AUDIO RESTORATION BY CONSTRAINED AUDIO TEXTURE SYNTHESIS

Lie Lu
Microsoft Research Asia
Beijing 100080, China
llu@microsoft.com

Yi Mao
Zhejiang University
Hangzhou 310027, China
myb_79@sina.com

Liu Wenyin
Dept. of Computer Sci.
City University of HK
csliuwy@cityu.edu.hk

Hong-Jiang Zhang
Microsoft Research Asia
Beijing 100080, China
hjzhang@microsoft.com

ABSTRACT

Audio texture, a new audio medium, is used to synthesize long audio stream according to a given short example audio clip. In this paper, we extend this idea to audio texture restoration, or constrained audio texture synthesis for restoring those missing parts in an audio clip. It is useful in many applications such as audio restoration and audio reconstruction. It can also be used in error concealment for audio/music delivery with packet loss on the internet. A novel method is proposed for constrained audio texture synthesis. Preliminary results are provided for evaluation.

1. INTRODUCTION

In our previous research [1], audio texture, which is inspired from video textures [4], was proposed as a new audio media. Audio data, as a signal sequence, presents self-similarity as a video sequence does. The self-similarity of music or audio [2] is used for audio texture analysis and construction.

Audio texture provides an efficient means of synthesizing a continuous, perceptually meaningful, yet non-repetitive audio stream from an example audio clip. It is “perceptually meaningful” in the sense that the synthesized audio stream is perceptually similar to the given example clip. However, an audio texture is not just a simple repetition of the audio patterns contained in the input; variations of the original patterns are fused into it to give a more vivid stream. The audio stream can be of arbitrary length according to the need. It is very useful in many applications, such as lullabies, game music and background music in screen saver. These sounds are relatively monotonic, simple in structure, and have repeated yet possibly variable sound patterns. A very long simple but not exactly repeating sound would require huge storage. Audio texture is proposed to solve this problem.

In some applications, a part of audio signal may be lost because of some reasons. For example, in the application of audio/music delivery on the internet, the above result will be caused by lost packets. In our paper, we assume the missing part is filled in zero value, such as Fig. 1 shows. How to restore the original audio is a challenging problem. In this paper, audio texture, which utilizes the audio self-similarity, is applied to restore the missing part.

There are many works on digital audio restoration. However, in these works, the objective is not to restore the missing part, but to restore the degraded audio signals, such as click removal, noise removal for gramophone recordings, film sound tracks, and tape recordings [6].

In some applications of error concealment for audio/music delivery with packet loss on the Internet, they should also restore the lost packets, e.g., as done in [5]. However, most of traditional error concealment methods only dealt with errors with a short length (typically around 20ms). Our method can reconstruct the audio with loss of a much longer length, such as 1 second.

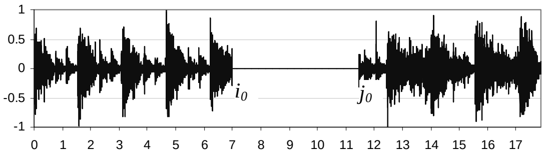

Fig. 1 An audio example which lost many frames

The main idea of our method is to restore the missing part by audio texture synthesis based on self-similarity of audio. In this case, the generated audio texture is constrained by the beginning and ending points of the missing part. That is, the synthesized part should be perceptually smooth at the joint points with the remaining audio clip. No uncomfortable break or click is expected at those points. Hence, we refer to this kind of audio restoration as audio texture restoration, or constrained audio texture synthesis. Unconstrained audio texture synthesis is discussed in our previous work [1], where an audio stream with arbitrary length is created from an original short audio clip. The key issue of constrained audio texture synthesis is how to generate a frame sequence which can be perceptually smoothly inserted to replace the missing part.

The rest of the paper is organized as follows. Section 2 presents an overview of the proposed method for constrained audio texture synthesis. Section 3 describes algorithms for analyzing audio structure. Section 4 describes the algorithms for constrained synthesis process. Section 5 presents settings for the experiments and provides preliminary results.

