MATCONCAT: AN APPLICATION FOR EXPLORING CONCATENATIVE SOUND SYNTHESIS USING MATLAB

Bob L. Sturm

Electrical and Computer Engineering
Graduate Media Arts & Technology Program (MAT)
University of California, Santa Barbara, USA
b.sturm@mat.ucsb.edu

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 audio mosaic 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. It has also been used to create several electroacoustic compositions. The application, and all of the sound examples presented here, can be downloaded for free from `http://www.mat.ucsb.edu/~b.sturm`.

1. INTRODUCTION

A mosaic is a picture assembled by smaller pieces that contribute to the overall perception of an image. Close up the picture is not clear, but further away an image emerges. Figure 1(a) shows a mosaic assembled by hundreds of photographs, Figure 1(b), instead of colored tiles [1]. This process, called 'photo-mosaicing,' selects picture-tiles that are most similar to portions of the original image.

(a) **(b)**

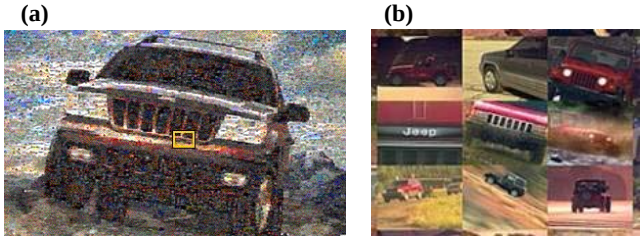

Figure 1: *A Photomosaic.*

A method similar to photo-mosaicing exists in the synthesis of speech, called 'concatenative speech synthesis' [2]. 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" ([3], [4], [5], [6], [7]). As in photo-mosaicing, a 'target' sound is approximated by samples from a 'corpus.' Schwarz [6] uses intelligent segmentation of the sounds by demarcating notes, or analyzing with a MIDI score. A deeper analysis is made by subdividing the segments into attack, sustain, and release portions. For each analyzed 'unit' Schwarz calculates a feature vector using several parameters, including mean values, normalized spectra, and unit duration. These units are then used to synthesize a target that is specified by either a symbolic score (MIDI) or audio score (sound-file). The units are selected based on their 'cost,' or perceptual similarity, to the original unit. Minimizing this cost results in the best synthesis possible using the database. Zils and Pachet [7] propose a similar method for creating "musaics," but include specific constraints, such as pitch and percussive tempo.

So far creative application of concatenative sound synthesis (CSS) is minimal, and software for exploring it is not available. The author thus decided to create an application to explore this technique. *MATConcat* is an implementation of CSS using feature matching in MATLAB. With this program 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. *MATConcat* has been used to create several intriguing sound examples, as well as some electroacoustic compositions. These demonstrate the potential of this technique for sound synthesis.

2. MATCONCAT

The algorithm used in *MATConcat*, Figure 2, is much more simple than in [4], [5] or [7]. Instead of segmenting the audio using an auxiliary score, or attempting to determine the content of a unit, the analysis 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 1 shows the current dimensions of the feature vector and interpretations of each component.

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. For instance, in the screenshot of *MATConcat*, Figure 3, the user has specified in the bottom-middle pane to first find all corpus frames that have a spectral centroid within $\pm 10\%$ of each target analysis frame; and from these matches pick the corpus frame that is within $\pm 5\%$ of the target analysis frame RMS.

