FULLY AND SEMI-AUTOMATIC MUSIC SPORTS VIDEO COMPOSITION

Jinjun Wang[1,2]*, Engsiong Chng*[1]*, Changsheng Xu*[2]

[1] CeMNet, SCE, Nanyang Technological University, Singapore 639798
jjwang@pmail.ntu.edu.sg, aseschng@ntu.edu.sg
[2] Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613
xucs@i2r.a-star.edu.sg

ABSTRACT

Video composition is important for music video production. In this paper we propose an automatic method to assist the music sports video composition operation. Our approach is based on Dynamic Programming algorithm which finds a set of video shots that best matches the music. The method by default is fully-automatic, and users specification could be inserted to control the composition process, making it a semi-automatic system. This research has obvious importance to reduce manual processing, and enables the generation of high quality personalized music sports video. The proposed method is generic and fast. The experimental results are satisfactory.

1. INTRODUCTION

The sports broadcasting industry is becoming one of the most profitable business nowadays. The media is kept a-buzz, feeding the latest sports events to the hungry masses. At the same time, home users also want to re-edit existing video materials to produce personalized sports video segments. One example of such video is the music sports video (MSV) that is widely used for sports game highlight, match review, sports program prologue and episode, etc. However, current production of MSV is time-consuming for professional and difficult for amateur users because editing video and music clips requires tools, skills, experience and artistic talent. Hence, by introducing tools which can automate the video composition process, production efficiency can be greatly improved.

Unlike the general video editing/authoring research which focuses on video search/skimming, video visualization, metadata creation, non-linear editing, etc tasks, the MSV composition operation requires 1) semantic sports video analysis and 2) video/music matching. The domain of semantic sports video analysis has been extensively studied [1] for object feature extraction, sports video content augmentation and content analysis such as structuring [2], summary [3], highlight generation [4], etc. In this paper, we focus on the video/music matching portion of the MSV composition problem. Existing work or applications related video composition can be classified into 3 categories of 1) Mostly manual, e.g. Adobe Premiere [5], 2) Fully automatic, e.g. the AVE system [6] and muvee [7] software, and 3) Semi-automatic, e.g. the Hitchcock [8].

In this paper we propose an intellectual method to perform fully-automatic and semi-automatic MSV composition. The rest of the paper is organized as follows: Section 2 discusses the problems in MSV composition. Section 3 introduces our automatic MSV composition method. Experimental results are listed in section4, and section5 draws conclusions and raises some future work.

2. PROBLEM FORMATION

The MV is a short film meant to present a visual representation companying popular music songs. The MSV composition task picks desired video and music content from a content pool and multiplexes them to generate the output MSV. The selection and combination of video and music should satisfy professional video composition rules and/or personal preferences. For example, muvee [7] requires the users to specify a "style" option before it generates MVs.

In [9] we proposed a video-centric and music-centric MSV composition schemes. Each scheme has different video/music selection and matching requirements. In addition, supplementary composition criteria, e.g. the order of video content, may also be required by users. As the music-centric scheme is more complex than the video-centric scheme, we will use music-centric MSV composition as an example to introduce our algorithm. The algorithm can also be extended to video-centric composition problem as discussed later.

There are 6 common requirements for current music-centric MSV production:

Req.1: The video is divided into scenes and each scene is subdivided into shot. Neighboring shots are separated by "shot boundary". The music is partitioned into "semantic structures" [10], each semantic music structure has several "lyric" and each lyric covers several beats.

Req.2: To have better visual representation, if any scene is to be selected in the output MSV, it's better that all the shots in this scene are selected.

Req.3: A shot can be selected only once.

Req.4: The shot boundaries are aligned with the music beat boundaries. Users might specify different video contents for different semantic music portions. For example, using landscapes scene for the "Intro" and use excited player or coach scene for the "Chorus".

Req.5: It's unnecessary to display all the video content chronologically. However, it's better that shots in the same scene (event) are presented chronologically. The music is played from the beginning to the end.

Req.6: Users might exclude certain video contents from selection. Users might specify certain video contents to be displayed at a specified portion of the music.

3. AUTOMATIC VIDEO COMPOSITION

To solve the music-centric MSV composition problem, we introduce a generic method based on Dynamic Programming (DP) algorithm which, by default, performs fully-automatic video content selection and video/music matching. User interventions, e.g. those listed in Req.6 above can be inserted to control the MSV production process. Thus our system has both the fully and semi-automatic mode [8].