2. SYSTEM OVERVIEW

The proposed method for constrained audio texture synthesis can be divided into two stages: analysis and synthesis, as shown in Fig. 2.

In the analysis stage, certain features are extracted to represent the original audio data. The most important feature in our approach is Mel-Frequency Cepstral Coefficients (MFCCs). Then, the structure of the audio clip is analyzed, and the similarity and transition probability between each pair of two neighboring frames are calculated for further synthesis. In the synthesis stage, we use frame instead of sub-clip [1] as the synthesis unit. The frame sequence that will be used to replace the missing part is determined based on the transition probabilities and constrained conditions.

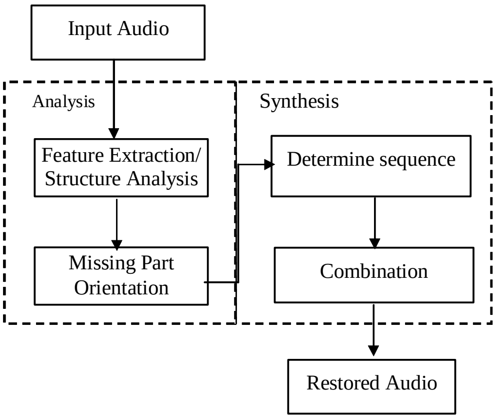

Fig. 2. System overview diagram

3. AUDIO TEXTURE ANALYSIS

This step is used to analyze the structure of the input audio clip and locate the position of the missing part.

In order to restore an audio texture, it is necessary to consider the similarity between any two frames and the transition probability from one to another. It is the basis for synthesis.

Let $\boldsymbol{V}_i$ and $\boldsymbol{V}_j$ be the feature vectors of frames i and j in the MFCC feature space. The similarity measurement is simply based on vector autocorrelation and defined as follows.

$$S_{ij} = \frac{\boldsymbol{V}_i \bullet \boldsymbol{V}_j}{\|\boldsymbol{V}_i\| \cdot \|\boldsymbol{V}_j\|} \tag{1}$$

The above measure considers the isolated two frames only. In order to give a more comprehensive representation of the similarity, it will be better if their neighboring temporal frames are taken into considerations. Suppose the previous m and next m frames are considered with weights $[w_{-m},\ldots,w_m]$, a better similarity is developed as follows.

$$S'_{ij} = \sum_{k=-m}^{m} w_k S_{i+k,j+k} \tag{2}$$

This method captures the time dependence of the vectors. To yield a high similarity score, it requires that the two subsequences be highly similar.

The transition probability from frame i to frame j depends on the similarity between frames $i+1$ and j. The more similar these two frames are, the higher the transition probability should be. In this principle, the transition probability is related to the similarity, as in the following exponential function.

$$P_{ij} = A \exp\left(\frac{S'_{i+1,j} - 1}{\sigma}\right) \tag{3}$$

where A is the normalizing constant such that $\sum_j P_{ij} = 1$, and σ is the scaling parameter.

After the audio structure is analyzed, we should estimate which part of the audio clip is missed. In our algorithm, novelty scores are used to locate the missing part. The novelty score is defined based on the similar matrix, as shown in Eq. (4). It can be used to measure the possibility of emergence of a new building pattern in the audio sequence [1].

$$N(i) = \sum_{m=-w/2}^{w/2} \sum_{n=-w/2}^{w/2} K_{m,n} S_{i+m,i+n} \tag{4}$$

where K is a kernel matrix, which is described in detail in [1].

Fig. 3 (a) shows an original audio clip and Fig. 3 (b) shows of the same audio clip but with a part missed. The novelty score curve for Fig. 3 (b) is shown in Fig. 3 (c). Obviously, the missing part is the part that has nearly zero value between the two highest peaks in its novelty score curve.