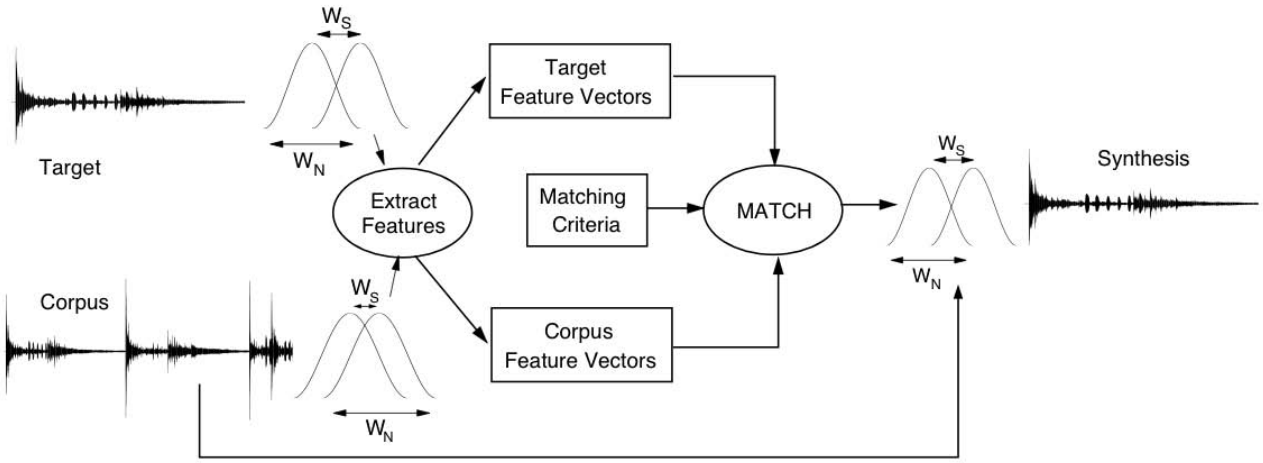

Figure 2: *Algorithm of MATConcat.*

Feature Measure	**Meaning of Feature**
Zero Crossings	General pitchiness
RMS	Mean acoustic energy
Spectral Centroid	Mean frequency of spectral energy distribution
Spectral Drop-off	Frequency below which 85% of energy exists
Harmonicity	Deviation from harmonic spectrum
Pitch	Estimate of fundamental frequency

Table 1: *Current Feature Vector Dimensions.*

The user can specify any number of features to match in any order; but as the number of features increases, the probability of finding matching frames becomes small unless the corpus grows in size. Once the best matches are found, a frame is either selected at random from these or the most optimal frame is chosen (an option specified by the user). The matching audio frame is then accessed from the corpus sound-file and written into the target synthesis according to the settings given in 'synthesis parameters,' e.g. window shape, size, and skip.

It is not necessary to keep the window or hop-sizes the same for the analysis and synthesis. One can specify a short hop-size for the target analysis and synthesize it with a larger hop-size. This will obviously make the synthesis longer than the original. For instance, in Figure 3, the panels at the top-left show information about the analysis databases. Note that the target was analyzed using a window size of 512 and window skip of 256 samples (512, 256). The corpus was analyzed with resolution (16384, 1024). If the synthesis uses a hop-size of 1024, its total duration will be four times that of the target.

Once the synthesis process has finished, *MATConcat* displays the synthesized sound in the upper-right corner and the matching process output in the lower-right corner. As can be seen, in frame 10 the number of corpus frames matching the spectral centroid criteria is 39; and from this the number of frames satisfying the RMS threshold is only 1. If no match is found then the frame is either left blank, a best match is forced, or the previous match is extended to fill the gap—depending on specified synthesis options.

There are currently six synthesis options. Specifying the 'Force Match' option finds the next best match if none is found within the given thresholds. If many matches are found, the default action is to select one closest to the original; this can be overridden by selecting 'Random Match.' 'Force RMS' will normalize the match to the RMS of the target frame. In this way one can preserve the amplitude envelope of the target while satisfying other matching criteria. If no match is found, this can either be left blank, or when the 'Extend Matches' is selected, the last successful match will be extended to fill the gap. This creates interesting moments when short frames are suddenly expanded to reveal longer phrases. One can also specify to reverse the corpus samples, or convolve the target and corpus frames.

3. EXAMPLES

Several intriguing sound examples have been created using *MATConcat*. 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 4, is particularly amazing. In this example the RMS and spectral roll-off components are matched to within $\pm 5\%$ and $\pm 10\%$, respectively. 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 an entirely different experience. The impressions of Mahler's crescendi remain however.

