CONTENT AND TRANSFORMATION EFFECT MATCHING FOR AUTOMATED HOME VIDEO EDITING

Xian-Sheng Hua, Hong-Jiang Zhang
Microsoft Research Asia
{xshua, hjzhang}@microsoft.com

ABSTRACT

While camcorders have become a commodity home appliance, few watch the recorded videos or share them with friends and relatives due to the difficulty of turning the raw footage into a compelling video story. Previous works on Automated Video Editing (AVE) developed an automatic solution for video content selection and video-music matching. However, to automatically apply video transformation effects, such as fast/slow motion, thresholding, binarization and watercolor, has not been solved or even addressed. In this paper, we proposed several automatic video effect and content matching schemes, which facilitate generating more compelling and interesting AVE results.

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.

Previous work on *Automated home Video Editing* (AVE) [1] tried to solve this issue by presenting a system that automates home video editing, in which near-professional results are created using a set of video and music content analysis algorithms. However, this system focused on video content selection and video-music matching. To generate more compelling video editing results, automatically applying transformation effects, such as fast/slow motion, thresholding, binarization and watercolor, on video clips is also a great help. This is the primary issue that will be addressed in this paper.

To automatically select appropriate video clips to apply these transformation effects, it is necessary to analyze that which types of video clips are suitable for these effects, as well as to analyze how to automatically detect these video clips and apply the effects on them. The rest of this paper is organized as follows. In Section 2, a brief description of AVE is introduced. Then, video content analysis for *content-effect* matching is represented Section 3. In Section 4, the principles and methods for *content-effect* matching is discussed. And last, experimental results are introduced in Section 5, followed by conclusion remarks and future work in Section 6.

2. AUTOMATED HOME VIDEO EDITING

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 [2], 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" or "informative" video segments that match video motion intensity with music tempo, as well as match shot boundaries with music beats and sentences in the audio track. The total length of the selected video segments may be determined either by the duration of the incidental music, or another desired value. This central stage is the primary and the most challenging one, which is formulated as a 0-1 programming system. The last stage is *Composition*, which renders the selected video segments with music by adding appropriate transitions between the selected video segments.

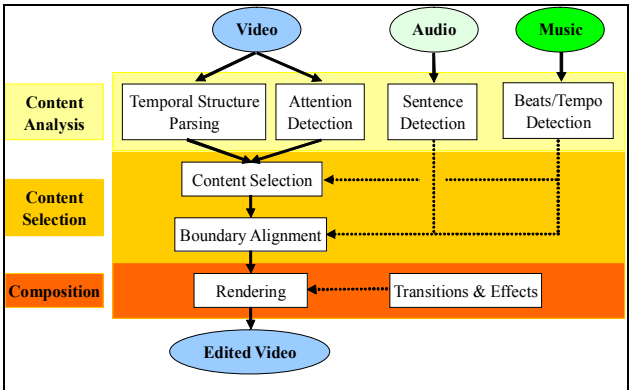

Figure 1. System Overview of AVE.

Due to limited shooting skills, shots in typical raw home videos are often long in duration compared with typical professionally edited video programs. These types of original shots often contain redundant information and boring sequences, as well as low quality frames. Consequently, AVE system segments video sequences during the temporal parsing process, and the "importance" index is computed for each sub-shots. Only the "best" sub-shots are used for the construction of the final video. For more details about AVE, please refer to literature [1].

In the final stage of AVE, transformation effects can be applied on selected video clips to make the edited video more compelling. However, how to automatically select appropriate effects for certain video clips are difficult and essential while it is not addressed in [1]. There are two primary issues for automatically applying transformation effects. One is to determine which kinds of video clips are suitable for certain transformation effects. For example, typically slow/fast motion is suitable for video clips that have dominant object motions, rather than near still clips or clips that have rather small moving objects. Another issue is how to applying the effects on selected video clips.

3. VIDEO CONTENT ANALYSIS

As mentioned above, there are two primary issues in automatically applying transformation effects on video clips. To solve these issues, in this section, a number of video content analysis algorithms are introduced, including camera motion detection, object motion detection, moving object detection, texture detection and contrast detection. It is necessary to mention here that all the below analyses are applied on a single video shot or sub-shot, obtained from the video structuring parsing procedure in AVE system [1].

