MUSIC DATABASE QUERY WITH VIDEO BY SYNESTHESIA OBSERVATION

Ruiduo Yang and M. S. Brown

Department of Computer Science
Hong Kong University of Science and Technology
Hong Kong

ABSTRACT

In this paper, we propose a novel framework to query a music database by supplying an MPEG video sequence. The music and video are treated as time series data and the retrieval query is based on the similarity between corresponding features in the music and video sequence (e.g. the tempo in the music and the motion in the video). A similarity measure for matching these features is proposed to capture the synesthesia effect in music-video.

1. MOTIVATION AND RELATED WORK

Multimedia information retrieval systems can generally be divided into two types: audio retrieval and video/image retrieval. In these systems, the user gives an example of one of the medias (i.e. music, image, or video) to make a query. Media items in the database that are similar in terms of either feature similarity or semantic similarity, or both, will be returned as the query results.

Various techniques have been proposed to do both visual and aural information retrieval, however these techniques have been designed to work either on visual and audio information separately. The matching between visual and aural data has received little attention. Allowing users to query music by visual examples would have great application for professional/amateur multimedia editing such as animation design, advertisement design, music video design, etc. For example, a user could author a short video and then try to find suitable music to accompany the video.

To this end, we consider the synesthesia effect between visual and aural data. Synesthesia is a condition in which one type of stimulation evokes the sensation of another. For example, when the hearing of a particular sound produces the effect of visualizing a color. Forerunners in art and computer animation have been aware of synesthesia for a long time. In [3], Whitney exploited the relationship between music and simple geometry animation. Whitney also proposed the technique called digital harmony that has been widely used in today’s digital based music players, for example the MP3 player Winamp [4] and Microsoft Media player [5]. Digital harmony tries to first detect the underline features inside the music such as transcription, genre, tempo, etc, after which the simple animation will be made in terms of dot, line, shape and their motion so that the synesthesia can be made automatically.

Similar efforts for making synesthesia between audio and visual can be found in recent work. Hiraga et al. [6] proposed a technique to do music performance visualization, by which the personal performance of an instrument can be visualized as 3D animation. Foote et al. [7] resize video data so that it can be perfectly matched to a given music. Their work can be considered as extension of digital harmony.

Previous work has also been revealing in help to summarize the relationship between aural and visual information [3,6,7]. Table 1 shows correspondences between aural and visual that compiled from several sources. These show proposed synesthesia between music and color (in HVS space) and shape.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Correspondence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pitch</td>
<td>Color scales</td>
</tr>
<tr>
<td>Amplitude</td>
<td>Loud or Muted</td>
</tr>
<tr>
<td>Overtones</td>
<td>Color tone &amp; overtones</td>
</tr>
<tr>
<td>Tempo</td>
<td>Modulation to nuance</td>
</tr>
<tr>
<td>Interval</td>
<td>Contrast intervals</td>
</tr>
<tr>
<td>Mode</td>
<td>Mode to color shade</td>
</tr>
</tbody>
</table>

Table 1: Correspondence between music features and visual features in HSV model.

Besides the synesthesia, great efforts have been made in recent years to do music information retrieval. This is due
to that the development in internet technology has made a large volume of music audio data available to the general public. In [8], Yang et al. has given an excellent review for the state-of-art’s music retrieval techniques. In particular, Yang et al. have divided the music data into three categories: symbolic music (music with transcriptions, for example MIDI), monophonic music and polyphonic music. We extend Yang’s classification as in Table 2. In the cell marked S2S both the query and the underlying database are in symbolic formats, in the cell marked by QBH, the monophonic acoustic query with a symbolic database represents a problem which is known as Query by Humming, or QBH [9]. Finally the cell marked with A2A represents the problem that Yang et al. want to solve is acoustic to acoustic matching. In particular they use the spectrograms of the music to process and match music.

