MATConcat: An Application for Exploring Concatenative Sound Synthesis Using MATLAB

Bob L. Sturm
Graduate Program in Media Arts & Technology (MAT)
University of California, Santa Barbara
b.sturm@mat.ucsb.edu

Abstract

MATConcat is a MATLAB application for exploring concatenative sound synthesis. Using this program a sound or composition can be concatenatively synthesized with audio segments from a database of other sounds. The algorithm matches segments based on similarity of specified feature vectors, currently consisting of six independent elements. Using MATConcat a recording of Mahler can be synthesized using recordings of accordion polkas; howling monkey sounds can be used to reconstruct President Bush's voice. MATConcat has been used to create many interesting and entertaining sound examples, as well as two computer music compositions. This application can be downloaded for free from `http://www.mat.ucsb.edu/~b.sturm`.

1 Introduction

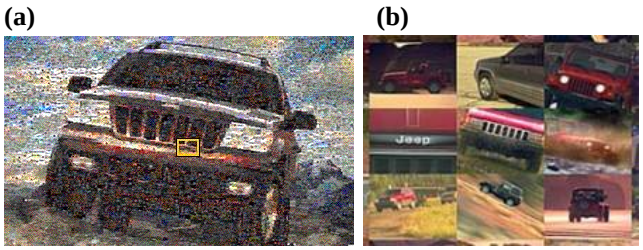

Figure 1: A Photomosaic

A mosaic is a picture assembled by smaller pieces that contribute to the overall perception of an image. Close up the picture isn't 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 (Silver 2003). A simple algorithm selects picture-tiles that are most similar to portions of the original image.

A method similar to photo-mosaicing exists in the synthesis of speech, called 'concatenative speech synthesis' (Hunt and Black 1996). 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" (Hazel 2003; Lazier and Cook 2003; Schwarz 2000, 2003; Zils and Pachet 2001). As in photo-mosaicing, a 'target' sound is approximated by sound samples from a 'corpus.'

Schwarz (2003) 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 available database. Zils and Pachet (2001) proposes a similar method for creating "musaics," but includes specific constraints, such as pitch and percussive tempo.

Creative applications of this work is minimal, and software for exploring these techniques isn't available. The author decided to create an application to explore the compositional usefulness of concatenative synthesis. *MATConcat* is a MATLAB application for exploring concatenative sound synthesis using six independent matching criteria. The application is wrapped in a graphical user interface (GUI), which makes the process and feedback much easier to work with. Using this program a sound or composition can be concatenatively synthesized using audio segments from a corpus database of any size. This application has so far been used to create many intriguing sound examples, as well as two compositions for CD.

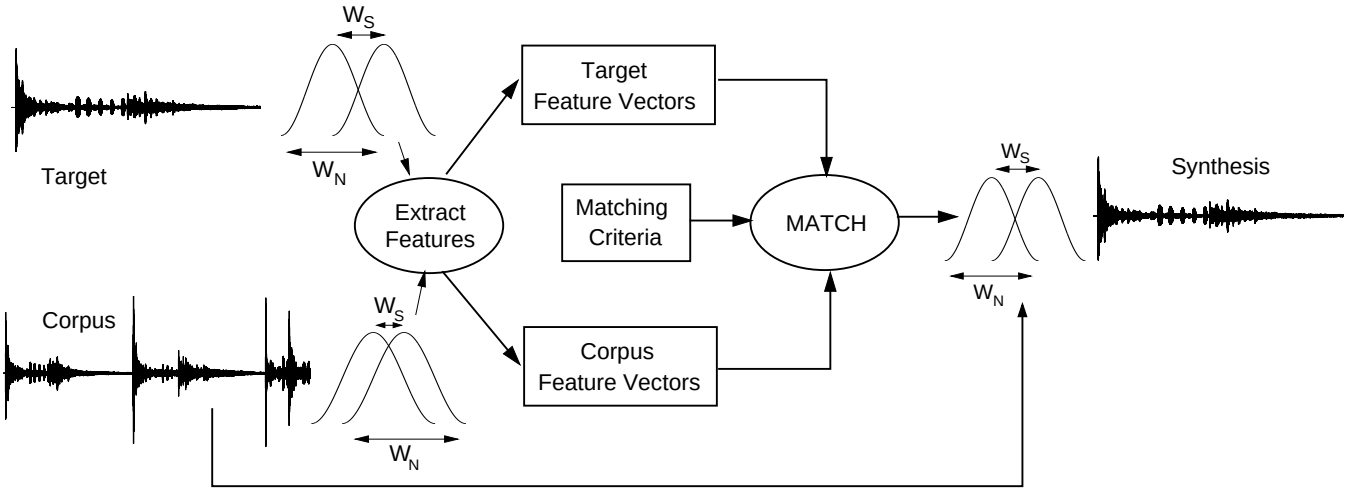

Figure 2: Algorithm of MATConcat

2 Methods

The techniques used in *MATConcat* are much more simplistic than in Schwarz (2000) or Zils and Pachet (2001). Figure 2 displays the algorithm used. Instead of segmenting the audio using a 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 window of the sound, which serves to characterize it. The analysis database of sounds used in the synthesis is called the corpus, which can be several seconds to hours long. The sound being approximated is called the target. Table 1 shows the current dimensions of the feature vector and interpretations of each component.

