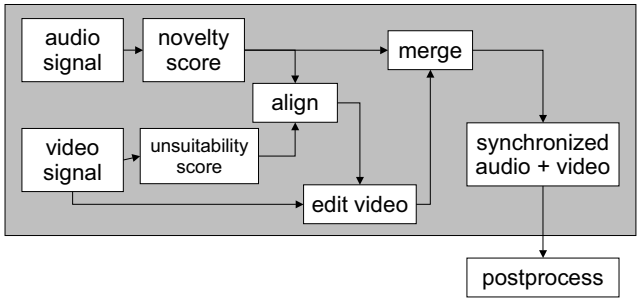


Figure 2. Automatic music video system block diagram

wide, and furthermore is backed by quantitative results. For example, Lipscomb [13,14] presents user studies demonstrating that the "effectiveness" of a film clip is generally enhanced when audio events are synchronized with video events. A third assumption is that *users require control over which video clips are included*, and thus fully-automatic solutions will be generally less satisfactory than a user-assisted approach. Section 5 presents a user interface that affords a great deal of user control with very simple operation.

2. RELATED WORK

Automatically aligning video with unrelated audio is relatively novel, though there has been work on aligning lip movements to recorded speech [1]. Several groups have reported work on musical beat-tracking and analysis. A recent approach uses correlated energy peaks across sub-bands [20]. Another approach relies on restrictive assumptions that the music be in 4/4 time with a bass drum on the downbeat [9]. Existing approaches commonly make limiting assumptions about rhythmic features of the audio signal, and will thus fail for any music, such as orchestral, which lacks them. In contrast, the methods used here have been shown to be robust across a wide variety of jazz, orchestral, and popular musical genres [5].

Various methods have been proposed to summarize video [2,18,21]. These methods chiefly rely on transcribed text to determine the video segments to be included in summaries. Such data is not usually available for home video. Work at Intel [12] discusses the issues of low quality home video and presents an automatic digest creation for home video. Their method selects portions of video shots with good quality and inserts video effects for transitions, however audio considerations are not addressed.

A commercial venture, muvee.com [16], advertises an automatic system for producing music videos. Though no details of the algorithm are available, editing is accomplished using a rule-based system to produce one of several video "styles." Although the system claims to automatically analyze the soundtrack audio and video streams, the system does not include any means for user interaction like that provided by our interface (as described in Section 5). Suzuki et al. have built video editing tools for composing movies, using heuristics derived from music theory [23]. As a result, videos manually produced using this system are well synchronized with sound, which is deemed to be desirable. This is an encouraging result for this work, which performs a similar task automatically.

3. AUDIO AND VIDEO ANALYSIS

Figure 2 shows a block diagram of the video creation system. First, video and audio are selected for input to the system. The input audio need not be related to the video in any way — to avoid confusion with the soundtrack from the source video (if any) the input audio is referred to as the soundtrack audio. The video may contain sound, which can either be discarded or mixed with the soundtrack audio to produce the final music video soundtrack. First, the source video is divided into clips. When editing video in the DV digital camcorder format, we denote a camera on-off sequence as a "take." These high-level boundaries can be determined from metadata in the DV stream. Each "take" is further segmented into "clips" separated by areas of fast camera motion. If video take boundaries are unavailable, they can be estimated in a similar manner as that used to determine the audio change [4], or by any number of video segmentation algorithms [2]. Next, the audio is analyzed to detect segment boundaries, which correspond to peaks in a "novelty score." The video clips are then automatically edited by discarding unsuitable portions so that the remaining video is aligned with the audio changes. Figure 1 depicts schematically how video clip boundaries are aligned to major changes in the audio, as described further in Section 4.

3.1 Audio Parameterization

A crucial step is the initial audio analysis. If changes cannot reliably be detected, the perceived quality of the resulting music video will suffer, per our assumptions. Straightforward approaches to audio segmentation based on spectral differences are generally unsatisfactory because they yield too many false alarms. Typical speech and music constantly fluctuate, and it is difficult to discriminate significant changes from ordinary variation. Instead, we employ audio self-similarity analysis, following [5]. At each instant, the self-similarity for past and future regions is computed, as well as the cross-similarity between the past and future. A significantly novel point will lie between regions of high self-similarity. These regions before and after the novel point will also exhibit low cross-similarity. The temporal extent of the "past" and "future" can be varied to change the scale of the analysis. An advantage of this approach is that it effectively uses the signal to model itself, and thus requires minimal assumptions about the nature or genre of the analyzed signal. This makes it robust to a wide variety of input sources, from Vivaldi to heavy metal.

For audio, we use a reasonably standard spectral parameterization, based on the short-term Fourier transform (STFT). Audio is first converted to a monophonic representation at a F_s = 22.05 kHz sampling rate. This is analyzed in short frames of 512 samples, spaced at 1/30 second intervals (735 samples). The DFT of each window is taken, and the log of the magnitude is calculated. The resulting power spectrum is quantized into 30 bins evenly spaced from 0 to $F_s/4$ Hz (roughly 5.5 kHz).This results in a 30-dimensional feature vector at a 30 Hz frame rate. The sampling rates, window sizes, or quantization bins may be varied to tailor the analysis to emphasize particular audio features. We have also successfully experimented with alternative audio parameterizations including Mel-frequency cepstral coefficients, Mel-scaled spectrograms, and spectral features based on the Karhunen-Loeve transform [22].

