CONCATENATIVE SINGING VOICE RESYNTHESIS

Nuno Fonseca[1]*, Aníbal Ferreira*[2]*, Ana Paula Rocha*[3]

[1]CIIC/ESTG, Polytechnic Institute of Leiria, Portugal [2] FEUP, University of Porto, Portugal [3] LIACC/FEUP, University of Porto, Portugal

ABSTRACT

The concept of capturing the sound of "something" for later replication is not new, and it is used in many synthesizers. But capturing sounds and use them as an audio effect, is less common. This paper presents an approach for the resynthesis of a singing voice, based on concatenative techniques, that uses pre-recorded audio material as an high level semantic audio effect, replacing an original audio recording with the sound of a different singer, while trying to keep the same musical/phonetic performance.

Index Terms— Resynthesis, Concatenative, Singing, Music.

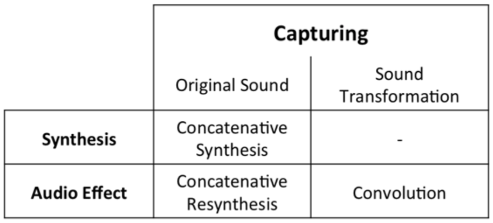
1. INTRODUCTION

The concept of capturing the sound characteristics of existing "things", for later replication, is not new. For instance, a sampler that replicates the sound of a Stradivarius violin, a reverb unit that replicates the reverberation of the Sydney Opera House, a digital audio effects processor that replicates the valve characteristics of legendary audio processing devices. Either capturing the sound created by "something" (e.g. a musical instrument) or capturing the characteristics of the sound transformation of "something" (e.g. acoustic space or sound device), this type of applications end-up having a lot of potential and use.

Sampling and other concatenative synthesis approaches [1] use pre-recorded material of existing music instruments with synthesis. By playing back fragments of the original instrument, a musician can obtain a high quality sound, and it is probably the synthesis choice of most high-end musicians when it comes to the replication of existing musical instruments.

With the increase of the processing power of today's DSP/CPU, reverb convolution that was known for many years is now a reality. By capturing the impulse response of an acoustic space, and using it within a reverb unit, it is possible to replicate the original audio space. The concept of convolution and related techniques (e.g. dynamic convolution [2]) can also be used to simulate existing legendary audio devices (e.g. microphones and audio processing units), allowing the computer or digital processing units to replicate the sound characteristics of different types of audio devices.

Figure 1 shows some applications regarding capturing sound. Capturing original sound and using it for synthesis is quite common (concatenative synthesis). Capturing the sound transformation and using it as an audio effect, is also common (e.g. reverb convolution). But capturing original sound and using it as an audio effect, like concatenative resynthesis, is not so common.



	Capturing	
	Original Sound	Sound Transformation
Synthesis	Concatenative Synthesis	-
Audio Effect	Concatenative Resynthesis	Convolution

Fig. 1 - Applications for capturing original sound or sound transformation

Concatenative resynthesis (also known as audio mosaic) receives an input audio stream, and recreate a completely new audio stream using pre-recorded audio, while keeping the same original semantic audio features. For instance, taking a homemade recording of a trumpet and recreating a new audio stream, using pre-recorded audio recordings (e.g. sound library) of a trumpet player (or any other instrument). An internal sound library consisting of good audio recording means (better acoustic space, better trumpet player, better instrument, better recording material and techniques), could keep the same musical performance, but adding a better sonic quality. This would allow the use of an audio effects unit like a synthesizer, choosing the type of instrument/sound that should be present at the output.

Concatenative resynthesis is a special case of concatenative synthesis. During concatenative synthesis, a target is defined, and pre-recorded audio material is selected and concatenated to obtain such target [3]. Instead of using a symbolic score information (e.g. MIDI file) to generate the target, an input audio file can be analyzed and used as target, allowing the system to be used as concatenative resynthesis system.

