Sequential Pattern Based Temporal Contour Representations for Content-Based Multimedia Timeline Analysis

Gang Ren[1], Joseph Johnson[2], Hyunhwan Lee[2], Mitsunori Ogihara[1,3]
[1] Center for Computational Science, University of Miami, Coral Gables, FL 33146
[2] Marketing Department, School of Business Administration, University of Miami, Coral Gables, FL 33146
[3] Department of Computer Science, University of Miami, Coral Gables, FL 33124

Abstract— **Temporal contour shapes are closely linked to the narrative structure of multimedia content and provide important reference points in content-based multimedia timeline analysis. In this paper, multimedia timeline is extracted from content as time varying video and audio signal features. A temporal contour representation is implemented based on sequential pattern discovery algorithm for modeling the variation contours of multimedia features. The proposed contour representation extracts repetitive temporal patterns from a hierarchy of time resolutions or from synchronized video/audio feature dimensions. The statistically significant contour components, depicting the dominant timeline shapes, are utilized as a structural or analytical representation of the timeline. The modeling performance of this proposed temporal modeling framework is demonstrated through empirical validation and subjective evaluations.**

Keywords— sequential pattern; multimedia structure analysis; multimedia signal processing; contour representations

I. INTRODUCTION

Multimedia timeline analysis is an important aspect in the design and production of multimedia commercials [1,2]. Multimedia timeline includes the temporal allocations of multimodal objects in both the video and audio dimensions. An example is the time arrangements of video color patterns and the audio event density. The temporal allocations of these video or audio objects form the temporal narrative timeline of multimedia content. Fig. 1 shows two temporal timelines for the video chroma feature and the audio onset density feature. Each time point in this analysis corresponds to 1/50 of the complete video length. The video chroma feature is obtained by measuring the average chroma of the video frames in this time segment. Chroma is the measurement of the dominant color, as will be detailed in Sec. II. A. The audio onset density is obtained by measuring the number of significant audio onsets, which occur at the time locations where a dramatic change of the frequency-domain energy content is observed. For example, time points with dense percussive instruments or frequent changes of music notes yield a higher onset density value. These two feature curves in Fig. 1 show the timelines of conventional arc shapes, similar to the expectation-resolution pattern illustrated in [1]. In practice, multimedia timelines usually show overlapped patterns and an in-depth analysis is essential for both content creation and information retrieval applications [3].

In this paper, we present a sequential pattern based approach for analyzing multimedia timelines. Our method progresses through three stages. First, the multimedia timelines are transformed into numerical contour sequences whose values denote the temporal shapes of the timeline such as ascending slope, flat, or descending slope. Next, the common subsequences of these numerical sequences are extracted using a sequential pattern discovery algorithm [4, 5]. These common subsequences, or sequential patterns, are employed as the representation of the temporal contours in multimedia timeline. For example, the arc shape contour as illustrated in Fig. 1 is one of the common temporal contours extracted from a comparison of multiple timelines extracted from an ensemble of multimedia files. These common contours serve as shape descriptors for analyzing multimedia timelines. Finally, a sequential pattern alignment algorithm is implemented that match back these common contours to the timeline profiles as the annotation tags and also as the link points for comparing multiple timelines [6].

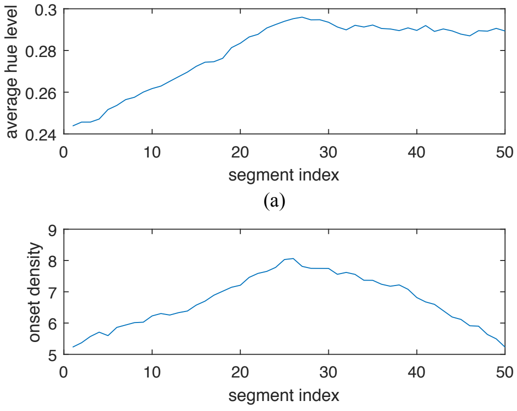
The use of temporal contours for multimedia timeline analysis are prevalent in both media creation and scholarly studies [7, 8]. Because simple temporal shapes are easy to recognize and memorize, creative artists, digital humanity scholars and marketing researchers utilize them extensively in their workflows. The arc shape illustrated in Fig. 1 is a specific contour type that often occurs in multimedia timeline analysis and each study domain developed its unique interpretations [7, 8]. The works in [1, 8] are based on a manual analysis approach, where human analysts inpsect the timelines and elicit recurring patterns as the graphical features for analysis. This manual approach is inefficient when analyzing a large amount of multimedia feature data or when conducting research on complex dependency patterns between multiple feature dimensions, e.g., a comparison of a video-induced timeline and an audio-induced timeline to study their synchronization.

For compuational implementations, a related time series modeling technique on temporal shape representation for time series object recognition is presented in [9, 10]. This technique identifies discriminative sub-sequences for classification application. Our proposed framework approaches sequence analysis problems in a complementary aspect by identifying sub-sequences (contour fragments) with maximum repetition between source timelines, instead of maximum algorithmic classification performance, to identify the dominant patterns. Applying the framework in [11], our proposed framework can also be generalized to classification problems similar to the implementation in [9, 10]. The methods in [9,10] is limited to predictive analysis because the significance quantification of subseqences relies on the validation from know class labels. Our proposed analysis framework overcomes these technical difficulties by automatically extracting common contours from

978-1-5090-6167-9/16 $31.00 © 2016 IEEE
DOI 10.1109/ICMLA.2016.159

multimedia timelines and employs these common contours as graphical "vocabulary" for timeline analysis. Our proposed framework provides statistical metrics for quantifying the significance of the extracted temporal contours. Furthermore, this computational framework allows the researchers to extend existing methods to larger ensembles of multimedia files or to more exploratory tasks such as analyzing complex dependency structures across multiple feature dimensions, as will be illustrated in the empirical studies in Sec. IV.