3.2 Self-Similarity Analysis of Audio

Once the signal has been parameterized, it is then embedded in a two-dimensional representation. The key is a measure D of the (dis)similarity between feature vectors $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ calculated from

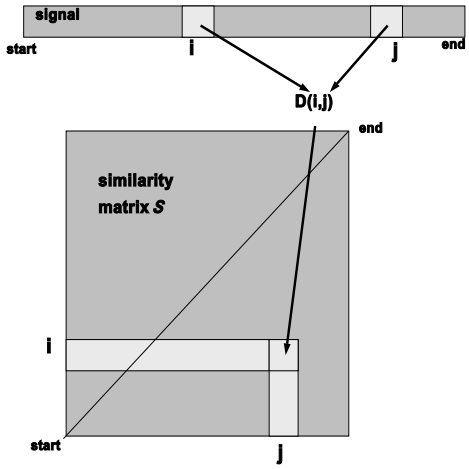


Figure 3. Similarity matrix embedding

frames *i* and *j*. We employ the cosine of the angle between the parameter vectors.

$$D_C(i,j) \equiv \frac{\boldsymbol{v}_i \bullet \boldsymbol{v}_j}{\|\boldsymbol{v}_i\|\|\boldsymbol{v}_j\|}$$

This similarity measure has the property that it can yield a large similarity score even if the vectors are small in magnitude. This is generally desirable so that similar regions with low energy will be judged highly similar. For most applications, subtracting the overall mean from each vector produces a more informative similarity score.

The similarity between every pair of instants in a source signal can be embedded in a square matrix ***S,*** such that the *i, j*th element of ***S*** is $D_c(i,j)$, as shown in Figure 3. Generally, ***S*** will have maximum values on the diagonal (because every window will be maximally similar to itself); furthermore if *D* is symmetric then ***S*** will be symmetric as well. ***S*** is readily visualized as a square image [5]. Each pixel *i, j* is colored with a gray scale value proportional to the similarity measure *D(i,j)*. These visualizations let us clearly see the structure of an audio file. Regions of high similarity, such as silence or a single note, appear as bright squares on the diagonal. Repeated notes are visible as bright off-diagonal rectangles. A high degree of repetition in the audio will be visible as diagonal stripes, offset from the main diagonal by the repetition time.

Figure 4 shows a similarity matrix for 30 seconds of the song *The Magical Mystery Tour* by The Beatles. Coherent audio segments are visible as large bright squares along the main diagonal. Segment boundaries produce distinctive checkerboard features on the diagonal, for example near 78 and 82 seconds. These correspond to significant chord changes in the bridge.

3.3 Audio Segmentation via Kernel Correlation

We measure the novelty as a function of time by detecting the checkerboard-like features along the main diagonal of the similarity matrix. Travelling along the diagonal corresponds to moving along in time. White squares on the diagonal correspond to high self-similarity regions; black squares on the off-diagonals correspond to regions of low cross-similarity. Using the cosine metric, similar regions will be close to 1 (brighter in Figure 4) while dissimilar regions will be closer to -1 (darker). To measure audio

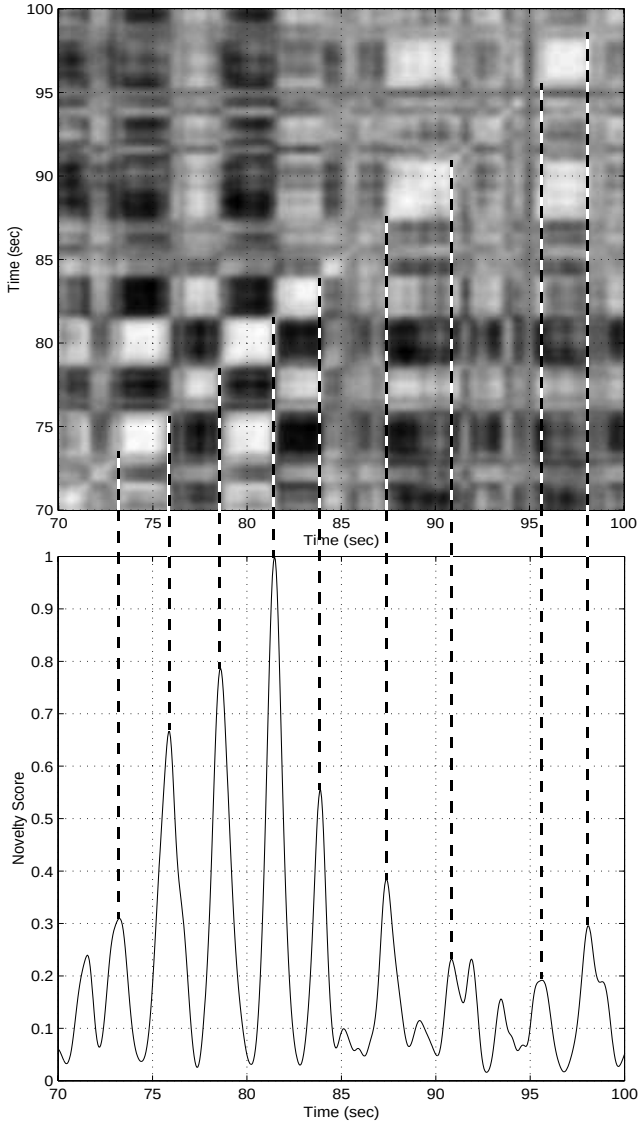


Figure 4. (top) Similarity matrix visualization for *The Magical Mystery Tour* by The Beatles.