After the missing part is located, the similarity measure and the transition probability should be modified correspondingly, since the data of the missing part should not be used in the measurement. The modification is quite easy. We just need to modify the upper or low bound of formula (2) to ensure no data is in the missing part.

4. CONSTRAINED SYNTHESIS

In this step, we will consider to restore the audio texture based on the analysis results. We use frame as the synthesis unit instead of sub-clip [1], because of the following reasons:

(1) The missing part usually does not begin at the beginning of one sub-clip and end at the ending of another sub-clip. It may begin or end at the inner of sub-clips, just as Fig. 1 shows.

(2) It is difficult to estimate how many sub-clips should be inserted into the missing part, since each sub-clip is not of equal length.

(3) The above two problems do not exist when using the frame as the synthesis unit.

In the constrained audio texture synthesis, it is impossible to restore it exactly the same as the original one, since we don't know what it originally was. Thus, our objective is to generate an audio clip, which can replace the missing part and can be smoothly integrated into the remaining audio without uncomfortable perception. It is feasible based on the characteristics of self-similarity of the remaining audio texture. Now, the key issue is how to determine the frame sequence which can fill in to replace the missing part in the audio clip.

In the following section, we will introduce our approaches on frame sequence determination for audio texture restoration. We suppose the missing region of an audio clip, or the region to be synthesized, is from frame i_0 to frame j_0, such as Fig. 1 shows.

4.1. Global Similarity Estimation

Since the missing part has the similar property with the remaining part, a simplest method that can be used for constrained audio texture synthesis is to replace the missing part by using the most similar piece of the same length in the remaining audio clip.

In this case, the neighbor frames around should be used to represent the feature of the [i_0, j_0] region, since the frames between [i_0, j_0] are unknown. Suppose the previous *m* frames before i_0 and the next *m* frames following j_0 are considered to represent the characteristics of the context of [i_0, j_0], the feature vector of [i_0, j_0] can be represented as

$$\overline{V}_{i_0,j_0} = (V_{i_0-m}, V_{i_0-m+1}, \ldots, V_{i_0-1}, V_{j_0+1}, V_{j_0+2}, \ldots, V_{j_0+m}) \quad (5)$$

Then, the similarity between the context around [i_0, j_0] and the context around [*i'*, *j'*] can be represented as

$$S_r(\overline{V}_{i_0,j_0}, \overline{V}_{i',j'}) = \sum_{k=1}^{m} w_k (S_{i_0-k,i'-k} + S_{j_0+k,j'+k}) \quad (6)$$

where $j' - i' = j_0 - i_0$, and [w_{-m},...,w_m] are weights.

After computing the similarity between the context of [i_0, j_0] and the context of each [*i'*, *j'*] with the same length, the most similar part [*i**, *j**] can be used to replace the missing part:

$$[i^*, j^*] = \arg\max\{S_r(\overline{V}_{i_0,j_0}, \overline{V}_{i',j'})\} \quad (7)$$

This simple method is feasible, but there still exist some problems:

(1) This method considers global similarity of the context of the missing part, but it did not consider the perception continuity at the joint point i_0 and j_0, where possible abrupt change or click may exist.

(2) This method copies a segment of remaining audio to replace the missing part. Such exact repeat of another part in a very close time may cause discomfort to some listeners.

In order to solve these problems, another optimization method is used in our real implementation.

4.2. Frame Sequence Determination

In this approach, we will restore the missing part frame by frame. The main problem is to determine the frame sequence which can be smoothly inserted and replace the missing part without uncomfortable perception. The optimal frame sequence should satisfy:

1) Maximize the transition probability from frame i_0 to frame j_0.

2) Keep perceptual smoothness.

The determination of frame sequence from i_0+1 to j_0-1 can be described in the following mathematic model,

$$\max P(i_0 \to j_0) = P_{i_0,i_0+1} \cdot P_{i_0+1,i_0+2} \cdots\cdots P_{j_0-2,j_0-1} \cdot P_{j_0-1,j_0} \quad (8)$$

with the constraints:

$$P_{i,i+1} > p_0, \quad i_0 \le i \le j_0 - 1 \quad (9)$$

where p_0 is a threshold to select frame with enough large transition probability, which can be used to control the perceptual smoothness.