3.1. Motion Detection

There are a number of camera motion and object motion detection algorithms in literatures [2]-[5]. In order to automatically apply video transformation effects such as slow/fast motion, we need to distinguish the following cases:

(1) Whether there is camera motion, and if yes, what the speed or intensity of the motion is.
(2) Whether there is dominant moving object, and if yes, what the speed or motion intensity of the object is.
(3) If there is a dominant moving object, whether it has a relatively "obvious" motion trajectory. This feature is applied to distinguish whether the primary motion is translation or deformation.

3.1.1. Camera Motion Detection

To answer the above questions, firstly a camera motion detector based on affine model is applied, which is similar to the method in [3], except that our algorithm works on motion vector fields instead of decompressed video frames. The output of this camera motion detector is a series of sub-segments of the shot or sub-shot with camera motion types and speeds, represented by

$$CM = \{(S_i, E_i, T_i, I_i), 0 \le i < m\} \quad (1)$$

where S_i and E_i are the start and end time of the camera motion sub-segment, T_i is the camera motion type, which may be *still*, *zoom in*, *zoom out*, *roll clockwise*, *roll anticlockwise*, *right*, *right up*, *up*, *left up*, *left*, *left down*, *down*, *right down*, and *others*. And I_i is the motion intensity or speed (normalized into [0, 1]) of the corresponding detected camera motion. Figure 2 shows the CM (camera motion) sequence of a 5-minute home video clip, which consists of four *still*, one *zoom in*, one *zoom out*, one *right* and two *left* sub-clips (note that for the last 2/3 part of this clip, the camera was fixed on a table, thus there is a long *still* sub-clip).

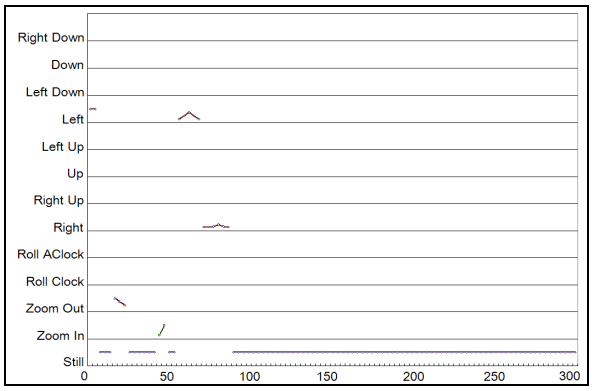

Figure 2. CM Sequence of a 5-minute home video clip (the horizontal axis represents the time in seconds, while the curves indicated the intensity of the camera motions).

3.1.2. Object Motion Detection

It is not easy to detect exact object motion if camera motion and object motion exist simultaneously. In this paper, the method in [2] is adopted to detect the salience maps of object motions. The OMS (*Object Motion Saliency*) maps are generated from motion vectors in MPEG streams by a temporal energy filter and a global motion filter. The OMS maps are sensitive to object motions, despite of whether there are camera motions or not [2]. For convenience, the OMS map sequences produced by the method in [2] are represented by

$$OM = \{OM_i, 0 \le i < n\} \quad (2)$$

where n is the number of OMS maps in the under-observation shot or sub-shot, and the OMS map is represented by

$$OM_i = \{p^i_{kl}, 0 \le k < K, 0 \le l < L\} \quad (3)$$

where K and L are the height and width of the OMS map, respectively, and p^i_{kl} is the intensity (normalized into [0, 1]) of the corresponding pixel (or the macro block in the original video frame). The object motion analysis in this paper is based on the above OMS map sequence. To be exact, two motion curves are derived from OMS image sequence, which will be used for content-effect matching (to be presented in Section 4). One is object motion intensity (OMI) curve; the other is object position variation (OPV) curve.