<table>
<thead>
<tr>
<th>Symbolic</th>
<th>Acoustic</th>
<th>Visual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbolic</td>
<td>S2S</td>
<td>QBH</td>
</tr>
<tr>
<td>Acoustic</td>
<td>A2A</td>
<td>(our approach)</td>
</tr>
<tr>
<td>Visual</td>
<td>Video/image retrieval</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Classification of multimedia retrieval, the different columns indicates different query data, while different row indicates different database media.

We extend the last column and row on the base to add visual data. For our approach, the incoming query will be visual such as a video sequence, with the corresponding music returned to be music data hopefully appropriate to accompany the video used in the query.

The remainder of this paper is organized as follows: in section 2 a framework for our system is overviewed, section 3 discusses the proposed similarity measure for matching music and video, followed by the experiment results in section 4. Finally section 5 concludes this paper with a discussion of future work.

2. MUSIC RETRIEVAL SYSTEM FRAMEWORK

In our proposed framework, an MPEG video sequence as input query. The database consists of large volumes of music data. Following [8], we considered this music data as polyphonic because as is the usual case for computer music video/animation design. The system is outlined at Figure 1. Our system tracks features in the music sequences and store them as time series data. The input video sequence also needs to be processed to extract the corresponding features which can be referenced as in table 1. A correlation measure will be set up (see section 3) specifically for music-video data matching. The corresponding music sequences with higher correlation score will be returned as results.

![Figure 1: Framework for our multimedia retrieval system.](image)

3. CORRELATION MEASURE BY SYNESTHESIA OBSERVATION

In our retrieval system, both the video sequence and the music sequence are considered as a time series. A time series is a sequence of real numbers, representing the measurements of a real variable at equal time intervals [10]. There are several similarity measures available for matching two time series. Goldin and Kanellakis et. al. suggests normalization to capture the variance and mean [11]. Jagadish et. al. use transformation rules [12]. Rafiei et al. use moving averages of smoothed data [13], and Berndt proposed the dynamic time warping approach [14]. None of these methods, however, considers the synesthesia effect between a music sequence and video sequence. We use a metric where the synesthesia observation is encoded between the two time series. Our approach is as follows.

First, we denote the Music and video as:

Music : \( M_i \), \( i = 1, \ldots, n \) is the time index
Video : \( V_i \), \( i = 1, \ldots, n \) is the corresponding time index

In which \( M_i = \langle x_1, \ldots, x_k \rangle \), \( V_i = \langle y_1, \ldots, y_k \rangle \), with the \( x_1 \) to \( x_k \) to be the feature of music, which can be tempo, pitch, amplitude etc., while \( y_1 \) to \( y_k \) is the corresponding visual features of the video, which could be motion, hue, shape etc.

For any observation \( < M_i, V_i > \) we generate the corresponding synesthesia observation which is:

\[ < SM_{r1,2}, SV_{r1,2} > \quad \text{and} \quad < SM_{r2,2}, SV_{r2,2} > \]
In which:

\[ SM_{i;2-1} = M_i, \quad SV_{i;2-1} = \arg\min_{V_j} \text{dist}(M_i, V_j) \]

\[ SV_{i,2} = V_i, \quad SM_{i,2} = \arg\min_{M_j} \text{dist}(M_j, V_i) \]

Here, \( \delta t \) is a reasonable window size in which the synesthesia can happen. \( \text{dist} \) is the normalized distance function between any pair of music-video data. Hence for the time series \( M \) and \( V \), we generate their corresponding synesthesia time series as:

Music : \( SM_i \quad i = 1, \ldots, n \) is the time index

Video : \( SV_i \quad i = 1, \ldots, n \) is the corresponding time index

At the last step, the similarity measure score is computed by the correlation of the transformed synesthesia time series:

\[
\text{Score} = \frac{\sum_{i=1}^{n} (SM_i \cdot \bar{SM}) \cdot (SV_i \cdot \bar{SV})}{\sqrt{\sum_{i=1}^{n} (SM_i \cdot \bar{SM})^2 \cdot \sum_{i=1}^{n} (SV_i \cdot \bar{SV})^2}}
\]

4. TEMPO TRACKING FOR MUSIC AND MOTION VECTOR SYNTHESIS FOR VIDEO

In our system, the assumption is made that the tempo of music will be related to the motion of the video frame by frame. Tempo is defined for the speed of the music. Scheirer et al [15] has proposed the excellent tempo tracking method and the test of large amount of music has proved its reliability.