This paper presents a concatenative resynthesis approach especially designed for singing voice. The proposed system receives a monophonic singing recording, and outputs an

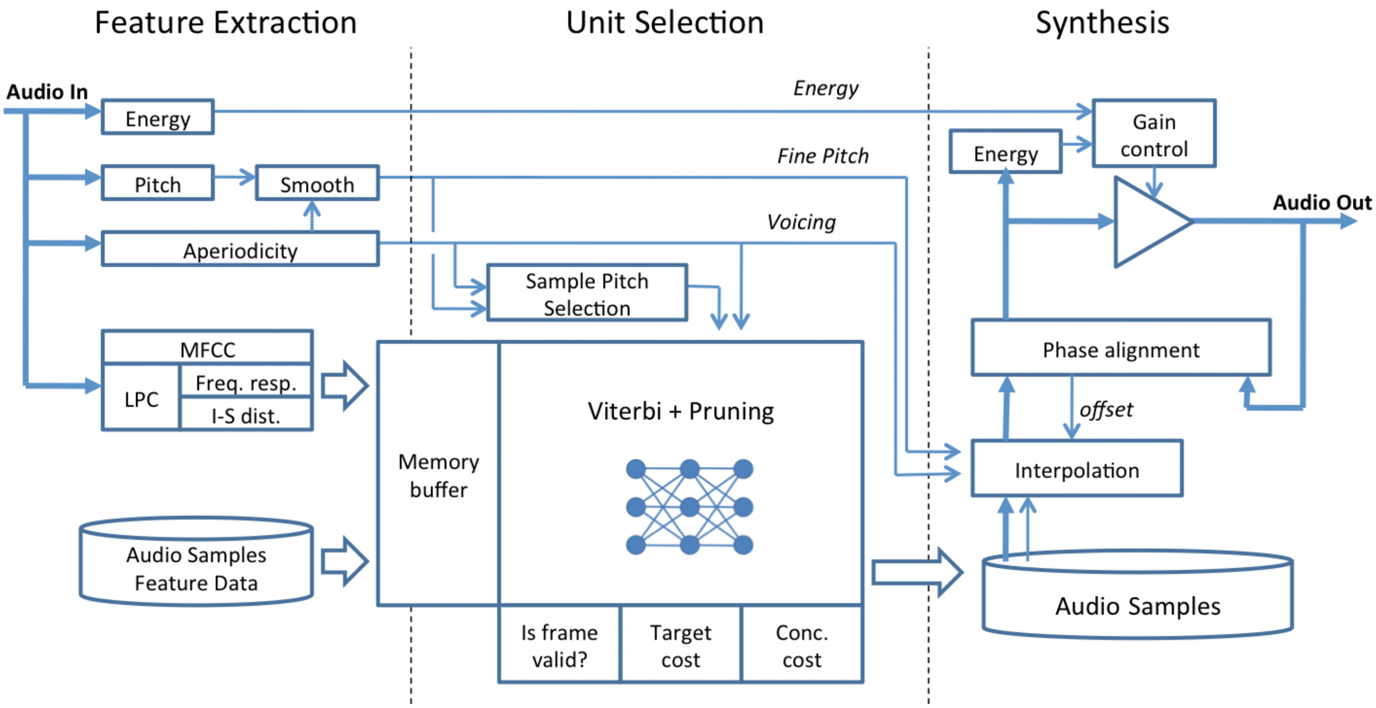


Fig. 2 – Proposed concatenative resynthesis method

audio stream with the same musical performance (music/lyrics) but created with a different voice, based on an internal vocal sound library.

The paper is structured as follows. Section 2 presents the proposed method for singing voice, while its implementation and tests are presented in section 3. Finally, conclusions and future work are presented in section 4.

2. PROPOSED SYSTEM

The proposed system considers three stages: feature extraction, unit selection and synthesis (see figure 2). All processing is done off-line, with frame sizes of 23 ms (1024 samples at 44.1 kHz) with 50% overlapping (hop size of 512 samples), using Hann windows.

2.1 Feature Extraction

The proposed system extracts three types of information: dynamics, pitch and phonetics. The dynamics module (considering the musical concept of dynamics) extracts the energy of the original audio stream. This information will later be used to replicate the same sound intensity behavior.