A recording of American President George W. Bush has been synthesized by corpora of monkeys, alto saxophone, and Lawrence Welk, and Bach's Partita for flute. By choosing the right window parameters the speech can still be understood—perhaps though only after hearing the original. When specifying a suitably small spectral centroid and roll-off, much of the sibilance and breathiness remains, especially when using the saxophone and flute corpora.

Specifying a target that is polyphonic understandably leads to trouble. The beginning of Schoenberg's fourth string quartet pro-

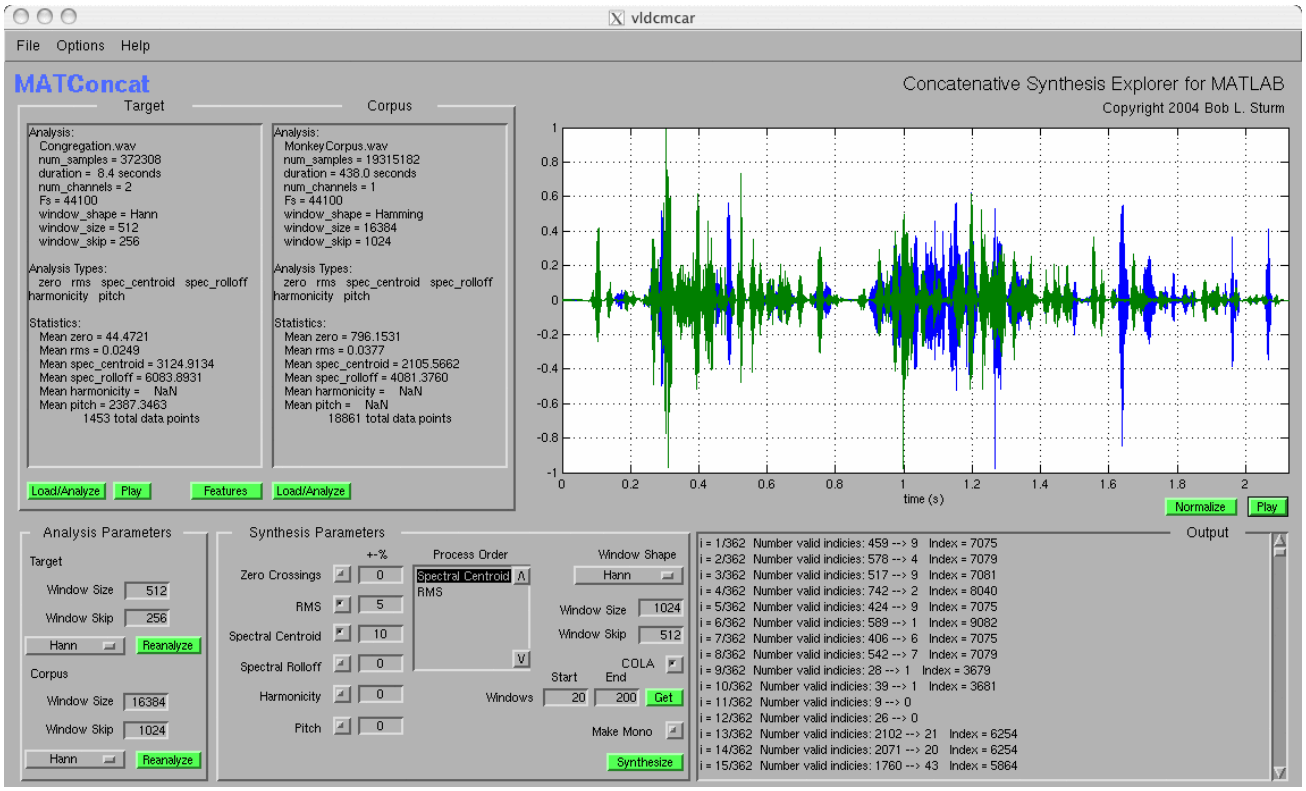

Figure 3: *Screenshot of MATConcat.*

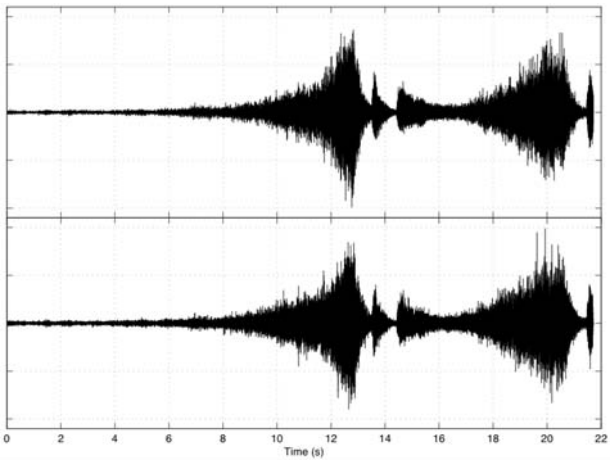

Figure 4: *Mahler's crescendi performed by London Symphony Orchestra (Gilbert Kaplan, cond.) (top), and performed by ensemble of Monkeys (bottom).*

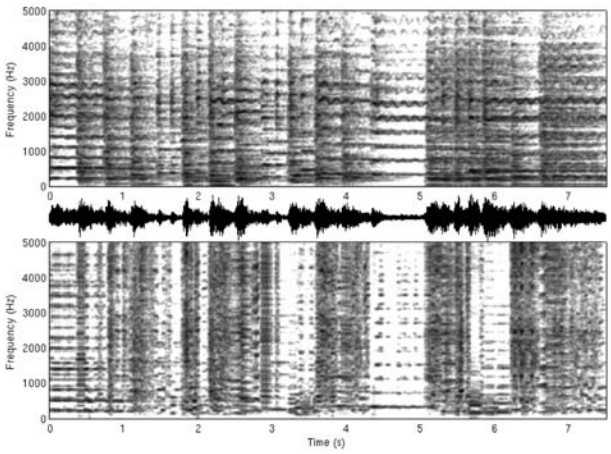

Figure 5: *Beginning of Schoenberg's fourth string quartet performed by the Arditti String Quartet (top and middle), approximated by Anthony Braxton on alto saxophone, matching pitch* $\pm 1\%$ *(bottom).*

vides an interesting example. A solo viola plays the main theme, punctuated by the other players. The first eight seconds of this piece were concatenatively synthesized using alto saxophone within a pitch threshold of $\pm 1\%$. The original time series and sonogram, along with the sonogram of the synthesis, are shown in Figure 5. Only at the times 0 – 1, 3, and 4.2 – 5 seconds does there appear to be any success. Auditioning the result confirms this observation; the melody is very discontinuous, but can be heard with effort.