Figure 5. (bottom) Measure of change derived from similarity matrix, using 121 x 121 sample checkerboard kernel

change, we determine how much a particular point on the diagonal looks like the crux of a checkerboard, that is, we look for regions of low cross-similarity (dark on the upper left and lower right) and high self-similarity before and after (white upper right and lower left). We measure this using a classic "matched filter" technique: by running a checkerboard along the diagonal and seeing how well it matches. This can be done by correlating ***S*** with a kernel that itself looks like a checkerboard. Perhaps the simplest is the 2x2 unit kernel:

$$C = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Larger kernels are easily constructed and can be smoothed to avoid edge effects using windows that taper towards zero at the edges. For the experiments presented here, a radially-symmetric Gaussian

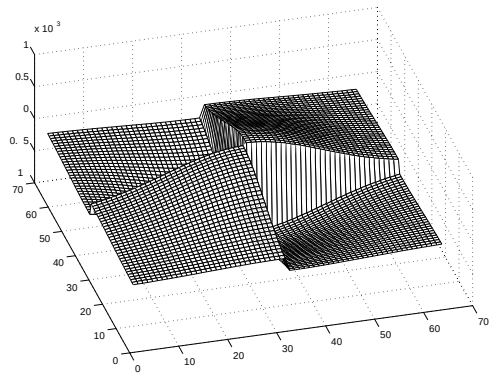


Figure 6. 64 x 64 checkerboard kernel with Gaussian taper

function is used. Figure 6 shows a a 64 x 64 checkerboard kernel with a radial Gaussian taper having $\delta = 32$.

Correlating a checkerboard kernel along the diagonal of similarity matrix ***S*** results in a measure of novelty. We calculate the correlation along the main diagonal of ***S*** to obtain a time-indexed measure of audio novelty *N(i)*, where *i* is the frame number:

$$N(i) = \sum_{m=-L/2}^{L/2} \sum_{n=-L/2}^{L/2} C(m, n) S(i + m, i + n)$$

The width of the kernel *L* directly affects the properties of the novelty measure. A small kernel detects novelty on a short time scale. Larger kernels average over short-time novelty, such as notes, and detect longer-term structure. We use this property directly for music video creation. To ensure a minimum edited video clip length of a few seconds, we use kernels of about this width to segment the audio soundtrack.

Figure 5 shows the novelty measure for an excerpt from *The Magical Mystery Tour*. Peaks in the novelty measure correspond to audio segment boundaries; the larger the peak, the more dramatic the change. A 121 x 121 kernel was used, corresponding to a kernel width of 6.05 seconds. Large peaks have been annotated with the corresponding times in the similarity matrix, showing how they detect significant audio transitions.

3.4 Segmenting and Editing Video

In general, video must be discarded to match the audio, which is typically shorter. Our approach is to preferentially discard video that is unsuitable due to excessive camera motion or poor exposure, both common problems in amateur video. To this end we compute an "unsuitability score" which is a measure of both camera motion and video brightness. High values of the unsuitability score indicate excessive camera motion or overexposure.

When editing video in the DV digital camcorder format, video take boundaries are stored in the video data. Takes can be further subdivided into "clips" based on camera motion and the amount of brightness. For example, a take might contain two still portions separated by a fast pan. In such a situation, it is better to have two clips and to trim each of them independently rather than attempting to manipulate the whole take.

To quantify the suitability, we represent estimated camera motion and brightness as a numeric unsuitability score [6,8]. To detect excessive camera motion, we first estimate camera speed and direction. This is done by finding the shift that results in the minimum root-mean-square difference between adjacent frames. The first difference is computed based on a shift of 32 pixels in the eight cardinal and diagonal directions; the second based on 16; and so forth. This vector is averaged over five frames. Because excessive tilt is subjectively worse than pan, we penalize the horizontal speed by a factor of three. We normalize this to an unsuitability score proportional to the square root of the camera speed. An unsuitability score of 0.5 represents a pan of one frame width in one second or a tilt of one frame height in two seconds. We consider this to be the maximum desirable camera motion; higher unsuitability scores help ensure that video with more objectionable motion is not included in the final output.

For determining suitability based on the brightness of a video frame, we compute the fraction of the total pixels above a brightness threshold. For the videos in our library[1], we found that 20% of the pixels with at least 45% brightness is a reasonable exposure. For the unsuitability score, we map the range between 0% and 20% to a linearly decreasing score. For the overall unsuitability score, we use the maximum of the brightness and motion unsuitability.

Figure 7 shows the unsuitability score for video taken by a colleague. The amount of camera motion here is excessive by professional video production standards, but common in home video. We identify candidate clip boundaries as peaks in the moving average of the unsuitability score. The window size for the moving average is adjustable to get more or fewer video clips. The peak threshold is adjusted between 0.5 and 1 proportionally to the window size. The gray area in the figure represents the moving average over a window of 1.5 seconds. The figure shows where the video is segmented into clips at average peaks that exceed a threshold of 0.75 (clip boundaries at 236.2, 242.1, and 277.9 seconds).