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

Table 1: Current Feature Vector Elements

Iterating through the frames of the target analysis, optimal matches are found in the corpus database using the chosen matching parameters within specified limits. 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 that of the target analysis frame; and from these matches pick the corpus frame that is within $\pm 5\%$ of the target analysis frame RMS. 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 audio-file and written into the target synthesis according to the settings given in 'synthesis parameters,' e.g. window 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 window hop for the target analysis and synthesize it with a larger hop size. This will make the target synthesis longer than the original. For instance in Figure 3, the windows at the top left shows 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 target sound is synthesized using a window skip of 1024, its total duration will be four times the original duration.

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

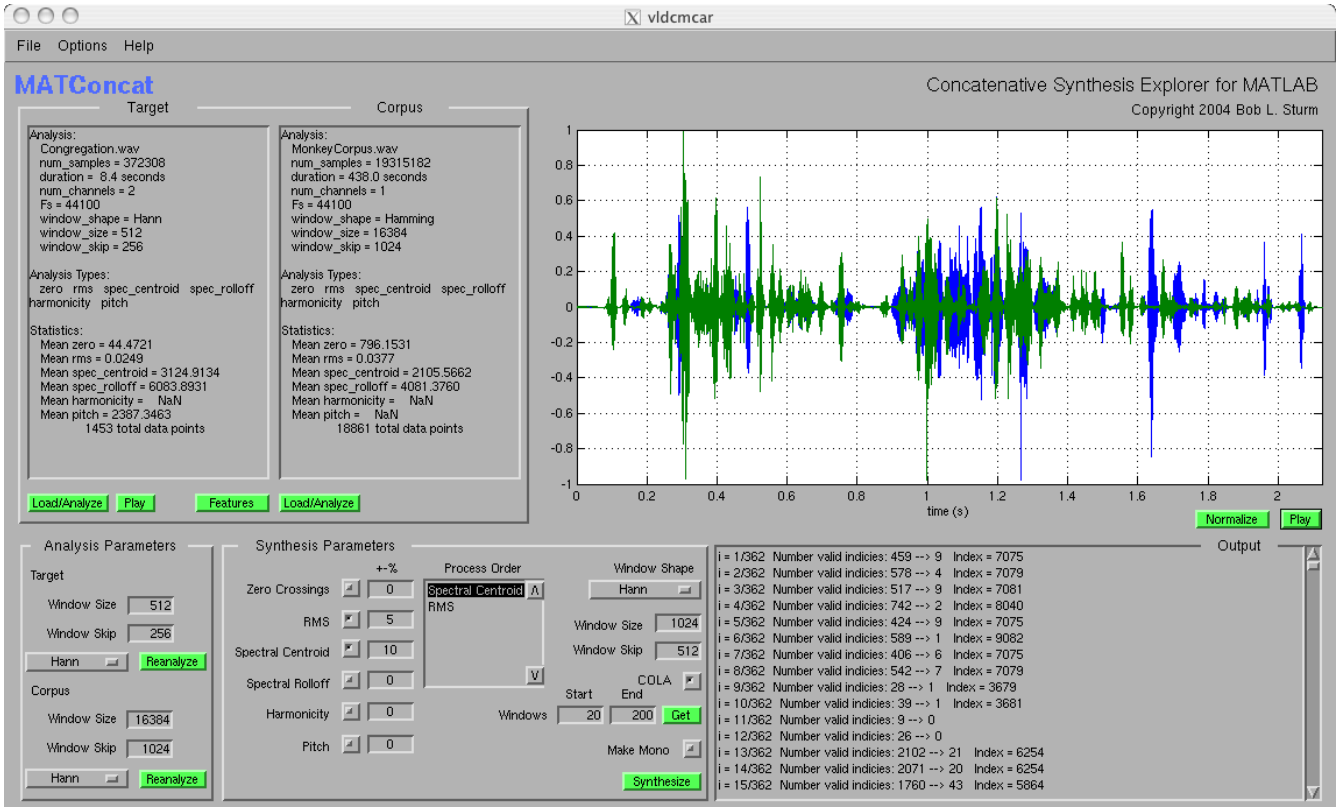

Figure 3: Screenshot of MATConcat

forced, or the previous match is extended to fill the gap—depending on specified synthesis options.

There are currently six synthesis options that affect the process. Checking 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 over-ridden 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 the selected matching criteria. If no matche 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 grains are suddenly expanded to reveal longer phrases. One can also specify to reverse the corpus samples, or convolve the target and corpus frames.

