Audio Engineering Society

Convention Paper

Presented at the 121st Convention
2006 October 5–8 San Francisco, CA, USA

# Unisong: A Choir Singing Synthesizer

Jordi Bonada[1], Merlijn Blaauw[1], Alex Loscos[1], and Hideki Kenmochi[2]

[1] Music Technology Group, Universitat Pompeu Fabra, Barcelona 08003, Spain
jordi.bonada@iua.upf.edu, merlijn.blaauw@iua.upf.edu, alex.loscos@iua.upf.edu

[2] Center for Advanced Sound Technologies, YAMAHA Corporation, Hamamatsu, Japan
kenmochi@beat.yamaha.co.jp

## ABSTRACT

Computer generated singing choir synthesis can be achieved by two means: clone transformation of a single voice or concatenation of real choir recording snippets. As of today, the synthesis quality for these two methods lack of naturalness and intelligibility respectively. Unisong is a new concatenation based choir singing synthesizer able to generate a high quality synthetic performance out of the score and lyrics specified by the user. This article describes all actions and techniques that take place in the process of virtual synthesis generation: choir recording scripts design and realization, human supervised automatic segmentation of the recordings, creation of samples database, and sample acquiring, transformation and concatenation. The synthesizer will be demonstrated with a song sample.

## 1. INTRODUCTION

Traditional approaches to choir synthesis have been typically based on transformation of solo voice recordings, either by morphing with choir samples [1] or by generation of multiple clones of a single voice, each of them with small and uncorrelated time, pitch and timbre variations added with the aim of emulating different singers of the choir [2,3]. This kind of approach results in a very artificial sound which comes nearest to chorus than to choir. Other approaches have been based on concatenation of actual choir recordings snippets [4]. These snippets are most of the times compound of isolated allophone samples which, when concatenated, make phonetic transitions unnatural and phrases unintelligible.

The system we present in this article, although grouped with those voice synthesizers based on the concatenation of samples recorded from real choir performances, presents two main differences: it uses sample transformation to cover the performance space, and it takes into account not only isolated allophones but also di-allophones or co-articulations between allophones. The aforementioned di-allophone consideration lengthens significantly the recordings required to cover most phonetic and musical contexts, an issue than might be critical for those synthesis engines that do not perform sample transformation and simply concatenate raw samples. In contrast to those, Unisong restricts its recordings to “a cappella” unison singing of single choir sections and works with a set of algorithms for transforming the acoustic material that allow simplified scripts.

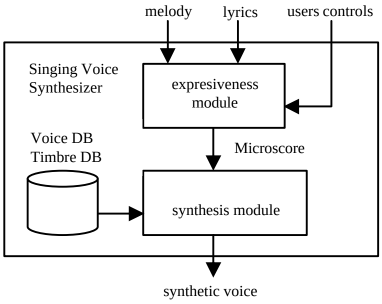

Figure 1 Block diagram of the singing synthesizer

The choir synthesizer we present in this article stands on a solo singing synthesizer which has been presented in previous articles [5][6][7]. Just like the previous synthesizer, the system generates an artificial singing performance out of the musical score and the phonetic transcription of a song by taking audio snippets from a real singer recording database. However, the original system has been tuned to work with choir singing audio data. Main modifications required for the transit focus on MicroScore creation and on sample transformation.

## 2. CHOIR DATABASE

Although the choir database format is basically the same for the solo database, the process of recording and editing present some peculiarities that are exposed next.

### 2.1. Recording Scripts

Recording scripts consist of meaningful sentences taken from a collection of digital books. Script text is chosen to maximize the number of different phonetic contexts while minimizing the number of required sentences. Resulting scripts are to cover 99.9% of most frequently occurring allophone combinations. Sentences are required to be sung at constant pitch and tempo values at each round.

The choir used to build Unisong prototype grouped two sections of 10 males and 10 females and each of them recorded the scripts separately, at 3 pitches and 2 tempos. Each section was conducted by the director of the choir, being him the only one provided with headphones to hear the music background. Ideally however, all section singers should be provided with headphones to avoid slight deviations from the original tuning and tempo and ensure singers are well synchronized in time.

Moreover, we chose Latin as the language of synthesis. For this reason we made up scripts using Catalan language, which is Latin root, and map it to the Latin phonetics [9] using the SAMPA alphabet notation [10].

### 2.2. Segmentation

An Automatic Speech Recognizer (ASR) toolkit was used a phonetic aligner between the recorded audio and the phonetic transcription in order to perform the automatic segmentation [7][8].