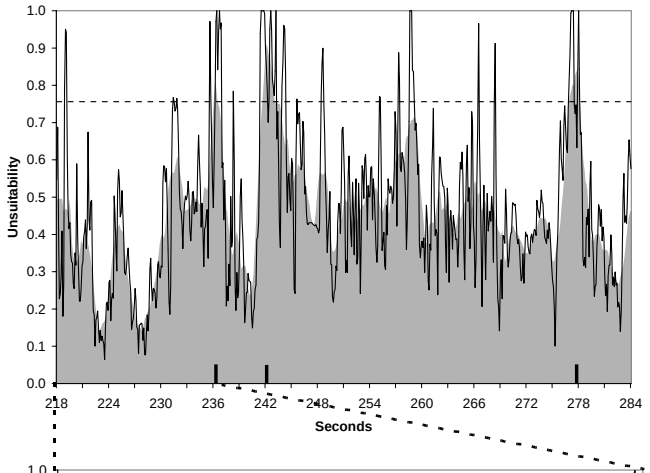
For the alignment to the soundtrack audio, we automatically adjust the window size until the number of clip boundaries equals the number of peaks in the audio novelty score. We then locate the peak for each part of the moving average curve that is completely above the threshold. Those peaks are our candidate clip boundaries. If a clip between the candidate boundaries does not meet the minimum length requirement (e.g., three seconds), we keep merging it with its neighbor until the minimum length is reached. Such a segmentation leads to clips mainly below the unsuitability threshold with regions above the threshold at the endpoints. Figure 8 for shows a video clip from Figure 7 in higher time resolution.

Depending on the audio, a smaller or larger portion of the low unsuitability region will be included in the video. For a given clip length, we choose the portion that minimizes the area under the unsuitability curve. Increasing the clip length pushes the selected region out into the "hills" of unsuitability. Figure 8 shows how increasing the clip length results in the black, dark gray, and light gray areas being respectively selected.

4. ALIGNING AUDIO AND VIDEO

Given the peaks from the audio novelty score and the video clip boundaries, we have experimented with several methods for align-

[1] 36 home and trip videos in DV format ranging from 10 to 90 minutes taped by user study participants, researchers, and friends. Total length is 1035 minutes.



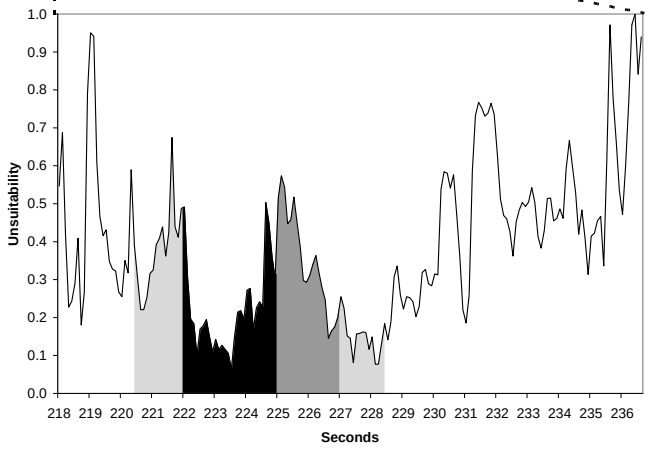


Figure 7. (top) Unsuitability Score with Clip Boundaries
Figure 8. (bottom) Selected Clip Portions

ing the audio and the video. One simple solution is to count the number of video clip boundaries and to select the corresponding number of the largest audio changes. This approach has the advantage that no thresholds are needed; ranking peaks by height results in a *de facto* self-adjusting threshold. If a fixed list of audio segments is supplied, then video clips can be selected and edited in time-order until there are enough suitable clips for the final music video. Those peaks in the audio novelty score can then be aligned with the video clip boundaries. As discussed in the previous section, the length of the video clips can then be adjusted to match the audio. If the time between two audio peaks is longer than the available clip, several clips can be combined. We assume, of course, that the video is longer than the audio[1]. If not, a simple solution is just to duplicate the video so some parts are repeated. We also assume home video is highly redundant, and if only portions of most clips are retained, the video can be reduced in length by a large factor without significantly omitting desired content.

Figure 1 illustrates our method for automatic music video creation. Once change measures have been computed for the soundtrack audio, major peaks are aligned with the video clip boundaries. The necessary clip lengths are found from the distance between audio novelty peaks. Then the video clips are truncated to the length of each audio segment according to the suitability score.

Automatic analysis of home video often produces a large number of video clips, owing to the typically poor quality of amateur video. Reducing the sensitivity of the video segmentation by increasing the window size for the moving unsuitability average can help, as can only using the video clips of a higher average suitability. In practice, we base the segment structure of the music video on that of the (professionally produced) soundtrack audio, which leads to a more coherent final product. We threshold the peaks of the audio novelty score so that only major audio changes are considered. In addition, we impose a minimum audio segment length of three seconds.