The rest of the paper is organized as follows. Sec. II illustrates the multimedia feature extraction process that forms the multimedia timelines from video files. The timeline analysis and "shape" contour representation are also illustrated. Sec. III covers the pattern analysis framework using sequential pattern discovery algorithm and the interpretation of contour representations. Sec. IV presents empirical studies based on several multimedia analytic applications. Sec. V provides a brief conclusion.



(a)

(b)

Fig. 1 Two examples of multimeida timeline extracted from content. A video files is divided into 50 equal-length segments. Then video/audio features are extracted from each segments to model the narrative timeline. (a) is the chroma (color) feature and (b) is the audio onset density feature.

II. Multimedia Timeline Feature Extraction

Multimedia features are extracted from both the video and the audio components of each multimedia file. These features are based on each video frames or audio segments and they form a multimedia timeline for contour analysis.

A. Video Feature Extraction

Video features are measured from the *HSV* color space representation extracted from each video frame [12]. In our implementation, one video frame is sampled from every ten video frames for feature extraction and these features measure the temporal evolution of the video scenes in the timeline.

The chroma profile measures the dominant color in the video scene. The term *chroma* refers to the different frequency (or wavelength) of the color. The "H" (hue) dimension of the HSV values measures the chroma of the pixels. The "H" values in our implementation range from [0, 1], with a circular arrangement as in Fig. 2. Red is denoted as "0". The "H" value increases as it goes through yellow, green, blue, and purple (0.75), then as the value approaches "1", the color goes back to red.

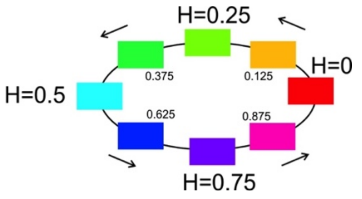


Fig. 2 Illustrations of chroma features. The chroma values form a circular motion from red to purple, as the value range [0, 1).

The "S" dimension of the HSV values measures the saturation profile. The term *saturation* refers to the "colorfulness" of a pixel. These values are in the data range of [0, 1]. A value of "0" means greyscale. A higher value means more color is added in.

The intensity profile measures the brightness of the video scene. The term *intensity* refers to the amount of light emitted from the pixels. In our implementation, the "V" dimension of intensity is measured on a [0, 1] scale, where "0" means no light emission and "1" means maximum light emission.

Each video frame is also segmented into smaller parts to track the localized image patterns and the spatial variations of the video features. In our implementation, video features are also extracted from smaller sub-frames and their differences are calculated to depict their contrasts.

B. Audio Feature Extraction

The audio features are based on the time-frequency analysis of audio signals [13]. The dynamics profile measures the energy distribution in a transformed time-frequency analytic space. The human auditory system has different response sensitivity to audio components at different frequencies [14]. This feature dimension is calculated from applying an auditory model to the audio spectrogram and then normalized as the ratio between the average value and the peak value of the auditory loudness curve.

The onset density profile is calculated as the number of audio onset in an audio segment divided by the length of that audio segment. In our implementation, the audio is first segmented into short clips (e.g., 5 seconds). We apply the onset detection algorithm in [15] and calculate the number of onsets and sum the onset strength in each short clip. The onset density and accumulated onset strength depicts the intensity of audio actions in each time segment. For example, the time segment with a quiet scene is usually connected with a lower number of audio onset density and accumulated onset strength.

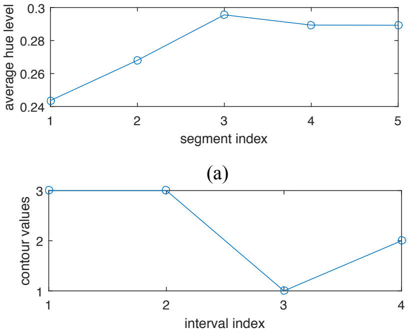
The timbre profile depicts the energy distribution in frequency and its evolution over time. The timbre descriptors in our application are slightly different from the timbre descriptors for musical signal analysis because the audio tracks in general audio materials are typically mixed narratives and music. Thus robust fundamental frequency estimation is not always possible. The timbre centroid is calculated as the weight center of the frequency-domain distribution of energy [15]. The timbre width is calculated as the frequency point below which 80% of the energy is contained. In our implementation, both the timbre centroid and timbre width are calculated for each 20 ms short time frame.

C. *Modeling Timeline Contours*

The timeline contours are modeled by its “shape” element of ascending, hold, and descending. For each time step in a source timeline sequence, the step distance between successive time frames is compared to a step threshold to categorize this step into the three contour shape elements. In this implementation, the step threshold is calculated as 10% of the standard deviation of the source timeline sequence. A step size within this step threshold is categorized as a “hold” step (flat trajectory) to exclude the nuance variations. Step sizes larger than this step threshold in the up and down directions are categorized as the ascending and descending steps correspondingly. The descending steps, the hold steps, and the ascending steps are then assigned the quantized “shape” values of “1”, “2”, and “3” respectively.