It is quite easy for the human listener to hear the melody as continuous in the original passage; but for the machine this task becomes impossible without expert knowledge, i.e. gestalt principles [8], timbre recognition, score following, etc. In the synthesis, the moments at which the theme becomes clear are those in which the viola is the only instrument heard. All other mo-

ments are squeaks and squawks attempting to accommodate the transients created by the accompanying strings playing staccato. A score following technique, such as that implemented by Schwarz [5], would probably work best for polyphonic targets.

Using CSS and *MATConcat*, two multi-movement electroacoustic compositions have been written by the author. The incredible amount of work done by composer John Oswald in his "Plunderphonics" pieces [9], where he combines by hand short samples of sound [10], cannot be reproduced so easily. CSS however leads to other interesting compositional possibilities, which can be explored quite rapidly with *MATConcat*.

3.1. Dedication to George Crumb, American Composer

At a composition master class given by the composer George Crumb a student asked about the influence of world music on his composition. Crumb related a story about how he collected recordings of musical traditions all around the world. Someone specifically asked about American Indian music and he stated he had never heard it.

For this stereo composition[1] a recording of a short movement of Crumb's was used as the target. It is recomposed into three movements, each using a different corpus of recorded American Indian music: a Navajo man and woman singing (45 minutes), three pieces for end-blown flute (5 minutes), and group dances of different tribes (53 minutes). The target and corpora were analyzed at several different resolutions to produce many sound files, which were then arranged to form each movement.