For fully-automatic operation, we can select the video clips with maximum suitability. To reorder video clips to match the audio, the average unsuitability score can be minimized by aligning the longest audio segments with the maximally suitable video clips. A simple approach is to sort the audio segments by descending length. For each segment, locate the most suitable remaining video clip for that length (time complexity $O(n^2)$, where n is the number of audio segments). Alternatively, best-first tree traversals or Dynamic Programming techniques (DP) [11] can improve and speed up the matching process. We are investigating DP to rearrange the order of video clips, for example, to match energetic musical passages to clips containing a lot of motion. The DP algorithm is well suited to the special conditions of this application; in particular the distance and matching weights can be chosen such that large peaks must match, and that large time jumps in the video are not penalized.

5. USER CONTROL

In many cases, it is preferable to let the user select and order the clips rather than letting the system do this automatically. This ensures that clips most important or interesting to the user are included, regardless of their quality. In this case, video clips are automatically segmented and then presented to the user, who then selects the ones to use. If the system produces too many or too few clips, the user may change the segmentation sensitivity.

We have built an interactive video editing system, dubbed Hitchcock, that facilitates easy drag-and-drop video editing [7]. The Hitchcock system allows users to connect a DV camcorder to a PC, then automatically copy the video to the PC and extract information such as camera on/offs and recording times for each take. The system calculates the video unsuitability, computes color histograms for clustering, and selects appropriate keyframes to represent each clip. Once the video has been copied and processed (which takes real time for the copying and half as much again to process), it can be edited with the Hitchcock user interface.

The Hitchcock interface allows the user to select clips from the raw video, and to easily adjust their length and order. Two graphical display regions comprise the Hitchcock user interface, as shown in Figure 9. The top region lets users select clips from the raw video. The bottom display allows users to order the clips along the timeline and change the lengths of the clips.

We cluster clips by one of several criteria such as similar recording times or similar color. Similar clips are placed into the same "pile." In Hitchcock, each clip is represented by one keyframe in a pile, and clips are stacked in temporal order. Mousing over the pile

[1] In our experience, this is a relatively safe assumption. The reader is invited to recall examples of home videos shorter than a typical 3-minute popular song.

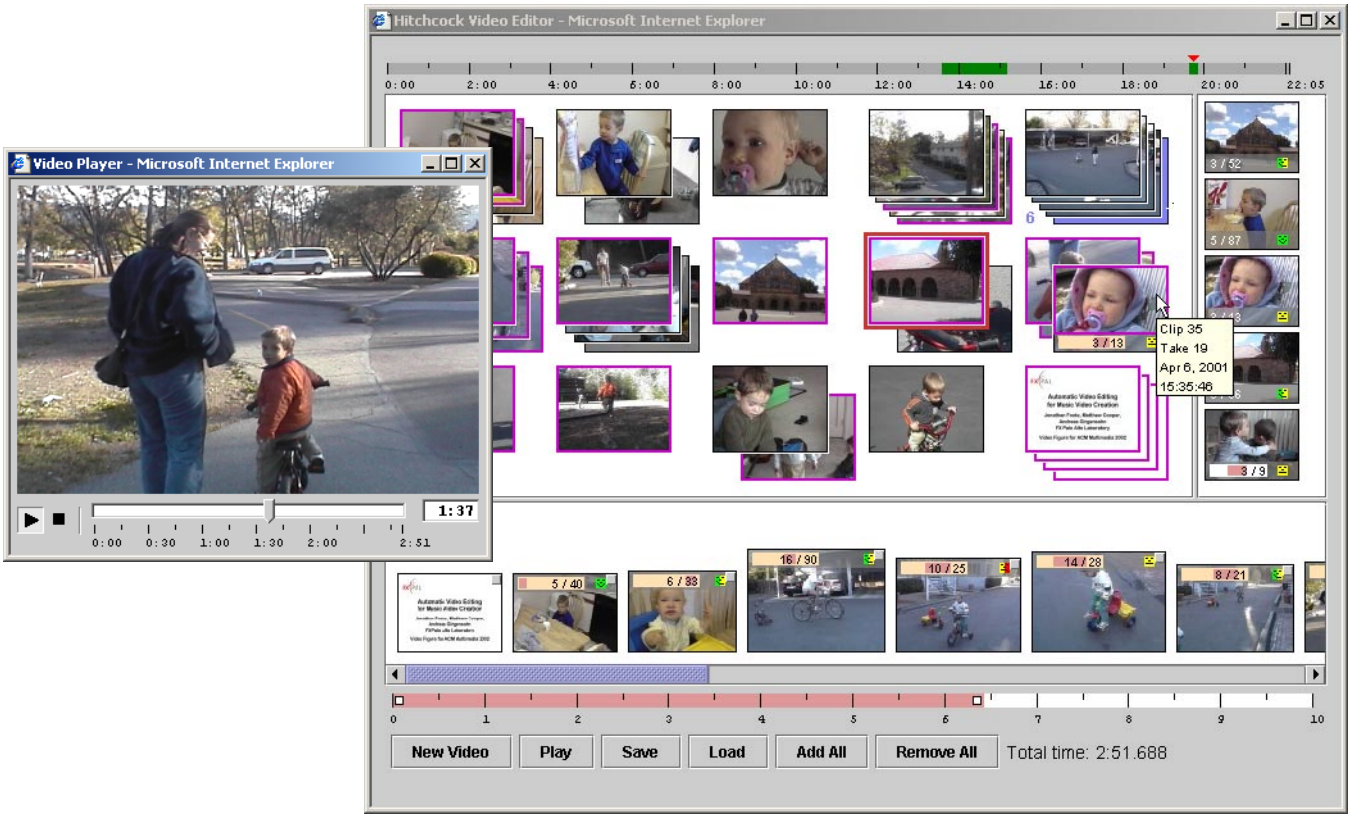


Figure 9. Hitchcock drag-and-drop video editing system.