The quantized shape values form a source contour sequence as the source data for subsequent analysis using seqential pattern discovery algorithm. Fig.3 shows a simple timeline shape and its corresponding source contour sequence. This timeline is taken from the onset density trajectory of a TV commercial, where onset density is measured from five equal-length partitions of the whole TV commercial. The circled dots (feature points) in Fig.3(a) shows the onset density measured from each audio segments. The step types and their corresponding quantized shape values form the contour sequence in (b). These quantized shape values correspond to the four intervals between segments are formatted as a time-stamped itemset sequence as shown in (c). Each time step begins with a time stamp, such as “<2>”. Then it includes a shape value such as “1”. A location indicator “-1” terminates every step. The complete time sequence then terminates with “-1 -2”. We note here that the time stamps in Fig. 3(c) starts with “<0>”, thus offset by 1 to the contour values in Fig. 3(b).

Fig. 4 shows a slightly longer timeline sequence in (a), with its contour sequence in (b). For these longer sequences, we can apply a moving sampling template to chop them into multiple short sequences as in (c), with each short sequence covering five interval points in (b). This template sampling process allow us to model the local behavior of the timeline in a higher time resolution.



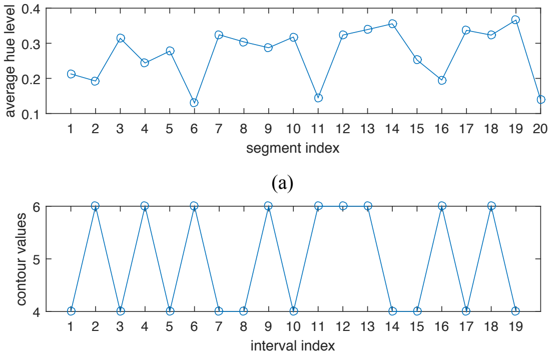
(a)

(b)

```
<0> 3 -1 <1> 3 -1 <2> 1 -1 <3> 2 -1 -2
```

(c)

Fig. 3 Examples forming a contour sequence from a timeline. The differences between adjacent steps of timeline sequence (a) is quantized as the contour sequence (b). The itemset notation of contour sequence is shown in (c**).**



(a)

(b)

```
<0> 4 -1 <1> 6 -1 <2> 4 -1 <3> 6 -1 <4> 4 -1 -2
<0> 6 -1 <1> 4 -1 <2> 4 -1 <3> 6 -1 <4> 4 -1 -2
<0> 6 -1 <1> 6 -1 <2> 6 -1 <3> 4 -1 <4> 4 -1 -2
```

(c)

Fig. 4 Example of a longer timeline sequence (a) and its contour sequence (b). The itemset sequence in (c) is obtained by applying a moving sampling template template over the contour sequence.

D. *Timeline Contour Comparison Between Different Resolutions and Feature Dimensions*

Most multimedia timelines can be extracted from multiple resolution levels. For example, video chroma value can be read from each sampled video frames. Then these sampled video frames can be aggregated into time sections to measure the average chroma within each time section. In our implementation, the default feature resolutions are obtained from split ratios of 5, 20, and 50 (i.e., from the lowest time resolution to the highest time resolution). At the lowest resolution level, a split ratio of 5 means that we split the complete video into five equal-length sections and measure this feature for each sections. For chroma feature, this means that the feature value for each section is measured as the average value of the chroma measurement of each sampled video frame inside this section. Then we obtain timelines with higher resolution by splitting the video into more segments, configured as 20 and 50 segments. For certain higher time resolution, the sampled video frames may not be evenly distributed into each time segment. In this case, we can either increase the number of the sampled video frames or extract feature from each available frames before resampling the feature sequence to the target resolution.

For the timeline of audio onset density feature, the feature values from the lower resolutions are more robust because each audio segment includes a sufficiently large number of audio onset. The feature values from the higher resolution shows more local behaviors such as the acceleration or deceleration of the beat pattern for percussive instruments. However, as the resolution increases, the measurement robustness decreases because a sufficiently large number of onset is not always included in each segment. Thus this specific feature dimension is different from the frame-based video feature in its random characteristics at higher resolution levels.

When comparing timeline sequences from different resolution levels, the timeline sequences at lower resolutions are first interpolated to the highest resolutions and aligned in time. This interpolation process allows a detailed investigation of the synchronization of multimedia events at multiple levels. We also implemented a mechanism that uses a moving template for sampling out a section of the multi-resolution timeline for analysis. The multi-resolution timeline is then quantized into contour values using the same thresholding process as in Sec. II. C. To compare the contour values from different resolutions, the resolution index I_r is embeded into the shape values as:

$$S' = S + 3(I_r - 1),\ I_r = 1, 2, 3 \tag{1}$$

where S is the contour values. Using this process, the contour values from the lowest resolution are still “1”, “2”, and “3”. The values from a higher resolution are transformed to “4”, “5”, and “6”. The transformed shape values are concatenated to form the combined source timeline sequence. The feature values from different resolution levels are combined into the same time steps, and the resolution levels corresponding to each feature token can be calculated from its value range.