Prior to video composition, our previous work [9] is used to automatically identify scenes of soccer event/player(s)/team from user specified videos and generate a video content pool. Each scene consists of multiple shots with shot boundary information available. The input music is analyzed to extract the music boundaries information. Once the content pool is ready, the system can proceed to the video/music selection and matching step. Our algorithm is discussed below:

Given a sequence of video scenes $\boldsymbol{S} = [\boldsymbol{S}^1, \boldsymbol{S}^2, ... \boldsymbol{S}^L]$ with L scenes, each scene $\boldsymbol{S}^l = [\boldsymbol{s}_1^l, \boldsymbol{s}_2^l, ..., \boldsymbol{s}_{M_l}^l]$, $l = 1, .., L$ containing M_l shots. We have

$$\boldsymbol{S} = [[\boldsymbol{s}_1^1, ..., \boldsymbol{s}_{M_1}^1], ..., [\boldsymbol{s}_1^L, ..., \boldsymbol{s}_{M_L}^L]] \quad (1)$$

Each shot $\boldsymbol{s}$ records a set of attributes of the shot. Currently the attributes adopted by our system are

$$\boldsymbol{s} = [normalized\, duration,\ normalized\, motion] \quad (2)$$

The total number of shot in $\boldsymbol{S}$ is M where

$$M = \sum_{l=1}^{L} M_l \quad (3)$$

For simplicity, Eq.1 can be rewritten as

$$\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_M] \quad (4)$$

Given a sequence of music structure boundaries $\boldsymbol{B}$. Each structure boundary includes several lyric boundaries and each lyric boundary covers several beat boundaries. Suppose the total number of beats is N ($N \leq M$), we have

$$\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, ..., \boldsymbol{b}_N] \quad (5)$$

Each $\boldsymbol{b}$ records a set of attributes of the music beat. Currently the attributes adopted by our system are

$$\boldsymbol{b} = [normalized\, length,\ normalized\, tempo] \quad (6)$$

The music-centric video composition scheme selects N shots from $\boldsymbol{S}$ that best match the N beat boundaries in $\boldsymbol{B}$. The key point of our approach is the using of a 2-D grid matrix $\boldsymbol{G}$ to cache the intermediate matching result, M by N in size. Each component of $\boldsymbol{G}$ is denoted as $\boldsymbol{g}_{ij}$, $i = 1, ..., M$, $j = 1, ..., N$ and

$$\boldsymbol{g}_{ij} = [d_{ij}, \boldsymbol{p}_{ij}]^T \quad (7)$$

where d_{ij} is the Edit Distance (ED) value and $\boldsymbol{p}_{ij}$ is the position of $\boldsymbol{g}_{ij}$'s ancestor in $\boldsymbol{G}$. As $\boldsymbol{G}$ is a 2-D matrix, $\boldsymbol{p}_{ij} = [r_{ij}, c_{ij}]$ where r_{ij} and c_{ij} are the row and column values, respectively.

In DP algorithm, the selection of $\boldsymbol{p}_{ij}$ is based on the following criteria:

$$\boldsymbol{p}_{ij} = [r_{ij}, c_{ij}] = arg \min_{\substack{r=1,...,M \\ c=1,...,N}} (d_{rc} + T(\boldsymbol{g}_{rc}, \boldsymbol{g}_{ij})) \quad (8)$$

where $T()$ is the transition cost function and $T(\boldsymbol{g}_{rc}, \boldsymbol{g}_{ij})$ is the transition cost from grid $\boldsymbol{g}_{rc}$ to grid $\boldsymbol{g}_{ij}$. Different $T()$ function can be used in our system to satisfy different matching condition, which will be discussed later.