reveals keyframes for the first five clips in the pile, so the pile's contents can easily be determined. To see keyframes for the remaining clips in the pile, the user can expand the pile by clicking on it. When expanding a pile, the current display is faded out and the keyframes of the expanded pile are shown. The timeline at the top displays the time extent of the pile in a darker color.

The lower region is a timeline for composing the output video. The user drags keyframes from the selection display and places them along the timeline as in a storyboard. Keyframes, and hence clips, can be reordered by dragging them around the timeline. Each keyframe's area in the storyboard is roughly proportional to the clip length. A handle in the corner of each keyframe allows the user to modify the clip length by resizing the representative keyframe. However, the user-selected length may be overridden by the alignment process. For each clip, the system determines a preferred length based on the user-selected keyframe size. The system automatically truncates the clip to the best portion of the preferred length using the unsuitability score described in Section 3.4. If users do not like how the system truncates a clip, they can may lock one or both ends to ensure that the clip length is only modified at the unlocked end, allowing the user to preserve the desired portions of the clip in the final music video (see Figure 10).

Users can also create title images to be placed between clips or at the beginning. Hitchcock automatically converts each title image into a 10 second video clip that can be manipulated just like other clips. Hitchcock allows instantaneous previews, unlike other video editors that need a time-consuming rendering phase before the edited video can be seen. This allows the users to instantly view their edited video without having to wait for any rendering, at the cost of fancy transition effects. The edit decision list for the video controls the Microsoft Media Player via a Java applet (see the player window in Figure 9). The player control jumps to a new point in the source video at the end of a clip. The applet allows the edited video to be previewed with VCR deck controls and a time slider that allows the user to jump to any point. When constructing music videos, we notice that the absence of clip transition effects enhances the impact of synchronous audio and video changes.

Once the user completes a rough video edit by selecting clips, order, and preferred length, we apply the methods of Section 4 to automatically align and edit the selected clips to the chosen soundtrack music. The source audio is replaced with the music in the resulting DV AVI file. If if the user has not selected enough video to cover the audio, an alert is generated (alternatively, the system could use other unselected video or truncate the audio). Title images are treated slightly differently: preferably they will

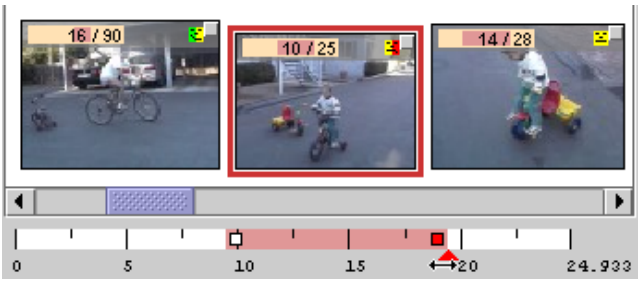


Figure 10. Resizing a video clip with a locked end.

start on a large audio change. If the users locked one or both ends of a video clip, the automatic alignment will respect those locks. This will lead to suboptimal results if both ends are locked. (Presumably the user has locked the clip for a reason, such as to ensure that a desired video passage is not automatically discarded.)

6. RESULTS

We have automatically created a number of music videos, using a variety of video sources such as the "Home Video of Lisa" [14] and videos taken by colleagues and friends, and a variety of soundtrack audio, including The Beatles, jazz, Vivaldi, and dance music. Because this produces multimedia output, it is not obvious how to present these results in a publishable form. Moreover, it would be very desirable to measure how well the algorithms work. However these are not objective results which can be benchmarked against some agreed-upon ground truth. Rather, proper evaluation requires subjective results from a large number of users. Informal judgements show that our method produces reasonably convincing music videos. Each video summary contains all the significant clips from the source video, which are noticeably aligned with musical transitions, such as soft/loud or verse/chorus changes in the soundtrack audio. For this paper, we have prepared a semi-automatically produced example music video. We condensed a 22 minute home video into a three minute music video set to *The Magical Mystery Tour* by The Beatles. We manually selected video clips with the Hitchcock user interface, added titles, and had our system automatically trim and align the video clips to the music.

Figure 11 shows the actual alignment of the first quarter of the home video example to the first quarter of *The Magical Mystery Tour* (note the different time scales). The top of the figure shows the video unsuitability score. Video takes determined by the camera are indicated by alternating light gray and white areas. The fifth take was further subdivided into four clips as indicated by dark gray areas (also shown in Figure 7). The alignment of video boundaries to audio changes (peaks in the audio novelty) is indicated by dashed lines.