A similar feature value transformation is implemented for comparing feature values form different feature dimensions by further embedding a dimension index I_d into the contour value as:

$$S'' = S + 3(I_r - 1) + 9(I_d - 1),\ I_d = 1, 2, \cdots \tag{2}$$

This implementation assigns different data ranges to different feature dimensions, assuming that each feature dimension includes three resolutions and each contour value has three possible levels. A general resolution/dimensionality embedding formula is:

$$S'' = S + n_s(I_r - 1) + n_s n_r(I_d - 1) \tag{3}$$

where S is the “internal” contour values within one resolution and within one dimension. n_s is the number of values. n_r is the number of resolutions. $I_r = 1, 2, \cdots$ is the index for resolution levels and $I_d = 1, 2, \cdots$ is the index for feature dimensions. Because the contour values from different resolution levels and different feature dimensions are allocated into different data ranges, the timelines from different resolutions or dimensions can be concatenated into a combined source timeline sequence for further analysis.

III. Sequential Pattern Based Temporal Contour Analysis

A. Sequential Pattern Discovery

Each contour sequence is represented as an itemset format:

$$S_u = \langle S_{u,1}, S_{u,2}, \cdots, S_{u,L_u} \rangle \tag{4}$$

where u is the index for each video, $S_{u,i}$ is an itemset that includes the multimedia features of the i-th temporal step of video u. L_u is the total temporal steps of source contour sequence S_u. Each itemset can be represented as:

$$S_{u,i} = \left(X_{u,i,1}, X_{u,i,2}, \cdots, X_{u,i,M_{u,i}}\right) \tag{5}$$

$X_{u,i,m}$ is the contour variable of feature m in the i-th step of the source contour sequence of video u. $M_{u,i}$ is the total number of contour variable in the i-the step of video $u's$ contour sequence. These contour sequences are aligned with each other to identify their overlapping sub-sequences.

The time stamps for source timeline sequence $S_u$is also tracked in the alignment process:

$$T_u = \langle T_{u,1}, T_{u,2}, \cdots, T_{u,L_f} \rangle \tag{6}$$

where $T_{u,i}$ is the time stamp for the i-th step of video $u's$ feature sequence. These time stamps locate the sequential patterns at the timeline for multimedia annotations and subsequent subjective validation experiments.

Sequential pattern is identified using a search process [4,5], which finds the overlapping subsequences among a group of source contour sequences $S_1, S_2, \cdots, S_V$, where V is the total number of videos. A sequential pattern P_q is denoted as:

$$P_q = \langle P_{q,1}, P_{q,2}, \cdots, P_{q,N_q} \rangle \tag{7}$$

q is the index for sequential pattern among all sequential patterns identified. $P_{q,n}$ is an itemset whose elements (items) are contour variables. N_q is the total number of itemsets in sequential pattern q, which is also called the length of sequential pattern q. This sequential pattern is a common subsequence of a group of contour sequences, i.e. The itemsets in this sequential pattern satisfies the following relationship with contour sequences $S_{SUP(1)}, S_{SUP(2)}, \cdots, S_{SUP(p)}, \cdots, S_{SUP(P)}$ for a list of integer indices $j_1, j_2, \cdots, j_{N_q}$

$$\begin{gathered} 1 \leq j_1 < j_2 < \cdots < j_{N_q} \leq L_{SUP(p)} \\ P_{q,1} \subseteq S_{SUP(p),\, j_1} \\ P_{q,2} \subseteq S_{SUP(p), j_2} \\ \cdots \\ P_{q,N_q} \subseteq S_{SUP(p), j_{N_q}} \end{gathered} \tag{8}$$

where $L_{SUP(p)}$ is the total number of steps in the source sequence $SUP(p)$. The indices $SUP(1), SUP(2), \cdots, SUP(P)$ is a selection of source timeline sequences. This group of source timeline sequences is called the support of the sequential pattern P_q. In other words, the sequential pattern P_q is contained in all source timeline sequences of its support.

B. Parameter Settings

The sequential pattern mining algorithm in [4,5] has implemented the options for setting several constraints in the sequential pattern mining process including minimum support threshold and gap control. The minimum support threshold η_{min} mandates the sequential patterns to have at least $\eta_{min} \cdot U$ supporting source sequences, where U is the total number of source sequences. When a smaller η_{min} is specified, a larger number of sequential patterns are admissible because subsequences with less repetition can qualify.

In [4, 5], sequential patterns from non-consecutive itemsets in the source sequences can be admitted by specifying several interval/gap-control constraints:

- **Minimum Successive Time Interval** is the minimum temporal steps (itemset gaps) between successive itemsets in the sequential pattern. In our implementation, this parameter is set to “1”, meaning no gap between itemsets is compulsory.
- **Maximum Successive Time Interval** is the maximum temporal steps (itemset gaps) allowed between successive itemsets in the sequential pattern. This parameter ensures that excessive amount of gaps

are not admitted. Setting this parameter to a larger number (more allowable gaps) poses a more relaxing constraint and thus admits more sequential patterns. However, if more gaps are allowed (the itemsets in the gap location is arbitrary in the source sequences), the supporting source sequences are aligned more incompletely, thus have less contextual relevance.

