MATConcat: An Application for Exploring Concatenative Sound Synthesis Using MATLAB

Bob L. Sturm

Abstract—The author has developed an application in MATLAB implementing concatenative sound synthesis (CSS) using feature matching. CSS is a process of combining short pieces of recorded sound to construct new sonic forms. Historically, CSS was developed for text-to-speech synthesis, but recently it has been explored as a musical sound synthesis method. The results have been called ‘musaics,’ the sonic analogue to mosaics made from small pieces of colored tile. Though this MATLAB application is less sophisticated than other CSS algorithms, it is meant to be a free and open application for demonstrating and experimenting with the process. The author has used this application to create many interesting and entertaining sound examples, as well as compositions. The application, and many sound examples, are available at http://www.mat.ucsb.edu/~b.sturm.

Index Terms—Concatenative sound synthesis, feature extraction, feature matching, composition

I. INTRODUCTION

A method exists in the synthesis of computer speech, called concatenative synthesis [1]. This technique, developed in the early sixties, is used mostly for text-to-speech synthesis. A computer segments written text into elementary spoken units that are synthesized using a large database of sampled speech sounds, like “ae”, “oo”, “sh”. These components are pieced together to obtain a synthesis of the text. These methods have recently been applied to creating “audio mosaics,” or “musaics” [2], [3], [4], but instead of using written text, they use recorded sound. As in mosaicing, a ‘target’ sound is approximated by small sound samples from a ‘corpus.’

Creative application of concatenative sound synthesis (CSS) has been minimal, and software for exploring it is not available. The author thus created an application to explore this technique for composing music. MATConcat is an implementation of CSS in MATLAB, a powerful but slow mathematics software language. With MATConcat a sound or composition can be concatenatively synthesized from audio segments in a database of any size. CSS provides many interesting and unique possibilities for sound design and electroacoustic composition. I have used it to create several intriguing sound examples, as well as electroacoustic compositions. These demonstrate the potential of this technique for sound synthesis.

II. ALGORITHM

The techniques used in MATConcat are very simple. Figure 1 displays the algorithm used. An analysis of the target, or the sound being approximated, produces feature vectors for ‘frames’ taken by sliding a user-specified window across the audio by a constant hop size. A six-element feature vector is created for each frame of the sound. Table I shows the current dimensions of the feature vector and interpretations of each component.

<table>
<thead>
<tr>
<th>Feature Measure</th>
<th>Meaning of Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Zero Crossings</td>
<td>General noisiness, existence of transients</td>
</tr>
<tr>
<td>Root Mean Square (RMS)</td>
<td>Mean acoustic energy (loudness)</td>
</tr>
<tr>
<td>Spectral Centroid</td>
<td>Mean frequency of total spectral energy distribution</td>
</tr>
<tr>
<td>Spectral Drop-off</td>
<td>Frequency below which 85% of energy exists</td>
</tr>
<tr>
<td>Harmonicity</td>
<td>Deviation from harmonic (integral) spectra</td>
</tr>
<tr>
<td>Pitch</td>
<td>Estimate of fundamental frequency</td>
</tr>
</tbody>
</table>

Table I: Current Feature Vector Elements

The analysis database of sound used for the synthesis is called the corpus, which can be several seconds to hours long. The analyzed sound being approximated is called the target. Iterating through the frames of the target analysis, optimal matches are found in the corpus database using specified matching parameters and thresholds. Other options can be specified, like forcing a match or selecting one at random, or extending the previous match if none is found. These numerous matching criteria and synthesis options creates many different possibilities.

III. EXAMPLES

Several intriguing sound examples have been created so far. The dramatic percussion crescendi from Gustav Mahler’s second symphony have been synthesized using corpora of monkey and animal sound effects, a Muslim Imam chanting the Koran, an hour of vocal music by John Cage, three hours of nostalgic Lawrence Welk, and all four string quartets of Arnold Schoenberg.

The example using the monkey vocalizations, shown in Figure 2, is particularly amazing. In this example the RMS and spectral roll-off feature elements are matched. The slowly building crescendo is ‘aped’ by the monkeys, creating a sense of increasing hysteria. At the climax the dominant gorillas grunt as lesser monkeys cower in submission. Synthesizing the same target using the same matching criteria, but from a corpus of John Cage’s vocal music, creates
Fig. 1. Algorithm of MATConcat. A target sound is analyzed by a constant hop-size window, creating for each frame a unique feature vector that identifies its characteristics (figure I). The best match for each frame in the target is found in the collection of sounds called the corpus—analyzed in the same manner—based on criteria specified by the user.

Fig. 2. Mahler’s crescendi performed by London Symphony Orchestra (Gilbert Kaplan, cond.) (top), and performed by ensemble of monkeys (bottom).

an entirely different sonic experience. The impressions of Mahler’s crescendi remain however.

IV. Conclusion

Through the sound examples and compositions created, MATConcat demonstrates that this relatively simple implementation of CSS creates effective and intriguing sound and music material. MATConcat serves well as a massive sample-mill, grinding sound into minuscule pieces for reconstitution into familiar forms. Surely with machine listening and score analysis, other interesting possibilities will emerge; but currently this implementation of CSS is far from being exhausted.

As it stands MAT Concat is prototype software. To take advantage of the numerous possibilities of CSS, this application will be ported to C++. Many improvements can be made including increased speed, increasing the dimensions of the feature vector, and expanding the list of synthesis options. Future work will implement the features of the MPEG-7 audio framework standard [5], and working toward a real-time implementation. These extensions will further open up the interesting avenues for creative CSS.

MATConcat, and many sound examples, are available for free at http://www.mat.ucsb.edu/~b.sturm.

REFERENCES