Once the onset times of the different phonemes are estimated with the ASR alignment, segmentation rules, which depend on the first (left-hand-side) and second (right-hand-side) phoneme in the phoneme-to-phoneme transition, are applied to obtain the articulation begin, end and middle markers. The middle marker is typically placed at the onset of the second phoneme in the articulation and is used to time-synchronize phoneme onsets with note onsets at synthesis. In cases where the onset of phonemes in an articulation is not clear because of slight differences in timing between members of the choir, an average or middle is used.

Although the ASR and the segmentation rules were both tuned for a single speaker, when applied to choir vocal signals, results were good enough to speed up the database creation process significantly.

## 3. SYNTHESIS

### 3.1. Choir Mode Score Creation

Like our solo singing synthesizer [8], our choir singing synthesizer first builds an internal score from user input before actually synthesizing the output signal. The input used to generate this internal score is typically notes with corresponding lyrics and a set of control curves, which can either be given directly by the user or derived from an expressive model. The generated score will be a sequence of references to unprocessed samples from the database and corresponding sample transformation to achieve the desired synthesis output.

Compared to the internal score of the solo singing voice synthesizer, the main difference is that the segments of the choir synthesizer's internal score are allowed to overlap. This is a direct consequence of the way the choir synthesizer concatenates samples. Our current implementation stores the internal score in a non-overlapped manner, with one segment after the other.

To allow at the most two segments to overlap, each segment also stores its true, overlapped begin and end in time relative to the synthesis output.

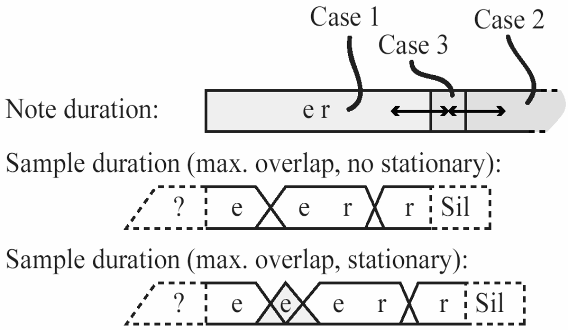
Case 1
Case 3
Case 2
Note duration:
e r
Sample duration (max. overlap, no stationary):
? e e r r Sil
Sample duration (max. overlap, stationary):
? e e e r r Sil

Figure 2 Score creation sketch

Another major change compared to the solo singing voice synthesizer is the way the segments are fit into the given notes. Ideally an overlap of 100 milliseconds is used between segments. This maximum value is limited by the available frames of the phoneme samples on either side of the segment joint.

To fit a given segment sequence into specific note duration a number of different cases depending on the duration of the sequence of segments and the note duration must be considered. Because the total duration of the segment sequence with the biggest possible, up to 100 milliseconds overlaps is dependent on whether or not a stationary is included in the sequence (see figure 2), there are three basic cases: the note duration is smaller than or equal to the sequence duration without stationary, the note duration is bigger than or equal to the sequence duration with stationary or the note duration is somewhere in between the first two. In the first case the sequence can be made to fit by first reducing the amount of redundant ("stable") frames used of a sample and later, to further reduce the segment's duration, time-compress the non-redundant ("non-stable") frames of the samples. In this case the overlap between segments may have to be reduced as the fitting procedure may resulting in less frames being available on the sides of the segment joint. In the second case the sequence can be made to fit the note duration by inserting an arbitrary duration stationary. In the third, infrequently occurring case, the sequence duration must be increased by reducing the overlap between segments.

## 3.2. Samples Transformation and Concatenation

The spectrum of a choir recording is much more complex to the one of a voice solo. Strong amplitude and frequency modulations occur as the result of the harmonics produced by each singer overlapping together. Most voice transformations techniques are devoted to voice solos, therefore not suitable for modifying choir recordings, and need to be adapted to the specific characteristics of those signals.

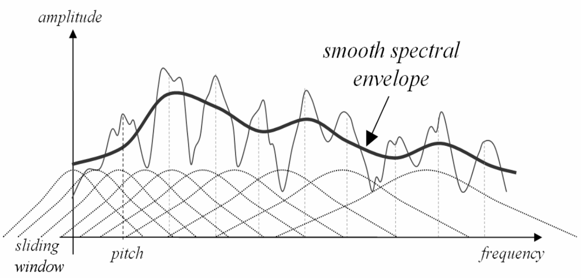

Figure 3 Spectral envelope estimation with a sliding window

In our case, the samples to be transformed are fragments of sentences sung at constant pitch, tempo and loudness values. Given these conditions, we assume that the spectra of quasi-voiced segments can be modeled as a sequence of harmonic peaks, each of them being the result of the superposition of several partials, one per singer, at similar frequencies but varied amplitudes. Besides, we assume as well that the sung pitch is exactly the one specified by the recording scripts, and that the spectrum matches a perfectly harmonic structure.