However, the feasible solution space of this problem is very large. Suppose under the constrained condition (9), the number of potential candidate frames after a given frame is N; and the number of missing frames is $M = j_0 - i_0 - 1$. Thus the size of feasible solution space is about N^M. This is too large for exhaustive search to find the optimal solution when *N* and *M* is large. Fortunately, dynamic programming can be used to find a sub-optimal solution, with an *O(NM)* complexity.

In general, dynamic programming is used to find a path from i_0 to j_0 which has the minimum cost. Hence, some modifications are needed to make the problem suitable for dynamic programming. Therefore, the transition probability P_{ij} should be changed to the cost of moving from point *i* to point *j*, which can be defined as:

$$c_{i,j} = \begin{cases} -\ln P_{ij} & (P_{ij} > p_0) \\ \infty & (P_{ij} < p_0) \end{cases} \quad (10)$$

After determining the frame sequence, the TD-SOLA (Time Domain – Synchronous OverLap-Add) method [3] is used to combine and smooth each two concatenated frames.

4.3 Global Amplitude Trend Consideration

In the above algorithm, we only considered the characteristics of spectrum. However, the optimal frame sequence had better keep not only the spectral characteristics, but also energy characteristics. We estimate the global amplitude variation trend in the missing part based on the method described in 4.1. Thus, when to determine the optimal frame sequence, the similarity of each two frames should be composed of two parts: spectral similarity and global amplitude tendency similarity. That is,

$$S = \lambda S_1 + (1-\lambda)S_2 \quad (11)$$

where S_1 and S_2 is the spectral similarity between two frame and the amplitude tendency similarity respectively, λ is the corresponding weight. The transition probability between two frames can then be calculated correspondingly.

5. PRELIMINARY EVALUATIONS

The proposed method does not attempt to estimate the original missing data, but to generate a replacement signal which is similar in the structure and can be smoothly integrated into the remaining data. It is difficult to give an objective evaluation, since it can not evaluate performance simply by comparing the waveform between original audio and restored audio. By casual subjective evaluation by some researchers, the proposed method gives promising results

Some examples of audio texture restoration are implemented by using the constrained audio texture synthesis algorithm presented in 4.2 and 4.3. The original audio clips are all 10-20 seconds long, sampled at the rate of 32KHz, mono channel, and encoded by 16bit per sample. Fig. 3 illustrates an example of our audio texture restoration algorithm. Fig. 3 (a) shows the waveform of an original audio clip; (b) shows the audio clip which is derived from (a) but lost some data of a long duration (from 7.4s to 8.5s); (c) shows the corresponding novelty score of (b), which is used to detect which part is missed in the audio clip; (d) shows the restored audio clip by using the proposed frame sequence determination algorithm.