3.2. Gates of Heaven and Hell: Concatenative Variations of a Passage by Mahler

The dramatic percussion crescendi in the final movement of Gustav Mahler's second symphony, ([11], measures 191–193, Figure 4) is said to signify the gates of hell opening. These short variations (1–3 minutes in duration) explore this brief passage, using five versions by different conductors. Each variation uses a target or corpus created from one or several of these renditions. The targets are sometimes the unmodified or even reversed originals. But to create more complex forms than the crescendo and decrescendo, the renditions themselves are chopped up and rearranged.

All movements explore the possibilities of CSS, and its application to composing variations of a theme. What is very unique about most of the movements is that they do not sound electronic or tampered with, and none are really recognizable. The Bach Partita for solo flute used in one variation is completely dissolved, but its acoustic essence remains. This poses interesting questions for the legality of such timbral appropriation.

4. CONCLUSION

Through the sound examples and compositions created, *MATConcat* demonstrates that this relatively simple implementation of CSS, compared with machine listening and score following, 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.

[1] Premiered at the 2004 International Computer Music Conference.

In a sense, the version of CSS implemented by *MATConcat* is just granular synthesis [12] with grains selected from sample data by matching features. Thinking of the algorithm in this way leads to interesting ideas for extensions: parameter envelopes, variable window-sizes and grain delay, pitch-synchronous and asynchronous synthesis, grain spatialization, etc. For instance, one could specify strict thresholds and gradually relax them, or suddenly change them. One could also specify fades between any number of corpora. These will be explored in future work.

Many improvements will be made to *MATConcat*, especially increasing the dimensions of the feature vector, and expanding the list of synthesis options. There are many more feature measures than the six currently implemented; and their use will serve to further characterize the frames. Future work will implement the features of the MPEG-7 audio framework standard [13]. These extensions will further open up the interesting avenues for creative concatenative composition.

MATConcat, and the sounds and compositions mentioned above, are available for free at `http://www.mat.ucsb.edu/~b.sturm`.

5. REFERENCES

[1] R. Silver, "Photomosaics," `http://www.geochron.co.uk/photomosaics.asp`, 2003.

[2] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*, 1996, vol. 1, pp. 373–376.

[3] S. Hazel, "Soundmosaic," `http://www.thalassocracy.org/soundmosaic/`, 2001–2003.

[4] A. Lazier and P. Cook, "Mosievius: Feature driven interactive audio mosaicing," in *Proc. of Int. Conf. on Digital Audio Effects (DAFx-03)*, London, 2003.

[5] D. Schwarz, "A system for data-driven concatenative sound synthesis," in *Proc. of Int. Conf. on Digital Audio Effects (DAFx-00)*, Verona, Italy, 2000.

[6] D. Schwarz, "New developments in data-driven concatenative sound synthesis," in *Proc. Int. Computer Music Conference*, 2003.

[7] A. Zils and F. Pachet, "Musical mosaicing," in *Proc. of Int. Conf. on Digital Audio Effects (DAFx-01)*, Limerick, Ireland, 2001.

[8] A. Bregman, *Auditory Scene Analysis*, MIT Press, Cambridge, Massachusetts, 1990.

[9] J. Oswald, "Plunderphonics," `http://www.plunderphonics.com/`, 2003.

[10] K. Holm-Hudson, "Quotation and context: Sampling and john oswald's plunderphonics," *Leonardo*, vol. 7, pp. 17–25, 1997.

[11] G. Mahler, *Symphonies Nos. 1 and 2 in Full Score*, pp. 322–325, Dover Publications, New York, 1987.

[12] C. Roads, "Granular synthesis of sound," in *Foundations of Computer Music*, C. Roads and J. Strawn, Eds., pp. 145–159. MIT Press, Cambridge, Massachusetts, 1985.

[13] J. Martínez, "Mpeg-7 overview," `http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm, 2003.`