

Structure-Aware Music Resizing Using Lyrics

Zhang Liu
Department of Computer
Science and Technology
Tsinghua University
Beijing 100084, China
liuzhang08@mails.thu.edu.cn

Chaokun Wang
School of Software
Tsinghua University
Beijing 100084, China
chaokun@tsinghua.edu.cn

Jianmin Wang
School of Software
Tsinghua University
Beijing 100084, China
jimwang@tsinghua.edu.cn

Wei Zheng
School of Software
Tsinghua University
Beijing 100084, China
zhengw04@mails.thu.edu.cn

Shengfei Shi
School of Computer Science
and Technology
Harbin Institute of Technology
Harbin 150001, China
shengfei@hit.edu.cn

ABSTRACT

World wide web provides plenty of multimedia resources for creating rich media web applications. However, the collected music and other media resources always mismatch in the metric of time length. Existent music resizing approaches suffer from perceptual artifacts which degrade the performance of resized music. In this paper, a novel structure-aware music resizing approach is proposed. Through lyrics analysis, our approach can compress different parts of a music piece in variant compression rates. Experimental results show that the proposed method can effectively generate resized songs with good quality.

Categories and Subject Descriptors

H.5.5 [Information Interfaces and Presentation]: Sound and Music Computing

General Terms

Algorithms

Keywords

Time-scale compression, music resizing, rich media

1. MOTIVATION

Web applications are equipped with rich media, e.g., Flash, Sliverlight, which provide better user experiences. Meanwhile, creating a rich media web application becomes much easier due to the plenty of multimedia resources available on the Internet. However, it is common case that two resources which are collected from the Internet do not match in the metric of time length, i.e., the time scale mismatch. For example, a user wants to accompany a 200s animation or a 220s speech audio with a 240s song. Compared with other media, e.g., video, speech, the semantic of music is more abstract and obscure. Manipulation of music makes fewer effects than that of other media in term of semantic integrity

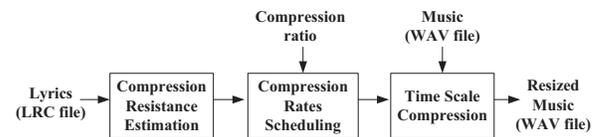


Figure 1: Proposed Music Resizing Framework

and continuity. Therefore, music is more appropriate to be resized when the time scale mismatch happens.

Existent music resizing approaches, i.e., time stretching [1], suffer from perceptual artifacts when the compression rate is still small, e.g., 15%. Through experiments, we discover that the perceptual artifacts always locate in the speedy parts with singing voice, but hardly happen in the pure instrumental parts or slow singing parts. In other words, the compression resistances of different segments in a music piece are different. This observation encourages our idea that compression rates on different music segments should be diverse according to the music structure and semantic content, which we name the structure-aware music resizing.

Contribution: In this paper, we propose a lyrics based structure-aware music resizing approach. Different from the existent time-scale compression technologies, our approach first obtains the semantic structure of a music piece through analyzing its lyrics. After that, a heuristic scheduling algorithm is adopted to arrange the compression rates of different music segments according to the music structure. Finally, the audio data of music is physically compressed via generic time scale compression algorithms, e.g., WSOLA [2].

2. MUSIC RESIZING WITH LYRICS

Our proposed structure-aware music resizing approach is composed of three components, i.e., the lyrics based compression resistance estimation, compression rates scheduling and time scale compression, as illustrated in Fig. 1.

2.1 Compression Resistance Estimation

Compression resistances of different music segments are estimated though analyzing the corresponding lyrics file (in Lrc format). Lrc files have been widely utilized by online music service providers and music players for synchronized

lyrics display. In a lrc file, lyrics words are organized as lyrics sentences and there are time tags indicating when each lyrics sentence begins playing. We can utilize the words and time tags to determine whether a segment has singing voice and estimate the singing speed of a segment. In details, we introduce the concept of *lyrics density*, which is defined as $d_i = \frac{n_i}{t_i}$, where n_i is the number of words in the i th segment and t_i is the play duration corresponding to the segment. The lyrics density can reveal the compression resistance of a segment. For example, high lyrics density indicates the segment is a speedy segment with singing voice. In contrary, a segment is considered to be pure instrumental music when the lyrics density is zero.

2.2 Compression Rates Scheduling

Given a music piece and a user demanding compression rate α , a scheduling algorithm calculates the compression rates of all segments in a music piece, aiming at reducing the effects of compression.