As in [8] transformations are obtained by non-linearly equalizing and scaling the spectrum, so that around each harmonic frequency the local polar spectrum behavior is not modified but shifted to new amplitude, frequency and phase coordinates. In the case of transposition, the phase modification applied to each harmonic is computed as the difference between the phase of two ideal sinusoids rotating at the original and the transposed frequencies respectively, accumulated for each frame, thus

$$\Delta j_i = 2p \cdot pitch \cdot (i+1) \cdot (transp-1)\Delta t \qquad (1)$$

where $i$ is the harmonic index (starting at 0, the fundamental), $transp$ the linear transposition factor and $\Delta t$ the time distance between consecutive frames.

Similarly to traditional time-domain methods, time-scaling transformation is obtained by repeating or dropping frames. When the frame sequence is not contiguous, a phase difference $\Delta q$ is added to the accumulated propagation phase $\Delta j$, computed assuming linear frequency evolution for each harmonic

$$\Delta q_i = 2p \cdot pitch \cdot \left(i+1\right) \cdot transp \cdot \Delta t \qquad (2)$$

In transposition modifications, timbre is reasonably preserved to a certain extent by an adaptive filter computed, for each bin, as the difference between the amplitude spectral envelope values at the transposed and original frequencies. The spectral amplitude envelope is calculated at each frequency bin as the energy of a window whose size depends on a logarithmic frequency scale (short window at low frequencies, long window at high frequencies), as shown in figure 3.

Using these techniques, we are able to transpose, time-scale and equalize the samples in the database with high quality in a small transformation range. Once samples have been appropriately transformed, the final synthesis step is to concatenate them by spectral cross-fading, achieving smooth timbre transitions thanks to an adaptive filter obtained by means of comparing the smooth spectral amplitude envelopes estimated at the joint boundary frames.

## 4. CONCLUSIONS

Unisong has proven able to generate synthetic results with characteristics which resemble the ones of a real choir with smooth transitions. Compared to the approach in [4] we are able to improve the intelligibility aspect (co-articulations are embedded in samples) and reach nearly similar sound quality (transformations degrade signal). Further improvements on the quality of our transformations will be carried out in future research.

## 5. ACKNOWLEDGEMENTS

We would like to thank all the people in the choir La Fuga dels Lluïsos de Gràcia in Barcelona for their kind support and collaboration.

## 6. REFERENCES

[1] Bonada, J., “Voice Solo to Unison Choir Transformation”, AES 118th Convention, Barcelona, Spain, May 2005

[2] Schnell, N. Peeters, G., Lemouton, S., Manoury, P., Rodet, X., “Synthesizing a choir in real-time using Pitch Synchronous Overlap Add (PSOLA) “, ICMC International Computer Music Conference, Berlin, 2000

[3] ‘Clone Ensemble’, www.cloneensemble.com

[4] East West’s ‘Quantum Leap Symphonic Choirs’, www.eastwestsamples.com

[5] Bonada, J., Loscos, A., Cano, P., Serra, X., Kenmochi, H., “Spectral Approach to the Modeling of the Singing Voice”, Proceedings of the 111th AES Convention, New York, USA, Sept 2001.

[6] Bonada, J., Loscos, A., Kenmochi, H., “Sample-based Singing Voice Synthesizer by Spectral Concatenation”, Proceedings of the Stockholm Music Acoustics Conference, Stockholm, Sweden, 2003.

[7] Bonada, J., Loscos, A., Mayor, O., Kenmochi, H., “Sample-based Singing Voice Synthesizer using Spectral Models and Source-Filter Decomposition”, Proceedings of 3rd Intl. Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications, Firenze, Italy, 2003.

[8] Bonada, J., Blaauw, M., Loscos, A., “Improvements to a Sample-Concatenation Based Singing Voice Synthesizer”, Proceedings of AES 121th Convention; San Francisco, 2006.

[9] Covington M. A., ‘Latin Pronunciation Demystified’ Program in Linguistics, University of Georgia, 2005

[10] ‘Speech Assessment Methods Phonetic Alphabet, a machine-readable phonetic alphabet’, http://www.phon.ucl.ac.uk/home/sampa/home.htm