3 Application

Several intriguing sound examples have been created using *MATConcat* from numerous types of targets and corpora. The dramatic percussion crescendo from Gustav Mahler's second symphony (Mahler 1987) has 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 is particularly amazing. The slowly building crescendo is 'aped' by the monkeys, creating a sense of increasing hysteria. At the climax the dominate gorilla grunts as other monkeys cower in submission. Synthesizing this same target using a corpus of John Cage's vocal music creates an entirely different experience, but the impression of Mahler's crescendo remains.

A recording of President George W. Bush has been synthesized by corpora of monkeys, Bach's flute Partita, alto saxophone, and Lawrence Welk. 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 range much of the sibilance and breathiness remains, especially when using the saxophone and flute corpus.

One speech example is greatly time-expanded by synthesizing it using frames of Lawrence Welk sixteen times longer than the analysis frames. In addition to the criteria of matching the spectral centroid and roll-off within $\pm 0.1\%$, the op-

tions of forcing the RMS and extending the matches are specified. This creates a humorous, smoothly moving medley, where one can hear recognizable snippets of tunes, but can't quite place them. Several sound examples can be heard at `http://www.mat.ucsb.edu/~b.sturm`.

Two compositions have so far been written using *MATConcat*. The incredible amount of work done by composer John Oswald in his "Plunderphonics" pieces, where he combines short samples of sound in a similar way as *MATConcat*, cannot be reproduced so easily (Oswald 2003, Holm-Hudson 1997). Concatenative synthesis however leads to other interesting compositional possibilities.

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 from all around the world. When asked specifically about American Indian music he said he had never heard it.

For this composition for CD[1] I used a recording of a short movement of Crumb's 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 with a drum, three pieces for end-blown flute, and group dances of different tribes. Several analysis and synthesis resolutions were used as well as different synthesis options. The produced sound files were then arranged to form each movement.

3.2 "The Gates of Hell: Concatenative Variations of a Passage by Mahler"

The percussion crescendo in the final movement of Gustav Mahler's second symphony, (Mahler 1987, measures 191-193) is said to signify the gates of hell opening. These short variations light-heartedly explore this brief passage, using five recordings by five different conductors. The movement 'The Gates Open' uses sound material from a trumpet and a swinging gate. 'Lix Tetrax' uses Bach's Partita for solo flute. 'Demons Acapella' uses pop vocal samples to synthesize a rhythmic target constructed from short snippets of the crescendo. 'Saxubus' is synthesized from solo alto saxophone with strict matching criteria and no extension of frames if matches are not found. Other movements explore the numerous possibilities of concatenative sound synthesis.

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

4 Conclusion

Through the sound examples and compositions created, *MATConcat* demonstrates that these relatively unintelligent methods of concatenative sound synthesis, compared with machine listening and score following, are very interesting and effective in musical contexts. *MATConcat* serves well as a massive sample-mill, grinding sound into minuscule pieces for reconstitution into familiar forms.

Many improvements will be made to *MATConcat*, especially increasing the dimensions of the feature vector, and expanding the list of synthesis options. Extensions will also be made to enable envelopes of parameters. For instance, one could specify strict parameters in the beginning and relax them throughout the synthesis. One could also specify a fade between two different corpora. These will further open up the interesting avenues for creative concatenative composition.

MATConcat is available at `http://www.mat.ucsb.edu/~b.sturm`.

References

Hazel, S. (2001–2003). Soundmosaic. `http://www.thalassocracy.org/soundmosaic/`.

Holm-Hudson, K. (1997). Quotation and context: Sampling and John Oswald's Plunderphonics. *Leonardo 7*, 17–25.

Hunt, A. and A. Black (1996). Unit selection in a concatenative speech synthesis system using a large speech database. *ICASSP 1*(1), 373–376.

Lazier, A. and P. Cook (2003). Mosievius: Feature driven interactive audio mosaicing. In *Proc. of the 6th Int. Conference on Digital Audio Effects (DAFx-03)*.

Mahler, G. (1987). *Symphonies Nos. 1 and 2 in Full Score*, pp. 322–325. New York: Dover Publications.

Oswald, J. (2003). Plunderphonics. `http://www.plunderphonics.com/`.

Schwarz, D. (2000). A system for data-driven concatenative sound synthesis. In *Proc. of the COST G-6 Conference on Digital Audio Effects (DAFX-00)*.

Schwarz, D. (2003). New developments in data-driven concatenative sound synthesis. In *Proc. of the 2003 Int. Computer Music Conference*.

Silver, R. (2003). Photomosaics. `http://www.geochron.co.uk/photomosaics.asp`.

Zils, A. and F. Pachet (2001). Musical mosaicing. In *Proc. of the COST G-6 Conference on Digital Audio Effects (DAFx-01)*. University of Limerick.