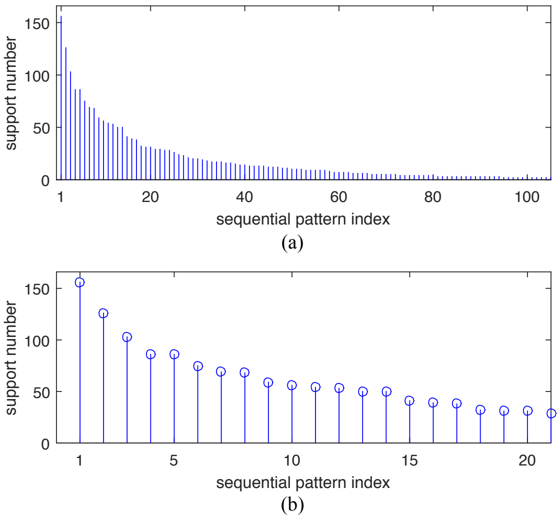
- **Minimum Total Time Span** is the minimum temporal steps between the first itemset and the last itemset. This parameter controls the length of admissible sequential patterns.
- **Maximum Total Time Span** is the maximum temporal steps between the first itemset and the last itemset. This parameter, in combination with the previous three parameters, controls the temporal distribution of itemsets and gaps in the admissible sequential patterns.

The combination of these parameters allows flexible control of the sequential motif structure. For example, setting these four parameters to (1, 1, 3, 3) admits length-four consecutive patterns, while (1, 2, 2, 3) configuration admits patterns with one gap in the middle. Admitting gaps significantly relaxes the constraints in the sequential pattern mining process and results in a larger number of sequential patterns for the same minimum support setting. This provides enhanced flexibility for our proposed analysis because more overlapping sequential patterns ensure more robust comparison between subsequences.

C. *Temporal Contour Analysis and Interpretations*

Each sequential pattern discovered shows a repeating pattern in the contour sequences. The significance of each sequential pattern is denoted by its support number, which is the number of source contour sequences that include this sequential pattern. In the graphical domain, the sequential pattern corresponds to a temporal contour fragment that embeds into several timelines. Fig. 5 shows the distribution of the support numbers of the discovered sequential patterns. The support numbers are organized in descending order, where sequential patterns with larger support numbers are shown on the left. These sequential patterns on the left side corresponds to the contour fragments that appear most often in the timelines thus they are most significant. In our analysis, we pick the left 20% portion as the salient temporal contours for subsequent analysis. We note here that the total support number in Fig. 5 is larger than the number of the source timeline sequences because a source timeline could contain multiple temporal contours.

Each sequential pattern can be represented as a contour sequence as in Fig. 6 (a) and (b). Then Fig. 6(c) shows a synthesized timeline from Fig. 6(b). This synthesized timeline starts from zero feature values and jump one unit up if the contour value in (b) indicates ascending (“3”), jump one unit down for descending contour value (“1”), and hold the current value for flat contour value (“2”). Fig. 7 shows a sequential pattern with a gap in step 3 (time stamp “<2>”). Here this gap permits all possible contour values, which form the branches at this time step.



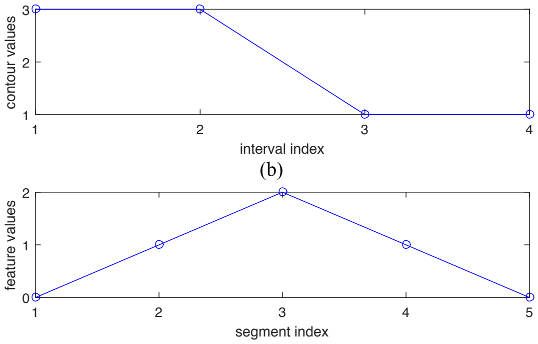
(a)

(b)

Fig. 5 Distribution of the support numbers for sequential patterns. The support numbers corresponding to 105 sequential patterns discovered from the contour sequences of 200 onset density timelines are shown in (a). The support numbers of the sequential patterns with top 20% support numbers are shown in (b).

```
<0> 3 -1 <1> 3 -1 <2> 1 -1 <3> 1 -1   #SUP: 58
```

(a)



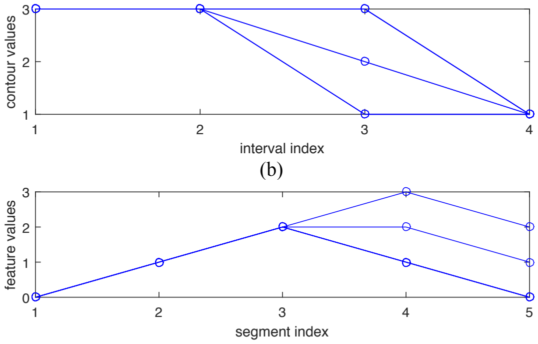
(b)

(c)

Fig. 6 Example of a contour sequence (a), its graphical representation (b), and the “synthesized” timeline (c).

```
<0> 3 -1 <1> 3 -1 <3> 1 -1   #SUP: 98
```

(a)



(b)

(c)

Fig. 7 Example of a contour sequence with gap value.

IV. Empirical Studies

The proposed multimedia timeline analysis framework is applied to the analysis of TV commercials' narrative timeline. We implemented a video dataset of 200 TV commercials and extract multiple video and audio features from segments of the TV commercials. For each TV commercial, we split the complete duration into 5, 20, and 50 segments to form a hierarchy of temporal resolutions. A feature point is then extracted from each segment for each feature dimension. The following empirical study is mainly applied to the audio onset density feature and the video chroma feature because of the page limit, however, the analysis of other feature dimensions can be performed in a similar manner.

Fig. 8 shows the temporal timelines extracted from the TV commercials with a time resolution of five splits per video for the feature dimension of audio onset density. From these timelines, we can see several familiar time contours such as rise-peak-fall pattern. Fig. 9 shows the contour representation for these timelines, where the interval index means the contrast points between two time segments in the timelines. Fig. 10 shows the resynthesized timelines from the contour representations using the method covered in Sec. III. C. We can observe here that these resynthesized timelines faithfully capture the dynamic patterns of the source timeline sequences in Fig. 8.