OMI curve, defined by the equation below, is used to represent the motion intensity of moving objects in the video.

$$OMI = \{OMI_i\} = \left\{ \left(\sum_{k=0,l=0}^{K-1,L-1} p^i_{kl} \right) / (KL) \right\} \quad (4)$$

OMI may consist of two components, including the shape variation (deformation) of the objects, and the coherent motion or position/location changes (translation) of the objects, while OPV only stands for the later one, which is defined by

$$OPV = \{OPV_i\} = \left\{ \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \right\} \quad (5)$$

where (x_i, y_i) is the (normalized) gravity center of the OMS image OM_i, defined by

$$x_i = \left(\sum_{k=0,l=0}^{K-1,L-1} p^i_{kl} x^i_{kl} \right) / \left(L \cdot \sum_{k=0,l=0}^{K-1,L-1} p^i_{kl} \right) \quad (6)$$

$$y_i = \left(\sum_{k=0,l=0}^{K-1,L-1} p^i_{kl} y^i_{kl} \right) / \left(K \cdot \sum_{k=0,l=0}^{K-1,L-1} p^i_{kl} \right) \quad (7)$$

where (x^i_{kl}, y^j_{kl}) is the coordinate of the corresponding pixel in the OMS image OM_i. Figure 3 shows the above two curves derived from the OMS image sequences of the 5-minute home video clip mentioned in Section 3.1.

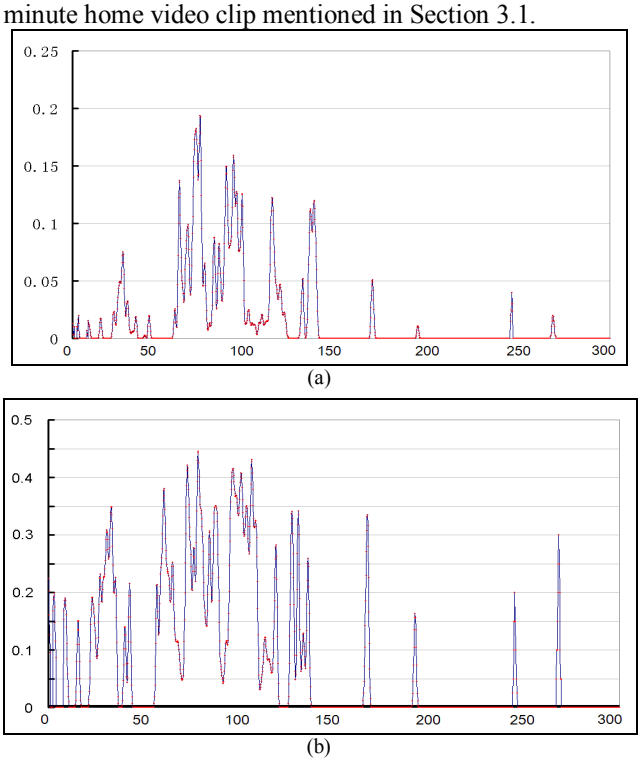

Figure 3. OMI (a) and OPV (b) curves of the sample video clip.

It should be mentioned here that OPV defined by equation (5) is a rough estimation of coherent motion intensity of moving objects. A more subtle estimation may be obtained by integrating object segmentation and tracking, as well as treating individual object separately, rather than consider them as one entity.

3.1.3. Moving Object Detection

In the application of this paper, we only need to detect whether there are moving objects. Therefore, the OMS map can also be used here to distinguish this case. Actually, OMI curve also reveals the possibility of existing moving objects.

3.2. Texture Detection

There are a number of texture detection algorithms in literature. In this paper, we adopt the method in [6], which is based on *grating cell operator* features obtained by a bank of Gabor Filters. Grating cell operator features are sensitive to textures but generally have no or few respond for non-texture areas. By applying the texture detector on sampled video frames (for example, I-frames), a curve representing the texture occupation ratio in each video frame is obtained, denoted by

$$TX = \{TX_i, 0 \le i < m\} \tag{8}$$

3.3. Contrast Detection

Contrast of a certain video frame i is defined by

$$CT_i = \left(H^i_{\max} - H^i_{\min}\right) / \left(H^i_{\max} + H^i_{\min}\right) \tag{9}$$

where H^i_{max} and H^i_{min} are the maximal and minimal non-zero illuminance in the thresholded grey histogram (i.e., the bins with relatively small values are set to zero) of video frame i, respectively. Thus, a curve representing the contrast of each video frame is obtained, denoted by

$$CT = \{CT_i, 0 \le i < m\} \tag{10}$$

4. CONTENT-EFFECT MATCHING

In this section, we discuss how to determine the suitability of applying certain transformation effect on a certain video segment, and how to apply the effects on the clips in AVE results based on the suitability curves.