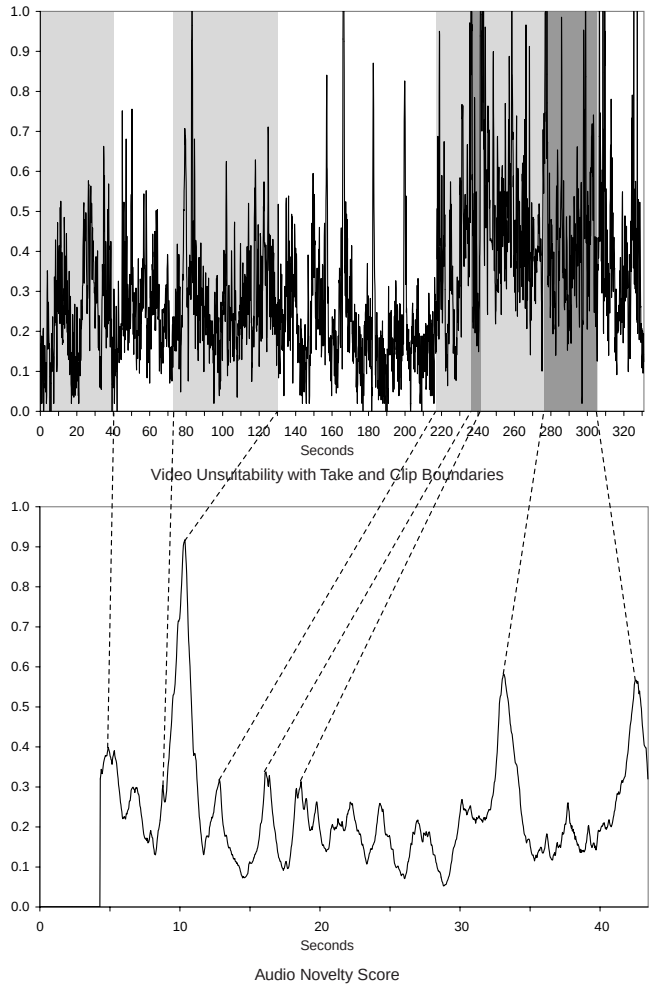


Figure 11. Aligning video boundaries to peaks in audio novelty.

In our experiments, we note a curious perceptual effect: frequent, though coincidental, correspondences between video and the unrelated soundtrack music. Surprisingly often, these enhance the overall effect. For example, children on the trampoline in the "Home Video of Lisa" often bounce in time to the music, regardless of the tempo. We call this the "Dark Side of the Rainbow" effect, after the remarkable correspondences that reputedly exist between the film *The Wizard of Oz* and the Pink Floyd album *Dark Side of the Moon*, even though these are purely coincidental [10].

7. FURTHER IMPROVEMENTS

Our first experiments cover only a small portion of the design space of these methods. Here we present some enhancements and alternative algorithms for automatic video creation.

Rhythmic synchronization

An alternative video alignment method is more appropriate for music with a distinctive tempo or beat, as is common in popular music. The music for the soundtrack audio is analyzed to detect the tempo at every moment in the music. Rhythmic analysis, such as the methods of [20] or [5], are used to extract a basic repetition time, or tempo. This serves as the minimum duration of the clip to be used at that time, henceforth called the "base duration." For each detected beat or bar, a video clip is assigned. Then a portion of the clip which has the equal length with the base duration is extracted. Concatenating the truncated clips yields a video digest synchronized to the chosen music. Using this method, slow-tempo music like ballads results in longer clips because the base duration is longer. On the other hand, more upbeat popular music results in shorter clips and more rapid video changes, a la MTV. To set a mood, users may also specify the desired length of included video clips as a multiplier of the base duration. In this case, correspondingly fewer video clips are used.

Combining source audio with soundtrack music

In the current system, any existing video sound is simply discarded. However, in many cases it might be desirable to retain the source audio, especially if it contains narration or other information. In this case, it is a straightforward matter to mix the existing video sound with the chosen soundtrack music, although it may have arbitrarily truncated edit boundaries. To avoid the arbitrary truncation, one can attempt to find sentence boundaries by finding long enough passages of silence. Video clips can then start and end at a sentence boundary.

Using automatic gain control, the soundtrack music can be "ducked" so that it is quieter when there is speech or dialog in the

original video soundtrack. Thus the music is primarily heard when people are not speaking, and can cover background noise and other imperfections. Conversely, noisy audio from the camcorder microphone can be gain-expanded or noise-gated such that low-level noise is effectively muted.

Iterative Semi-automatic Music Video Construction

We are also working on an iterative semi-automatic approach to building the music videos. Given a new source video and separate audio file, the system will automatically perform the audio and video analysis, and output the most suitable video to the storyboard window of the Hitchcock user interface. At this point the user can reorder clips, replace them with other clips from the raw video, or lock particular clip boundaries. The process can be repeated until a satisfactory video is produced. This general framework may provide a simple and lightweight means by which users can efficiently construct music videos.

8. CONCLUSION

We have presented a music video creation system that can automatically select and align video segments to music. Because it produces multimedia output, we can only present static representations of the algorithm in this paper. We have included an example video produced by our approach in this submission for interested readers to view, at the URL given above. Informal judgements and user reactions show that our method produces convincing music videos. Each music video is comprised of the high quality portions of the clips from the source video. The video clip boundaries are aligned with musical transitions, such as soft/loud or verse/chorus changes in the soundtrack audio.