In Fig. 11 through Fig. 13 we overlap the time trajectories of 200 timelines extracted from 200 videos to show their overlapping patterns. The plotting lines include a small vertical offset to show their overall patterns. The source timelines are shown in Fig. 11, where we see a rough arc pattern. This pattern is more evident in Fig. 12, where alternating decrease ("10" for onset feature dimension) and increase ("12") are dominant. The synthesized patterns in Fig. 13 show several arc patterns but the center arc pattern is most dominant.

The sequential pattern analysis results show similar patterns as we observed from the overlapped source sequence visualizations. Fig. 14 shows the contour representation and synthesized sequences in (a) for the identified sequential pattern sequences (b). These sequences have the top five support numbers and thus most significant. We see the familiar arc pattern in the first two sequential patterns, yet more complex patterns in the other three sequential patterns.

Fig. 15 and Fig. 16 show the patterns from the most significant length-three and length-four patterns. These patterns are too complex to discover from a manual inspection of Fig.11 through Fig. 13. Employing the proposed analysis framework, these patterns enable the human analyst to expand the analysis scope to more complex patterns or to larger datasets. Also using alignment algorithm [6], these sequential patterns can be mapped back to the multimedia content for a comparison with manual analysis results.

Subjective rating experiments are also conducted on the discovered sequential patterns in Fig. 14 through Fig. 16. For each sequential pattern, its time location is mapped back to the source video content and this video segment is utilized as the "sonification" of this pattern. Because each sequential pattern can be mapped back to multiple videos, only the first video in the dataset, starting at video 1, is selected. Each rating target is only rated once. These simplifications are necessary because subjective ratings of high level multimedia patterns require a large volume of repetitive and intensive studies. For each sequential pattern, a human analyst first assigns a significance value on how dominant the "shape" pattern is. The five patterns in Fig. 14 were assigned significance scores of "6", "6", "2", "2", "4" in Likert scale, where "7" means this pattern can be readily perceived in the audio or video an "0" means no meaningful pattern. Fig. 17 also shows a similarity rating between contours, where "7" means most similar and "1" means maximum dissimilar. The five patterns in Fig. 15 were assigned significance scores of "4", "4", "4", "6", "2". The contour similarity rating is included in Fig. 18. These subjective ratings show that simpler patterns and arc-type patterns are assigned higher significance scores. The similarity ratings also show that human raters have more discerning power over these simpler patterns or arc-type patterns. Subjective ratings collected for multi-resolution patterns and multi-dimensional patterns show less pattern significance and mean similarity comparing to Fig. 17 and Fig. 18.

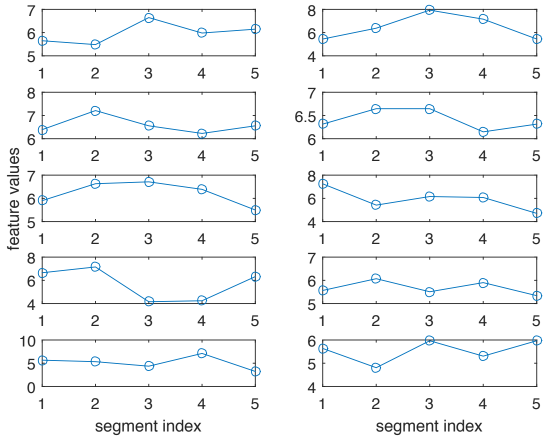


Fig. 8 The timeline extracted from ten videos. The feature dimension of audio onset density is applied.

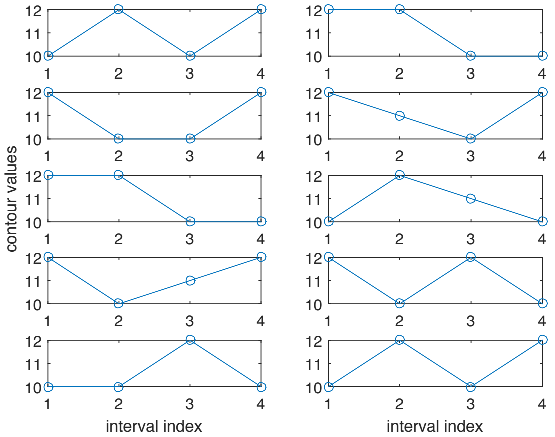


Fig. 9 The contour sequences for these ten videos.

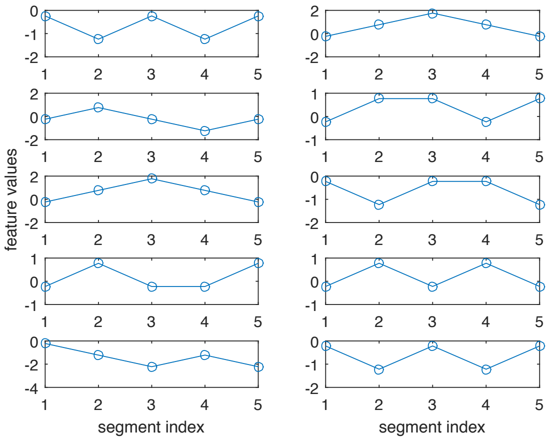


Fig. 10 Synthesized timelines from contour sequences. This synthesized timelines emphasize the variation patterns.