4.1. Motion-Based Matching

In this sub-section, we discuss motion-related content-effect matching schemes, including slow motion, fast motion, and "magic moving" (to be explained later).

Observed from typical TV programs and films, slow motion is often applied to represent fast motions thus motion details are easy to be perceived by viewers. Slow motion is also frequently employed to represent strong emotional motions, romantics or similar cases. While fast motion often is used to speed up relatively long-existing object motions to decrease the showing time but not affect viewers' understanding. And, fast motion is also used to produce funny effects.

Fast or slow motion are typically applied on video clips that have continuing dominant moving objects, whether there are dominant, mild or no camera motions. Accordingly, the suitability of slow motion and fast motion can be roughly estimated by OMI curve. The average OMI value within a certain sub-shot indicates the suitability of applying slow/fast motion effect.

"Magic Moving" is a special effect, which represents the original video clip by a much shorter one, consisting of a series of sub-clips connected by cross-fade (a transition effect). There are two typical cases suitable for applying magic moving. One is to shorten a clip with continuing moving objects and obvious motion trajectories but no or mild camera motions. The other case is to represent a long and boring scene with no or mild camera motions. Accordingly, the suitability of magic moving for the former case is roughly estimated by OPV curve when T_i is equal to "still", or the motion camera motion intensity is lower than a threshold if T_i is not equal to "still". More precisely, it is estimated by (the overbars indicate that they are the average values of the corresponding indices within a certain segment):

$$\overline{OPV_i} \times \left(1 - \overline{I_i}\right) \tag{11}$$

For the second case, which is often used when there is a very long scene with no or mild camera motion, whether there are dominant object motions or not. Therefore, the condition of using this effect is: having a long still shot or a shot with very mild camera motions.

And the remaining issue for the above two types of magic moving is how to select appropriate sub-clips from the matched shot. To keep higher *information fidelity* in the edited video, clips centered by local maximums in OMI curve are selected.

4.2. Object-Based Matching

In this section, we discuss content-effect matching based on objects, including thresholding, binarization and watercolor.

Thresholding, binarization and watercolor are often used to produce funny or similar effects. However, if these transformation effects are applied on video clips that have no dominant objects or clips mainly consist of textures, or the contrast is too low, the transformed video clips will be too blurry or obscure to be perceived clearly. Furthermore, if there are few object motions, video clips after applying these kinds of effects will be difficult to be understood. Accordingly, to obtain better results, it is necessary to determine whether there are dominant moving objects with less textures while high contrast. That is, clips with high OMI and high CT while low TX are suitable for these three effects. And, among these kinds of clips, the ones with high OPV are even better for these effects. To be exact, the suitability for these three effects is estimated by (α is set to 0.25 here)

$$\overline{OMI_i} \times \overline{CT_i} \times \left(1 - \overline{TX_i}\right) + \alpha \overline{OPV_i} \qquad (12)$$

In fact, for watercolor effect, color entropy (i.e., the entropy of the quantized HSV color histogram [1]) is also a significant indicator. If we denote color entropy of a video frame as CE_i, a more precise suitability estimation for watercolor effect is

$$\overline{OMI_i} \times \overline{CT_i} \times \overline{CE_i} \times \left(1 - \overline{TX_i}\right) + \alpha \overline{OPV_i} \qquad (13)$$

Figure 4 shows the suitability curve of the first 5 sub-shots in the aforementioned sample video for slow/fast motion, magic moving (the first case), thresholding/binarization, and watercolor effects, which indicates sub-shot 3 is more suitable for magic moving and slow/fast motion, while sub-shot 5 is more suitable for thresholding, binarization, watercolor and slow/fast motion.