The pitch module extracts pitch information. Not only the music notes, but also fine pitch information, that will be used to replicate the same note sequence and pitch related effects (*vibrato*, *portamento*, pitch attacks). It is based on the YIN method [4], which outputs not only the pitch value, but also an aperiodicity value that can be used as a confidence measure or a voicing measure.

The phonetic module extracts phonetic related information: MFCC and LPC coefficients. For MFCC, 12 coefficients ($C_1..C_{12}$) are considered (disregarding C_0/energy). For LPC, the input audio stream is resampled at 10 kHz, and 12 LPC coefficients are extracted. The same type of information was previously extracted from the internal sound library.

2.2 Unit Selection

After the feature extraction, the system will then select which audio fragments from the internal sound library should be used during the synthesis. To prevent significant pitch changes (and their impact on sound quality), for each input audio frame, the system only considers internal frames with similar pitch values (± 1.5 semitone). The exception is for unvoiced input frames that present no pitch.

Like many concatenative systems, the proposed approach uses the concept of target cost (TC) and concatenation cost (CC) [5]. Target cost measures how similar are the internal audio fragments regarding the original audio, and concatenation cost measures how well two different internal audio fragments can be concatenated together. The best sequence is the one that presents the lowest overall cost (target costs + concatenation costs), i.e., the sequence that is most similar with the original audio (target cost), but that doesn't present too much abrupt transitions when concatenating audio fragments (concatenation cost).

For the target cost, measuring the difference between the original frame i and an internal frame j, an Euclidean distance is used (Eq. 1) between 4 domains: differences between the MFCC coefficients (Eq. 2), differences between the frequency response of the LPC coefficients (128 bins and log amplitude scale) (Eq. 3), a symmetrical version of the Itakura-Saito LPC distance (Eq. 4) [6], and the aperiodicity/voicing (YIN based) difference (Eq. 5).

$$D_{i,j} = \frac{\sqrt{D_{MFCC}(i,j)^2 + D_{LPC\,resp}(i,j)^2 + D_{LPC\,dist}(i,j)^2 + D_{Ap}(i,j)^2}}{2} \quad (1)$$

$$D_{MFCC}(i,j) = \frac{\sqrt{\sum_1^{12}(c_n^i - c_n^j)^2}}{norm_1} \quad (2)$$

$$D_{LPC\,resp}(i,j) = \frac{\sqrt{\sum_1^{128}(X_n^i - X_n^j)^2}}{norm_2} \quad (3)$$

$$D_{LPC\,dist}(i,j) = \frac{\left[D_{IS}\big(LPC(i), LPC(j)\big) + D_{IS}\big(LPC(j), LPC(i)\big)\right]}{norm_3} \quad (4)$$

$$D_{Ap}(i,j) = \frac{|Ap(i) - Ap(j)|}{norm_4} \quad (5)$$

Other speech related parameters were tested, but the best results were obtained with these, and by combining them into an Euclidean form. Each dimension has a *norm* factor, that is responsible to normalize the weight of each domain, and its value is the mean error difference within each domain. For concatenation cost, responsible to detect abrupt changes between adjacent output fragments (by measuring the difference between two internal frames *i* and *j*), the LPC frequency response (Eq. 3) was used.

The best sequence is then obtained using a Viterbi method that obtains the sequence with the lowest cost. Although heuristics can be used for this search [7], a Viterbi approach is able to get the sequence with the lowest cost, without being stuck at local minimum. To decrease the computational and memory requirements of the Viterbi search, a pruning mechanism is used. For each frame of the original audio stream, the internal frames that present a target cost above the lower 10% range are disregarded. Using this rule, the system reduces by 90% the number of frames to evaluate for each situation.

2.3 Synthesis

Before the concatenation takes place, the chosen frames (from the internal sound library) will probably need a slight pitch modification. Although the "unit selection" process only considers frames with a very similar pitch value (e.g. ± 1.5 semitone), a pitch correction must be done, not only to get the desired music note, but also to get the desired fine pitch value, replicating all pitch related performance from the singer (e.g. vibrato).