First, the total time reduction is calculated by $\Gamma_{total} = L \cdot \alpha$, where L is the music length. In the following steps, Γ_{total} will be distributed to segments of the music piece. Zero-density segments are processed before dealing with positive-density segments, because (1) the zero-density segments do not contain singing voice, and (2) the non-vocal parts of a music piece are less meaningful than their vocal counterparts in term of semantics. To make full use of non-vocal segments, the scheduling algorithm distributes Γ_{total} to zero-density segments as much as possible, say 80%.

In the second step, we start scheduling the remaining time reduction to positive-density segments if zero-density segments do not absorb all the time reductions. The scheduling process goes in an iterative way. More specifically, the remaining time reduction is split into m time slices with fixed length δT . During scheduling, one piece of time slice is assigned to a segment at each iteration. The selected segment at each iteration is called an active segment which should be selected strategically. For example, we can assign one time slice to the segment with the lowest lyrics density at each iteration. In this paper, we select the segment which has the *minimal lyrics density growth rate*. The lyrics density growth rate, κ , is defined as

$$\kappa_i = |d_i'| = \frac{\ell_i}{t_i^2}, (0 \leq i < n) \quad (1)$$

where ℓ_i and t_i represent the number of words and the length of the i th segment, respectively; n is the number of segment in a music piece. Once a time slice is assigned to the active segment, the length of active segment will decrease by δT and lyrics density of the active segment will increase correspondingly. The iteration goes on until there is no time slice to schedule. After all time slices are distributed, the compression rate of each segment can be calculated. And the audio data of the music piece can be physically compressed using a generic time scale compression algorithm according to the scheduled compression rates.

3. EVALUATION

Waveform graphs of the original song, resized songs are illustrated in Fig. 2. The sample song is *Don't cry for me, Argentina* and the compression rate is set to 30%. Figure 2(b) shows the waveform of song resized by homogeneous approach, which is exactly the same as the original one in

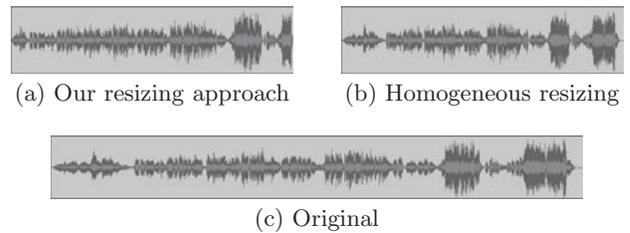


Figure 2: Waveform graphs of resized song using our resizing approach (a), song using homogeneous resizing (b) and the original song(c)

Fig. 2(c). Figure 2(a) displays that our approach compress sparse region more intensively so the waveform of the song looks more uniform than the waveform in Fig. 2(b).

To validate whether the resizing approach is acceptable to audiences, a user study is conducted. 10 skilled audiences participate the audition experiments. Every audience selects 8 different songs and each song is resized by our approach with compression rates of 10%–50%. They are asked to listen to all the compressed edition of a song and select the acceptable resized edition with the maximal compression rate. Experimental results are illustrated in Table 1. The global average value of maximal acceptable compression rates is 30.13%. Therefore, we can declare that most of audiences can accept the music pieces resized by our approach when the compression rate is smaller than 30%.

Table 1: Average acceptable compression rates

Tester ID	1	2	3	4	5
α (%)	20.00	38.75	38.00	23.75	36.00
Tester ID	6	7	8	9	10
α (%)	33.75	37.14	31.25	21.25	21.43

4. CONCLUSION

In this paper, we propose a structure-aware music resizing approach to address the time scale mismatch problem. In order to measure the compression resistances of different segments, we employ the concept of lyrics density. And we develop a heuristic scheduling algorithm to schedule compression rates on different segments in a music piece. Extensive experiments are conducted proving that our approach has better performance than the existent technologies.

5. ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China (No. 60803016), the National Basic Research Program of China (No. 2007CB310802) and the National High Technology Research and Development Program of China (No. 2008AA042301).

6. REFERENCES

- [1] J. Laroche and M. Dolson. Improved Phase Vocoder Time-scale Modification of Audio. *IEEE Transactions on Speech and Audio*, 7(3):323–332, May 1999.
- [2] W. Verhelst and M. Roelands. An Overlap-add Technique based on Waveform Similarity (WSOLA) for High Quality Time-scale Modification of Speech. In *IEEE ICASSP*, volume 2, pages 554–557, 1993.