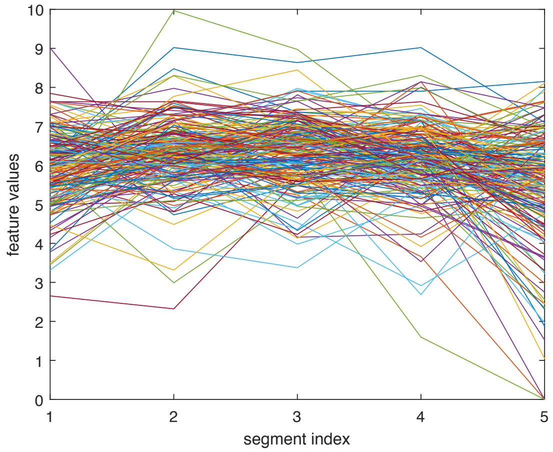


Fig. 11 Overlapped timelines of audio onset density from 200 videos. The density of timelines roughly shows the global distribution of the dominant shapes. An arc form from segment 1 to 5 is clearly visible, which corresponds to the conventional expectation-relaxation pattern.

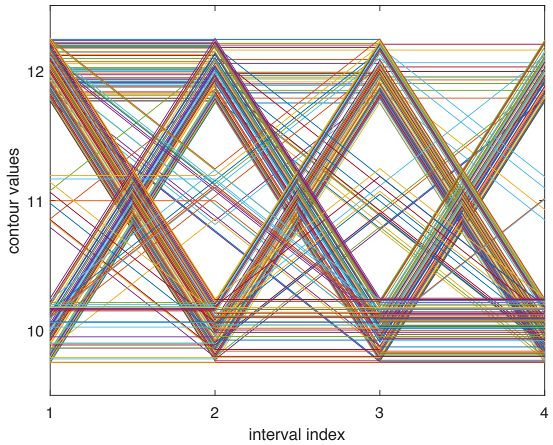


Fig. 12 Overlapped contour sequences extracted from 200 video timelines. A small vertical offset is appended to each contour shape to demonstrate their distributions. The dominant arc shap is evident, for example, interval "1" and "2" shows frequent ascending shapes.

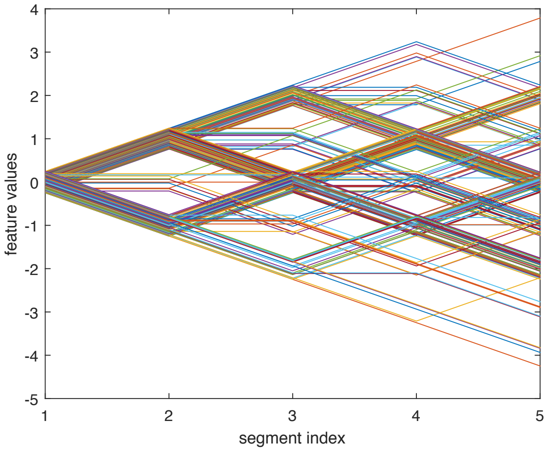
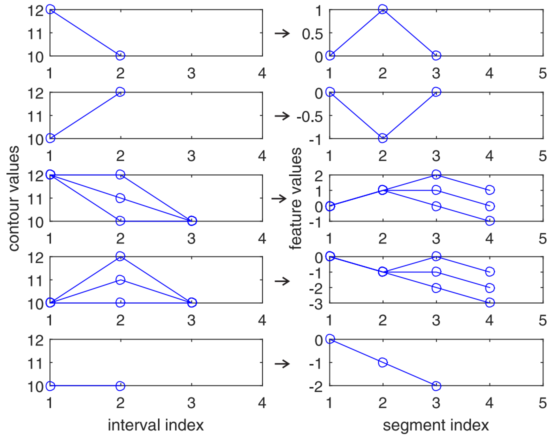


Fig. 13 Overlapped synthesized sequences.



(a)

```
<0> 12 -1 <1> 10 -1  #SUP: 156
<0> 10 -1 <1> 12 -1  #SUP: 126
<0> 12 -1 <2> 10 -1  #SUP: 103
 <0> 10 -1 <2> 10 -1  #SUP: 86
 <0> 10 -1 <1> 10 -1  #SUP: 86
```

(b)

Fig. 14 Contour pattern/timeline visualizations of the five most supported sequential patterns (a) and their text strings (b).

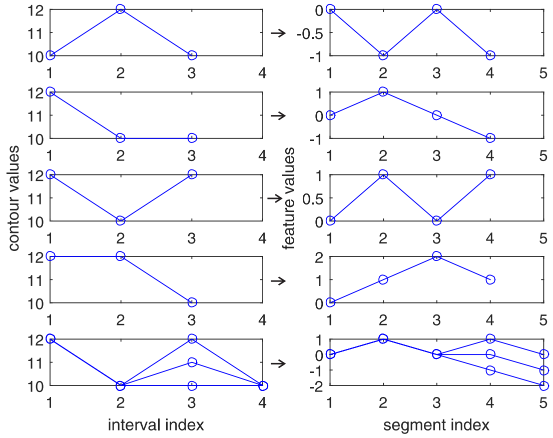


Fig. 15 Contour pattern/timeline visualizations of the five most supported length-three sequential patterns.