The pitch correction is done with interpolation (spline method). By changing pitch, interpolation also changes the time (speed) and formant locations of the original audio, but since the maximum amount of pitch correction is small, the time/formant effects are not relevant (± 1.5 semitone correspond to ± 9%). Instead of defining the amount of pitch change to be applied to the whole frame, interpolation allows the system to gradually change pitch within the frame, by defining, at a sample level, where the interpolation takes place. Since frames overlap in 50%, this allows adjacent frames to gradually change pitch within the overlapping area. In this scenario, an interpolated pitch smoothing is created, where each frame will gradually change pitch from p1 (desired pitch value of the previous frame), to p2 (desired pitch of the current frame), to p3 (the desired pitch of the next frame). Figure 3 shows an exaggerated example of what can be done overlapping two frames of different pitch values (no interpolated pitch smoothing) versus using an interpolated pitch smoothing.

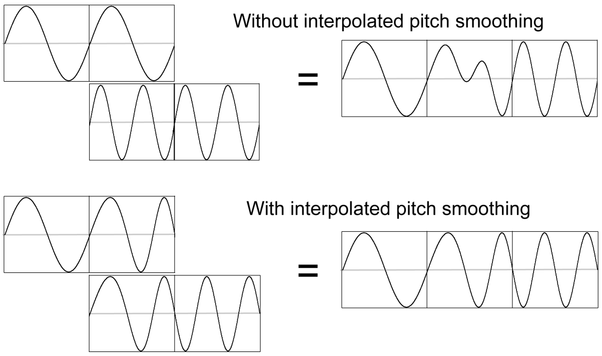

Without interpolated pitch smoothing
=
With interpolated pitch smoothing
=

Fig. 3 - Overlapping frames (Hann window) without interpolated pitch smoothing (top) vs. with interpolated pitch smoothing (bottom)

During the overlapping of frames, phase artifacts may occur, because of phase differences between frames. To prevent such artifacts, a phase alignment method is used. By applying an offset between 0 and *n* samples (being *n* the number of samples that correspond to the period of the pitch value), a correlation vector is obtained and the best offset value is chosen and applied.

Finally, to replicate the original dynamics behavior, based on the original frame energy value and the energy value of the chosen frame, a gain factor is applied to the frame, making sure that a similar energy level is obtained.

2.4 Overcoming Time Differences

Although the system works at a frame level – each input frame is replaced by an internal frame – there are two practical circumstances that alter this concept. In the synthesis module, by changing pitch, the system may require more or less samples. For instance, to output a sequence of 20 frames that need their pitch slightly increased, the system may require 21 frames from the internal sound library. Also during synthesis, to align phase between frames, the system may add a slight offset. All these time shifts may create an issue, because unit selection does not take these events into consideration. To overcome such time differences, without heavily increasing the complexity of the unit-selection process, the system implements a conservative approach, considering that the worst scenario is the one that will occur.

Considering the Maximum Phase Offset (MPO) as the maximum amount of offset that might occur during synthesis (measured in frame size, and usually with values <1), and the Accumulated Time Shift (ATS) as the amount of time shifts that were accumulated in past frames (also measured in frame sizes) within the current audio fragment due to pitch changes, the unit selection module will consider Eq. 6 and Eq. 7 as target cost (TC) and concatenation cost (CC) respectively. Eq. 6 considers that target cost is the worst scenario regarding the MPO, and considering the ATS, where TC' represents the previously defined target cost (Eq. 1). Concatenation cost (Eq. 7) considers the worst scenario regarding the MPO, also taking ATS into consideration, where CC' represents the previously defined concatenation cost (Eq. 3).

$$\begin{aligned} &TC(i,j) \\ &= Max[TC'(i,j+ATS), TC'(i,j+ATS+MPO(j))] \end{aligned} \tag{6}$$

$$\begin{aligned} &CC(i,j) \\ &= Max\big[CC'(i+ATS,j); \\ &CC'(i+ATS+MPO(i),j); \\ &CC'\big(i+ATS,j+MPO(j)\big); \\ &CC'\big(i+ATS+MPO(i),j+MPO(j)\big)\big] \end{aligned} \tag{7}$$

Since Eq. 6 and Eq. 7 require fraction (non-integer) values (e.g. calculating the target cost between original audio frame 24 and internal frame 467.25), the system uses a linear interpolation to obtain fraction values.