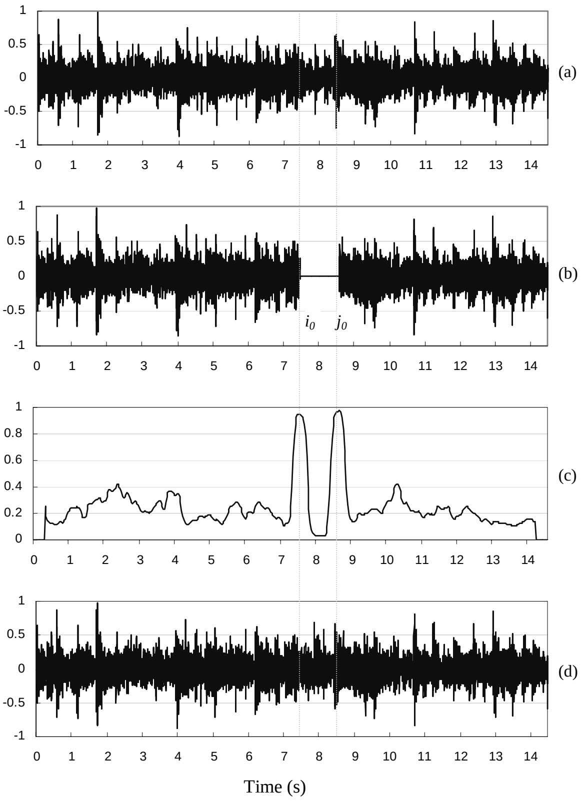

Fig. 3 An example of the audio texture restoration process. (a) waveform of the original audio clip; (b) a long duration (7.4s – 8.5s) of data is lost in the original audio clip; (c) novelty scores for detecting which part is missed in the audio clip; (d) the restored audio clip by using constrained audio texture synthesis.

Some preliminary experiment results are presented on the website: http://research.microsoft.com/~llu/AudioTextures/. The interested reader may want to compare the original sound and the restored audio texture.

6. LIMITATION AND FUTURE WORK

Although the proposed algorithm structure works promisingly in our testing data, it still has some limitations.

(1) The proposed algorithm is much more suitable for those audio textures with simple structures. For audio of simple structures, the missing signal is similar to the remaining part, so it can be restored well; but for those audio of complex structures, it is difficult to restore a perceptually smooth signal.

(2) There is no perception measure/criterion to control the texture synthesis procedure. Local perception and global perception are both extremely useful in the proposed approach, but they are difficult to obtain.

The constrained audio texture synthesis technique can be improved in several aspects in the future work. In the analysis step, we just used a correlation to measure the similarity between each two frames. It will be more useful if we could find a perceptual similarity measurement. In the synthesis step, we generated frame sequence based on local similarity. How to control the global perception of generated texture is still a difficult task. Other features, such as harmonics and pitch, will be helpful for audio texture restoration. In experiments, it would be better if more effective evaluation could be used on our algorithm. We would also extend our work to more traditional music. Thus, more powerful signal processing methods are needed.

7. CONCLUSION

In this paper, we have proposed our approach on audio texture restoration, or constrained audio texture synthesis. It is used to restore the missing part of an audio texture. The restored audio part can be smoothly integrated into the original audio clip without perceptually discomfort. Some casual subjective evaluation proved that the proposed method is promising.

There are many potential applications of this approach, such as audio restoration and audio reconstruction. It can also be used in error concealment for audio/music delivery with packet loss on the Internet. Audio texture synthesis is a new concept. We also hope the new concept could inspire more research work in the audio and related field.

8. REFERENCES

[1] L. Lu, S. Li, L. Wenyin, H. J. Zhang and Y. Mao. “Audio textures”. Proc. of ICASSP2002, Vol. II, pp. 1761 – 1764, 2002.

[2] J. Foote. “Visualizing Music and Audio using Self-Similarity”. In *Proc. ACM Multimedia ’99*, pp. 77-80, Orlando, Florida, November 1999.

[3] E. Moulines, F. Charpentier. “Pitch-Synchronous Waveform Processing Techniques for Text-To-Speech Synthesis using Diphones”, *Speech Comm.*, Vol.9, pp453-467, 1990.

[4] A. Schodl, R. Szeliski, D.H. Salesin, I. Essa. “Video Textures”. *Computer Graphics Proceedings, Annual Conference Series*, pages 33-42, Proc. SIGGRAPH 2000, July 2000. ACM SIGGRAPH

[5] Y. Wang. “A Beat-Pattern based Error Concealment Scheme for Music Delivery with Burst Packet Loss”. In *Proc. ICME’01*, pp.73-76, 2001

[6] S. J. Godsill, P. J. W. Rayner, and O. Capp'e. “Digital Audio Restoration”. In K. Brandenburg and M. Kahrs, editors, *Applications of Digital Signal Processing to Audio and Acoustics*. Kluwer Academic Publishers, 1996