We followed this tempo tracking method. Suppose the original music signal is \( m_i \). After tempo tracking, we have \( T(m_i) \). In order to match this data with the video, \( T(m_i) \) will be smoothed and windowed at the input video’s frame rate. Let’s suppose the input video has a frame rate of \( f \) frames per second. The smoothing function is \( S(x) \) and the windowing function is \( W(x, f) \). Thus the derived music data to be:

\[ M_i = W(S(T(m_i))), f; \]

For video data, since we assume it is MPEG compressed the motion vector inside each frame can be directly extracted from the compressed stream. For each frame \( I \), we define the motion energy as:

\[
ME_i = \frac{\sum_{j \text{ for all block}} |mv_{j}| + \sum_{j \text{ for all block}} |\text{residual}_{j}|}{N_{1i} + N_{2i}}
\]

Here, \( mv \) and \( \text{residual} \) can be directly extracted from the MPEG video. \( N_{1i} \) is defined as the number of non-stational (non-zero motion vector) block, while \( N_{2i} \) is defined as the number of non-stational (non-zero residual) block. Both \( N_{1i} \) and \( N_{2i} \) are used to capture the local motion energy.

In this representation, each pair of observation at a certain time is denoted as:

\[ < M_i, ME_i > \]

After this, we will generate the synesthesia time series and compute the correlation score. The algorithm is showed below:

1. Track the tempo of each music data in the database to get \( M_i \).
2. For the input MPEG video. Compute the \( ME_i \)
3. For each music sequence in the database, generate the synesthesia time series \( < SM_i, SV > \).
4. Return a sorted list of music with the highest correlation score of \( SM \) and \( SV \)

In the experiment, we use several professional made music videos as the test example. Our music data base has 100 polyphonic music sequences, which includes the extracted music sequences for those professional made music-video. Table 3 shows us the matching result. In Table 3 column 1 is the name of the selected music videos. Column 2 returns the correlation score between the original video and its accompanying music. Column 3 returns the highest correlation score between the original video and the music in the database while column 4 is the average correlation score. The 5th column returns the rank of the original music that accompanied the video.

From the results we can see that the original music always has the best correlation with the input video. This is because the author that created the music-video has selected music that is intended to maximize the synesthesia effect. The other music, however, may have the boundary mismatching or tempo mismatching with the original video, and hence induce penalties the correlation scores.
<table>
<thead>
<tr>
<th>Sequence</th>
<th>Original Music Score</th>
<th>Highest Score</th>
<th>Average Score</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basket Ball</td>
<td>0.23</td>
<td>0.23</td>
<td>0.08</td>
<td>1</td>
</tr>
<tr>
<td>Cell Phone Ad.</td>
<td>0.36</td>
<td>0.36</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>Computer Ad.</td>
<td>0.30</td>
<td>0.30</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>Company Ad.</td>
<td>0.38</td>
<td>0.38</td>
<td>0.12</td>
<td>1</td>
</tr>
<tr>
<td>Flash</td>
<td>0.33</td>
<td>0.33</td>
<td>0.07</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3. Experiment Result for 5 professional made music videos. In the experiments, all the video has a frame rate of 30fps. Sequence “basketball” has 2000 frames while the other sequences have 900 frames. We set $\delta t$ to be 0.05 seconds.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a framework to perform music retrieval using a video query. We also provided a similarity measure to effectively match the video and music data of corresponding synesthesia features. We demonstrate this idea based on the assumption that the motion in video can be related to the tempo in music. Future work can include more corresponding features between the video and music. A learning based approach can also be taken with a large database of professionally made music video to try to automatically derive synesthesia features. Efficient means of indexing and retrieving queries based on synesthesia observation should also be considered.

6. REFERENCES