3. TESTS

The proposed system was implemented in MATLAB (except the unit selection module that was implemented in C). For the internal sound library, pre-recorded material from a commercial sound library [8] was used, created with a female solo voice, and covering 46 words, with a different recording for each music note within the singer's range.

The 46 used words (with durations up to 9 seconds) were: *Bene, Breathe, Close, Dark, Death, Domine, Dream, Drown, Im, Fall, Fire, Fly, Gaia, Grass, Hasan, Hate, How, In, Len, Love, Luxet, Ly, Mei, Ness, Of, Ooze, Pray, Priest, Row, Ruins, Run, San, Sing, So, Soft, This, True, Uram, Ventius, Ver, Vosh, Fortuna, From, Gravis, Is, Rain, The*.

The system was tested with song fragments of well-known female singers, singing "*a cappela*":

- Amazing Grace - LeAnn Rimes (0:16)
- Bohemian Rhapsody - Lauryn Hill (0:11)
- Frozen - Madonna (0:15)
- I Will Survive - Diana Ross (0:10)
- Tom's Diner - Susanne Vega (0:04)
- Whenever - Shakira (0:06)

Both inputs and output audio files can be accessed at http://www.estg.ipleiria.pt/~nuno.fonseca/papers/dsp2011/.

Although still presenting audio artifacts (mainly due to the concatenation transitions), the system is able to generate an audio stream that significantly replicates dynamics, pitch and phonetics. Within these three domains, phonetics is the one where the system presents a bigger imperfection during replication, which was already expected due to its complexity.

4. CONCLUSIONS

The paper presents a resynthesis approach for the singing voice, based on concatenative techniques. By combining both music and speech techniques, this analysis/synthesis approach allows it to replace a singing recording with audio from another singer (present at the internal sound library), acting as a high level/semantic audio effect. Besides its application as an audio effect, the same system can also be used as a better user interface for singing voice synthesizers. In the future, work will be done with two main goals: to increase the audio quality of the output results (mainly decreasing the existing artifacts and improving phonetic intelligibility); and to add real-time support to the system, allowing its use on live audio scenarios.

5. REFERENCES

[1] D. Schwarz, "Current research in concatenative sound synthesis,". Proceedings of ICMC, Barcelona, Spain, 2005.

[2] M.J. Kemp, "Analysis and Simulation of Non-Linear Audio Processes Using Finite Impulse Responses Derived at Multiple Impulse Amplitudes", in proc. 106th AES convention, May 1999.

[3] D. Schwarz, "A System For Data-Driven Concatenative Sound Synthesis", Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00), Verona, Italy, December, 2000.

[4] A. Cheveigné, and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music", The Journal of the Acoustical Society of America, Vol. 111, No. 4. (2002), pp. 1917-1930.

[5] A.J. Hunt, and A.W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database", ICASSP96, volume 1, 7-10 May 1996, pp. 373 – 376.

[6] F. Itakura, and S. Saito, "A statistical method for estimation of speech spectral density and formant frequencies", Electronics & Communications in Japan, 53A: 36-43, 1970.

[7] N. Fonseca, and A. Ferreira, "Singing Voice Resynthesis Using Vocal Sound Libraries", proc. 13th International Conference on Digital Audio Effects (DAFx-10), September, 2010, Graz, Austria.

[8] EASTWEST, "EW/QL Voices of Passion," information available at http://www.soundsonline.com/Quantum-Leap-Voices-Of-Passion-Virtual-Instrument-pr-EW-174.html, Accessed in Feb. 2011.