Then d_{ij} is computed as

$$d_{ij} = d_{r_{ij}c_{ij}} + T(\boldsymbol{g}_{r_{ij}c_{ij}}, \boldsymbol{g}_{ij}) + C(\boldsymbol{s}_i, \boldsymbol{b}_j) + d_{ij}^0 \quad (9)$$

where d_{ij}^0 is initialized to 0 (the user specified shot inclusion/exclusion requirements will change d_{ij}^0, which is discussed later), and $C()$ measures the similarity between shot $\boldsymbol{s}_i$ and music boundary $\boldsymbol{b}_j$ as

$$C(\boldsymbol{s}_i, \boldsymbol{b}_j) = ||\boldsymbol{s}_i - \boldsymbol{b}_j||^2 \quad (10)$$

To build $\boldsymbol{G}$, DP starts from the grids in the first column of $\boldsymbol{G}$, i.e. $\boldsymbol{g}_{i0}$, $i = 1, ..., M$. Particularly, $\boldsymbol{p}_{i0}$ is set to $[-1, -1]^T$ as it has no ancestor. So we have

$$\boldsymbol{g}_{i0} = [C(\boldsymbol{s}_i, \boldsymbol{b}_0), [-1, -1]]^T \quad (11)$$

Then j is increased to 1 to compute grids $\boldsymbol{g}_{i1}$ using Eq.8 and Eq.9. This matching process is repeated till $j = N$. The grid that has the minimal ED value in the last column is found and denoted as $\boldsymbol{g}_{\hat{i}N}$ which indicates the end of the best matching path. This best matching path can finally be found by tracing the chain of $\boldsymbol{g}_{\hat{i}N}$'s ancestors till the first column of $\boldsymbol{G}$.

Fig.1 shows a typical video/music matching sample. In this sample, 8 shots are to be selected from 7 soccer scenes (these scenes contain $\{5, 3, 3, 3, 4, 5, 4\}$ shots respectively as shown in the left-most column) and matched with 8 music beats (these beat boundaries belong to 3 structure boundaries, each contains $\{3, 4, 1\}$ beats respectively as shown in the top-most row). User specified requirements are **Req.3**: shot candidates for music structure 1 ($\boldsymbol{b}_1$ to $\boldsymbol{b}_3$) can only be from soccer scene 1, 2, 3 and 4, shot candidates for music structure

2 ($\boldsymbol{b}_4$ to $\boldsymbol{b}_7$) can only be from soccer scene 5, 6, 7, and shot candidates for music structure 3 ($\boldsymbol{b}_8$) can be from all 7 soccer scenes, and **Req.6**: shot 23 ($\boldsymbol{s}_{23}$) is excluded, shot 4 ($\boldsymbol{s}_4$) must be placed at music beat 2.

In this sample, as the system is under the semi-automatic mode (there are some user specifications), some initialization parameters are applied to Eq.8 and Eq.9 as discussed below:

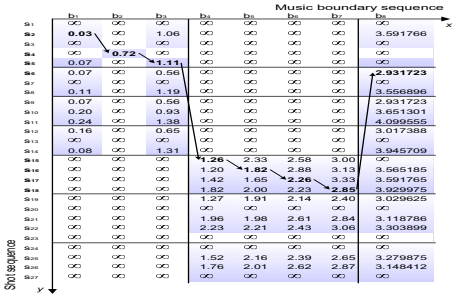

Fig. 1. A shot selection and video/music matching example

Req.1: This requirement is always satisfied because our algorithm represents the video/music using a hierarchical structure where "shot"/"beat" is the basic unit. (Eq.4 and Eq.5).

Req.5: To satisfy the requirement that shots in the same scene (event) are presented chronologically, in Eq.8, the choosing of grid $\boldsymbol{g}_{ij}$'s ancestor's row value r_{ij} must satisfy the following: If $\boldsymbol{s}_i$ and $\boldsymbol{s}_r$ belong to the same event, then

$$r_{ij} \in R_1 = \{r | r < i,\ \boldsymbol{s}_r \in \boldsymbol{S}_l \ and\ \boldsymbol{s}_i \in \boldsymbol{S}_l\} \tag{12}$$

If $\boldsymbol{s}_i$ and $\boldsymbol{s}_r$ belong to different events, then

$$r_{ij} \in R_2 = \{r | r \neq i,\ \boldsymbol{s}_r \in \boldsymbol{S}_l,\ \boldsymbol{s}_i \in \boldsymbol{S}_m\ and\ l \neq m\} \tag{13}$$

Hence we have

$$r_{ij} \in (R_1 \cap R_2) \tag{14}$$

To satisfy the requirement that music are played from the beginning to the end, in Eq.8, the column value of grid $\boldsymbol{g}_{ij}$'s ancestor c_{ij} must satisfy:

$$c_{ij} = j - 1 \tag{15}$$

Hence we have the following for Eq.8

$$\boldsymbol{p}_{ij} = [r_{ij}, c_{ij}] = arg \min_{\substack{r \in (R_1 \cap R_2) \\ c = j-1}} (d_{rc} + T(\boldsymbol{g}_{rc}, \boldsymbol{g}_{ij})) \tag{16}$$