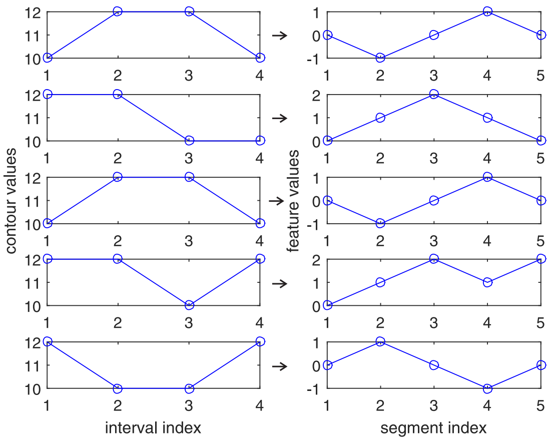


Fig. 16 Contour pattern/timeline visualizations of the five most supported length-four sequential patterns.

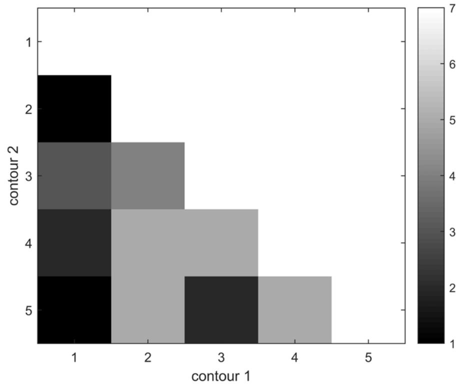


Fig.17 Similarity scores of contour patterns in Fig. 14, obtained from manual rating experiments. Ratings are in "1-7" Likert scale.

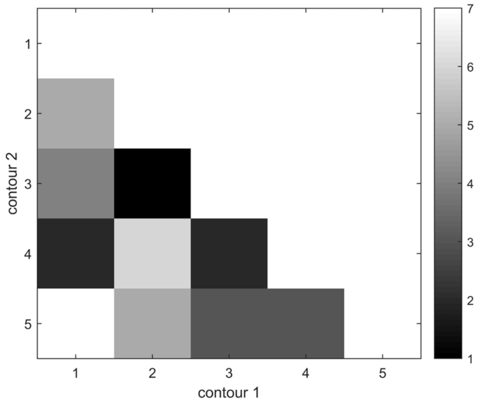


Fig. 18 Similarity scores of contour patterns in Fig. 15. These more complex patterns show less difference in the manual evaluation analysis.

V. Conclusions

In this paper, we implemented a computational framework for content-based multimedia timeline analysis using temporal contours represented as sequential patterns. Video and audio features are extracted from short segments of multimedia files and form the source timeline sequences for modeling the temporal dynamics of multimedia content. A temporal contour representation is implemented to model the different shapes inside each timeline dimension. Then the sequential motif discovery algorithm is employed to locate the statistically significant temporal contour elements. These significant contour elements form important devices for multimedia content analysis. Their relevance is testified from visual analysis and from subjective rating experiments.

The future research directions of our proposed processing framework are mainly on more comprehensive application scenarios and more rigorous subjective validation. Some immediate steps are listed below:

- Cover additional multimedia features, especially features that imitate human perception and cognition.
- A more comprehensive vocabulary of contour representations
- Modeling long time span timelines in high temporal resolutions. This is essential for connecting contour shapes at different resolution levels.
- More detailed subjective validation experiments, for example, increasing the number of repetitions in the rating experiments and comparing ratings from multiple multimedia segments with common contour shapes.

References

[1] L. Elin, A. Lapides, Designing and Producing the Television Commercial, Pearson, New York, 2003.

[2] I. Cury, Directing and Producing for Television: A Format Approach, 4th ed., Taylor & Francis, Boca Raton, FL, 2012.

[3] M. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 2(1), pp. 1-19, 2006.

[4] Y. Hirate and H. Yamana, "Generalized sequential pattern mining with item intervals," Journal of Computers, vol. 1(3), pp. 51–60, 2006.

[5] P. Fournier-Viger, SPMF: A Java Open-Source Data Mining Library, http://www.philippe-fournier-viger.com/spmf/

[6] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, M. Westover, Q. Zhu, J. Zakaria, E. Keogh: "Searching and mining trillions of time series subsequences under dynamic time warping," KDD 2012, pp. 262-270.

[7] N. Graakjaer, Analyzing Music in Advertising: Television Commercials and Consumer Choice (Routledge Interpretive Marketing Research), Routledge, New York, 2014.

[8] J. Bignell, An Introduction to Television Studies, 3rd ed., Routledge, New York, 2012.

[9] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, "Learning time-series shapelets," in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '14), pp. 392-401, 2014.

[10] T. Rakthanmanon and E. Keogh, "Fast shapelets: a scalable algorithm for discovering time series shapelets," proceedings of the 2013 SIAM International Conference on Data Mining, pp. 668-676, 2013.

[11] L. Ye and E. Keogh, "Time series shapelets: a novel technique that allows accurate, interpretable and fast classification," Data Mining and Knowledge Discovery, vol. 22(1), pp.149–182, 2011.

[12] Illustration of HSV color space: http://www.mathworks.com/help/images/convert-from-hsv-to-rgb-color-space.html

[13] T. Quatieri, Discrete-Time Speech Signal Processing: Principles and Practice, Prentice Hall, Hoboken, NJ, 2001.

[14] B. Moore, An Introduction to the Psychology of Hearing, 6th ed., Emerald Group Publishing, Bingley, United Kingdom, 2012.

[15] M. Müller, Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications, Springer, New York, 2015.