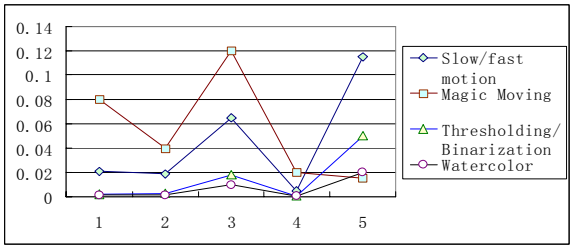

Figure 4. Effect suitability curves of the sample video clip.

Effect-content matching for AVE is implemented in a similar way as *motion-content matching* in [7], except it is based on the suitability curves, instead of the *suitability matrix* in [7]. The main idea of this algorithm is try to maximize the average effect suitability of selected sub-shots (on which certain effect will be applied), constrained by effect distribution uniformity, and the consistency of actual and desired appearing rates of the effects. This issue is then converted into a 0-1 programming problem and solved by conventional *Genetic Algorithm*.

5. EXPERIMENTS

In this section, performance evaluation for content and transformation effect matching is presented, as well as some automatic editing examples produced by the proposed scheme are provided.

5.1. Performance Evaluation

As it is not easy to objectively evaluate the performance of the content-effect matching results, we carry out a user study to compare the results matched by the proposed schemes with random results. Ten raw home videos (about 30 minutes each) and ten pieces of music (about 3 minutes each) are fed into the AVE system. Two sets of edited videos are produced, one integrates automatic content-effect matching, and the other applies effects randomly (the number of sub-shots applied with certain effect are identical for these two sets of results). So totally there are 20 edited videos.

Ten users are invited to provide a score between 0 and 5 to each edited video which indicates the satisfaction of the transformation effects applied on it. Evaluation results show that the satisfaction score for the results produced by the proposed matching scheme is much higher than those with random applied effects (3.9 vs. 2.3, about 70% higher). The main reason is that the random effect applying does not take content into consideration, thus in many cases the transformed video clips have worse perception effects, which is contrary to the original objective.

5.2. Editing Examples

To illustrate the differences between the results produced by automatic content-effect matching and random matching, several example clips are provided on the Internet for downloading (*http://research.microsoft.com/~xshua/cem*). For each transformation effect presented in this paper, four clips are provided, including *suitable original clip*, *suitable edited clip*, *unsuitable original clip*, and *unsuitable edited clip*. All the original clips are excerpted from the aforementioned 5-minute home video, with the highest or lowest suitability values.

6. CONCLUSION AND FUTURE WORK

In this paper, we addressed the issue of automatically applying transformation effects on automatically edited home videos, which makes AVE results more compelling and interesting. The principles and methods for applying several motion-based and object-based effects are investigated, based on a set of video content analysis algorithms. Actually, more transformation effects can be handled in a similar way. Furthermore, in this paper we mainly focused on studying the necessary conditions for applying the transformation effects, while relatively higher level semantics, which are relatively more difficult, are not discussed. A work moving one-step ahead may be integrating *mood detection* for the incidental music [1] and matching the music with the effects or effect parameters, which will be one of our future works.

7. REFERENCES

[1] X.S. Hua, L. Lu, and H.J. Zhang, "AVE – Automated Home Video Editing," *ACM Multimedia 2003*.

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

[3] D.J. Lan, Y.F. Ma, H.J. Zhang, "A Novel Motion-Based Representation For Video Mining," *ICME 2002*.

[4] R. Jin, Y. Qi, and A. Hauptmann, "A Probabilistic Model for Camera Zoom Detection," ICPR 2002.

[5] R. Milanese, F. Deguillaume, and A. Jacot-Descombes, "Video segmentation and camera motion characterization using compressed data," *Proc. SPIE Conf. on Multimedia Storage and Archiving Systems II*, Dallas (TX), 1997.

[6] S. E. Grigorescu, N. Petkov, and P. Kruizinga, "Comparison of Texture Features Based on Gabor Filters," *IEEE Trans. on Image Processing*, Vol. 11, No. 10, 2002.

[7] X.S. Hua, L. Lu, and H.J. Zhang, "Photo2Video," ACM Multimedia 2003.