Req.2: To satisfy the requirement that the shots in the same scene are better selected together, the transition cost function $T()$ (Eq.8) has the following form:

$$T(\boldsymbol{g}_{rc}, \boldsymbol{g}_{ij}) = F(Dist(r, i)) \tag{17}$$

where $Dist(r, i)$ is the distance between $\boldsymbol{s}_r$ and $\boldsymbol{s}_i$. To differentiate the distance for shots of the same scene from that of different scene, suppose the index of shot $\boldsymbol{s}_r$ in Eq.1 be shot $\hat{r}$ of scene l ($\hat{r} = 1, ..., M_l$), and $\boldsymbol{s}_i$ be shot $\hat{i}$ of scene m ($\hat{i} = 1, ..., M_m$), then

$$Dist(r, i) = \begin{cases} \hat{i} - \hat{r} & l = m \\ (M_l - \hat{r}) + \hat{i} & l \neq m \end{cases} \tag{18}$$

Then to decide the form of $F(x)$, since $F(0) = 0$ and $F(1) = 1$ (normalized), we choose

$$F(x) = Log_2(x + 1) \tag{19}$$

as **Req.2** prefers the system to transit from one shot to another shot of the same scene rather than transit to shots of another scene (so more shots from the same scene could be selected) hence a function with a bulgy shape curve is more suitable. Now Eq.17 becomes

$$T(\boldsymbol{g}_{rc}, \boldsymbol{g}_{ij}) = \begin{cases} Log_2(\hat{i} - \hat{r} + 1) & l = m \\ Log_2(M_l - \hat{r} + \hat{i} + 1) & l \neq m \end{cases} \tag{20}$$

Req.3: To satisfy the requirement that a shot is selected only once, in Eq.8 the choice of $\boldsymbol{g}_{ij}$'s ancestor can only be from those grids whose ancestor chain does not include any of $\{\boldsymbol{g}_{i0}, \boldsymbol{g}_{i1}, ..., \boldsymbol{g}_{ij-1}\}$.

Req.6: To exclude $\boldsymbol{s}_{23}$ from selection, we set

$$d^0_{ij} = \infty \quad i = 23,\ j = 1, ..., 8 \tag{21}$$

and then DP will never select $\boldsymbol{s}_{23}$ in the composition.

Similarly, to fix $\boldsymbol{s}_4$ to be used at beat $\boldsymbol{b}_2$, we set

$$d^0_{ij} = \infty \quad \begin{matrix} i = 1, ..., 3, 5, ..., 27,\ j = 2 \\ or\ i = 4,\ j = 1, 3, ..., 8 \end{matrix} \tag{22}$$

Req.4: Similar to the way to satisfy Req.6, we set

$$d^0_{ij} = \infty \quad \begin{matrix} i = 1, ..., 14,\ j = 4, ..., 7 \\ or\ i = 15, ..., 27,\ j = 1, ..., 3 \end{matrix} \tag{23}$$

With the above mentioned limitations for Eq.8 and Eq.9, the best matching result found by our algorithm is that shots $\{\boldsymbol{s}_2, \boldsymbol{s}_4, \boldsymbol{s}_5, \boldsymbol{s}_{15}, \boldsymbol{s}_{16}, \boldsymbol{s}_{17}, \boldsymbol{s}_{18}, \boldsymbol{s}_6\}$ be selected and matched with respective music beats. In Fig.1, the middle part matrix shows the values of d_{ij} ($i = 1, ..., 27$, $j = 1, ..., 8$) in $\boldsymbol{G}$. The best matching path are highlighted.

4. EXPERIMENTAL RESULTS

In the experiments, our video data set contains 7 World-Cup 2002 and 4 Euro-Cup 2004 soccer games videos, totally 16 hours. The music clips used are "Forca" by "Nelly Furtado", 220 seconds long, "Do I have to cry for you" by "Nick Carter", 217 seconds long, and "When you say nothing at all" by "Boyzone", 256 seconds long.