Drag-and-drop clip selection and ordering when combined with automatic music analysis substantially simplifies music video creation. This is important as more and more people start to use PCs to edit heir growing collections of raw video. Preliminary studies indicate that users find it easy to interact with the system and to create music videos [7]. At the same time, the study uncovered areas where we can improve the system's usability and functionality. For example, the source audio from the DV should play a significant role in determining clip boundaries. Future work will address this shortcoming and integrate additional features.

9. REFERENCES

[1] C. Bregler, M. Covell, and M. Slaney. "Video rewrite: Driving visual speech with audio." *Computer Graphics Annual Conference Series*, 1997

[2] Boreczky, J. and Rowe, L., "Comparison of Video Shot Boundary Detection Techniques," in Proc.SPIE Conference on Storage and Retrieval for Still Image and Video Databases IV, San Jose, CA, February, 1996, pp. 170-179

[3] Christel, M., Smith, M., Taylor, C. and Winkler, D., "Evolving Video Skims into Useful Multimedia Abstractions" in *CHI 98 Conference Proceedings* (Los Angeles, CA), New York: ACM, pp. 171-178, 1998.

[4] Cooper, M. and Foote, J., "Scene Boundary Detection Via Video Self-Similarity Analysis." Proc. IEEE Intl. Conf. on Image Processing, pp. 378-81, 2001.

[5] Foote, J., "Automatic Audio Segmentation using a Measure of Audio Novelty." in *Proc. of IEEE ICME*, vol. I, pp. 452-455, 2000.

[6] Foote, J., and Uchihashi, S.,"The Beat Spectrum: A New Approach to Rhythm Analysis," submitted to *ICME 2001*.

[7] A. Girgensohn, S. Bly, F. Shipman, J. Boreczky, and L. Wilcox. "Home Video Editing Made Easy — Balancing Automation and User Control." In *Human-Computer Interaction INTERACT '01*, IOS Press, pp. 464-471, 2001.

[8] Girgensohn, A., Boreczky, J., Chiu, P., Doherty, J., Foote., J., Golovchinsky, G., Uchihashi, S., and Wilcox, L. (2000), A Semi-Automatic Approach to Home Video Editing, in *UIST '00 Proceedings*, ACM Press, pp. 81-89.

[9] Goto, M. and Y. Muraoaka (1994). "A Beat Tracking System for Acoustic Signals of Music," In *Proc. ACM Multimedia 1994*, San Francisco, ACM.

[10] Kennedy, H.,"A Floydian Analysis of 'The Wizard of Oz'," in *The New York Daily News*, May 13, 1997, (also http://www.straightdope.com/mailbag/mdarkside.htm).

[11] J. Kruskal and D. Sankoff, "An Anthology of Algorithms and Concepts for Sequence Comparison," in *Time Warps, String Edits, and Macromolecules: the Theory and Practice of String Comparison*, eds. D. Sankoff and J. Kruskal, CSLI Publications, 1999

[12] Lienhart, R., "Abstracting Home Video Automatically," in *Proc. ACM Multimedia* '99 (Part 2), pp. 37-40, 1999.

[13] Lipscomb, S.D. (1997). "Perceptual measures of visual and auditory cues in film music." *JASA* **101**(5, ii), p. 3190 (online version at http://imr.utsa.edu/~lipscomb/JASA97/)

[14] Lipscomb, S.D. & Kendall R.A. "Perceptual judgment of the relationship between musical and visual components in film." *Psychomusicology*, **13**(1), pp. 60-98, (1994) (online version at http://imr.utsa.edu/~lipscomb/Thesis/thes00.html)

[15] MPEG Requirements Group. *Description of MPEG-7 Content Set*, Doc. ISO/MPEG N2467, MPEG Atlantic City Meeting, October 1998.

[16] muvee AutoProducer, http://www.muvee.com

[17] W. R. Neuman, "Beyond HDTV: Exploring Subjective Responses to Very High Definition Television"; *MIT Media Laboratory Report*, July, 1990.

[18] Pfeiffer, S., Lienhart, R., Fischer, S. and Effelsberg, W., "Abstracting Digital Movies Automatically," in *Journal of Visual Communication and Image Representation*, 7(4), pp. 345-353, December 1996.

[19] Sack, W., and Davis, M. "IDIC: Assembling Video Sequences from Story Plans and Content Annotations." In *Proc. IEEE International Conference on Multimedia Computing and Systems*. Boston, Ma., May 14 - 19, 1994.

[20] Scheirer, Eric D. (1998). "Tempo and Beat Analysis of Acoustic Musical Signals." In *J. Acoust. Soc. Am.* **103**(1) (Jan 1998), pp. 588-601.

[21] Smith, M. and Kanade, T., "Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques," in *Proc. Computer Vision and Pattern Recognition*, pp. 775-781, 1997.

[22] Suzuki, R. and Iwadate, Y., "Multimedia Montage--Counterpoint synthesis of movies," in *Proc. IEEE Multimedia Systems* '99, Vol. 1, pp. 433-438, 1999

[23] Van Trees, H., *Detection, Estimation, and Modulation Theory, Pt. 1*. J. Wiley and Sons, 1968.