4.1. Objective measure

Various attributes have been proposed in literature to evaluate the performance of video/music matching. For example, Hua [6] introduced the *Distribution Uniformity* attribute to measure the uniformly of the selected shot's distribution. In our study, three measures are used, including ***Time*** which measures the time spent to come to a solution, ***Equality*** which measures the average matching degree, and ***Integrality*** which measures the integrality of the selected scenes. Specifically,

$$Equality = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(\frac{length\ of\ b_i}{dura.\ of\ \hat{s}_i})^2 + (\frac{tempo\ of\ b_i}{motion\ of\ \hat{s}_i})^2} \tag{24}$$

where $\hat{s}$ is the selected shot, and

$$Integrality = \frac{1}{C}\sum_{i=1}^{C}\frac{\hat{M}_i}{M_i} \tag{25}$$

where C is the number of selected scenes and $\hat{M}_i$ is the number of selected shots in scene S^i.

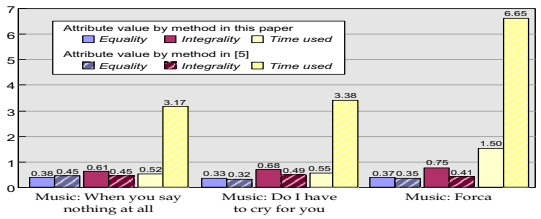

Fig. 2. Performance of our proposed method

Fig.2 illustrates the performance of our approach in comparison to [9] which used Greedy algorithm. The result shows that our approach enhances the *Integrality* attribute and significantly reduces the processing time. Another advantage of our algorithm is that the semi-automatic processing can be easily incorporated while in [9] such operation was implemented by using complicated *if-else* rules, hence production efficiency is improved.

4.2. Subjective measure

A user study is carried out to evaluate the quality of generated MSVs similar to that in [9]. The results show that the quality of generated MSV is satisfactory (Table.1 where the 5 scale corresponds to strongly accept 5 to strongly reject 1).

5. CONCLUSION AND FUTURE WORK

In this paper we propose a method to assist MSV generation. The method by default is fully-automatic. Users could also manually specify the desired video segments to be selected, and/or amend the default video composition rules to get more satisfactory MSV output. The method can also be used to generate video-centric MSV because the two major requirements by video-centric MSV generation, i.e. the video-centric MSV uses the scenes chronologically, and the music are background music without alignment to shots boundary, can be easily satisfied by the proposed algorithm.

Table 1. Soccer MTV Quality

Criteria	When you...	Do I...	Forca
Clarity	4.33	4	4.66
Conciseness	4.33	4	4.33
Coherence	4.33	4.33	4
Overall Qlty.	4.33	4	4.66

This research has obvious importance to reduce manual processing, and furthermore enable the generation of personalized MSV. The generated video material can be used for customized sports video summarization, highlight generation, sports MTV production, etc tasks.

6. REFERENCES

[1] N Adami, et al, "An overview of multi-modal techniques for the characterization of sport programmes," *Proc. of SPIE-VCIP'03*, pp. 1296–1306, 2003.

[2] Lexing Xie, et al, "Unsupervised discovery of multilevel statistical video structures using hierarchical hidden markov models," *Proc. of IEEE ICME'03*, 2003.

[3] N. Nitta, and N. Babaguchi, "Automatic story segmentation of closed-caption text for semantic content analysis of broadcasted sports video," *Proc. of MIS'02*, pp. 110–116, 2002.

[4] Y. Rui, et al, "Automatically extracting highlights for tv baseball programs," *Proc. of ACM MultiMedia'02*, pp. 105–115, 2002.

[5] Adobe Systems Inc., "Adobe premiere^{TM}," 2005.

[6] Xiansheng Hua, et al, "Ave: automated home video editing," *Proc. of ACM MultiMedia'03*, pp. 490–497, 2003.

[7] MuVee Technologies Pte. Ltd, "Muvee^{TM}," 2000.

[8] A. Girgensohn, et al, "A semi-automatic approach to home video editing," *Proc. of UIST'00, ACM Press*, pp. 81–89, 2000.

[9] Jinjun Wang, et al, "Automatic generation of personalized music sports video," *Proc of ACM MultiMedia'05*, pp. 31–38, 2005.

[10] Namunu Maddage, et al, "Content-based music structure analysis with applications to music semantics understanding," *Proc. of ACM MultiMedia'04*, pp. 112–119, 2004.