DT2212 Project: Choir Effect Simulation

Xuehua Fu
KTH Royal Institute of Technology
xuehua@kth.se

Sara Nazeriha
KTH Royal Institute of Technology
saranaz@kth.se

ABSTRACT

The idea of this project was to implement a choir effect on a ten seconds long single vocal containing 4 pitches using MATLAB. A single vocal was recorded in a studio and a beating was added to each partial of the vocal, to obtain variations in amplitude and perceive the vocal as a choir. The idea of the algorithm was to apply amplitude modulations to the harmonic partials of the input vocal in the frequency domain. The random beating for each partial was generated from a Gaussian white noise, with a speed proportional to the frequency of the corresponding partial. As a result, ten seconds of vocal with choir effect was generated as an output and the program was able to simulate an ensemble effect for a single vocal with variations in pitch. Another output with same properties was also generated in which the size of the choir effect changed with time.

1. INTRODUCTION

What is a choir? A choir is a musical ensemble of singers. A body of singers who perform together as a group is called a choir or chorus. Naturally, a choir or ensemble sound is created when a group of singers perform together, producing a monophonic or polyphonic melody. In real ensemble sounds, slight differences of pitch, time delay, amplitude and timbre among individual sounds results in a brand new sound with a rich and complex spectrum, rather than the sum of clones of a single sound. Furthermore, on every sustained note, each singer makes small pitch and amplitude variations, even when singing without vibrato [1]. Such random fluctuations in pitch and amplitude cause uncorrelated beatings of harmonic partials, from which the ensemble quality probably derives. The term choir effect refers to the acoustic transformation from a single sound source to multiple sources playing in unison [2], and is to simulate the ensemble quality in this project.

In 1983, Mark Dolson [3] found that the quasi-random amplitude modulation of beating partials alone can cue the perception of ensemble. Based on this finding, Daniel Kahlin and Sten Ternström [4] implemented three approaches to simulate ensembles from a solo signal in real time. The key idea was to imitate the random beating that occurs in ensemble sounds by modulating harmonic partial amplitudes in the frequency domain. A real ensemble sound was analyzed to obtain amplitude data for the modulation.

The goal of this project is to develop a MATLAB program which outputs a simulated choir sound from a single vocal. The program works by modulating the harmonic partials of the original vocal in the frequency domain, to imitate the random beatings existing in real ensemble sounds. Theoretically, the program works for simple vocal inputs with variation in both pitch and vowel.

2. BACKGROUND

Generating a choir from one vocal, namely, achieving the choir effect is always considered a challenging task. A real ensemble sound has abundant and complex features both acoustically and perceptually, significantly distinguished from a single vocal or instrument sound. A common approach used in music production softwares works by duplicating the vocal on parallel tracks and mix them with tiny time shifts. Such an approach, however, lacks important features in ensemble sounds, and usually results in comb filter effects.

Mark Dolson [3] found in 1983 that amplitude modulation of partials leads to the perception of an ensemble sound. This finding is intuitively reasonable since strong and random amplitude variations or beatings occur in real ensemble sounds, and also agrees with the observations made by Ternstrom et al. [1]in their research work. As mentioned by Ternstrom et al., on every sustained note, each singer of a choir makes small pitch and amplitude variations, even when singing without vibrato. The ensemble quality then probably derives from the superposition of fluctuations in pitch and amplitude from all singers.

With reference to Dolson's result, Daniel Kahlin and Sten Ternström [4] implemented three frequency-domain algorithms for analyzing and modulating the spectral levels of the original sound, using authentic amplitude data from a real ensemble sound, and showed that a simulated ensemble effect can be achieved by modulating the partials of a single voice in the frequency domain. In their investigations, Kahlin and Ternström further asserted that pure amplitude modulations alone, however, cannot simulate all aspects of the ensemble sound, because spectrum properties at high frequencies of a choir sound were found to be elusive, where the modulation by beating becomes very rapid.

3. METHOD

A vocal was recorded in a studio with four pitches and one vowel which was used as an input in this project. For implementing the choir effect on the input, MATLAB was

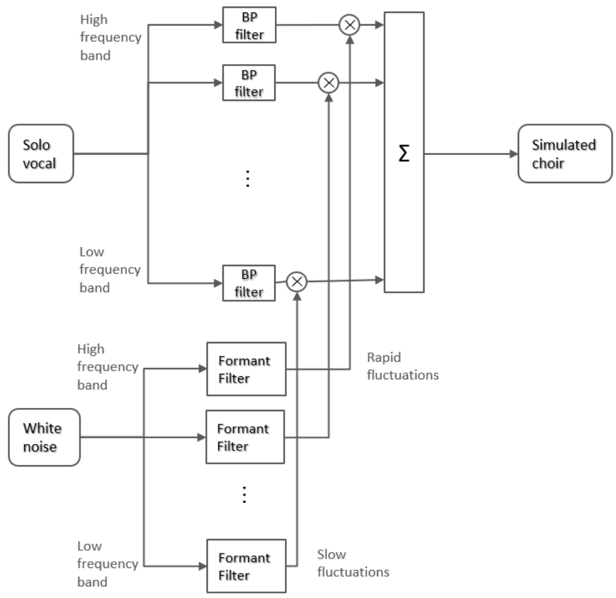

Figure 1: Overview of the algorithm.

used and an algorithm was developed which will be explained in this section.

The algorithm used in this project (Fig. 1) logically follows the common principle of the three frequency-domain approaches in Kahlin and Ternström's study [4]: The single vocal is to be passed through a 23-band filter bank, and each partial is to be mixed with a beating with a frequency in proportion to the band center. The beating is synthesized by a low-passed noise extracted from a Gaussian white noise, through a formant filter, with a resonance frequency proportional to the center frequency of the corresponding band-pass filter. By summing up every partial multiplied with a random beating, a simulated choir sound is obtained.

Equation (1) describes the basic principle of the algorithm. On the left side of the equation two tones at approximately the same pitch (when δ is small) were added. As a result, the audience will perceive the signal as one tone with a beat which is $\cos(\delta x)$ on the right side of the equation. The superposition of two sine waves can therefore be expressed as a single signal multiplied with a "beating", which gives the choir effect to the signal.

$$\sin((\omega + \delta)t) + \sin((\omega - \delta)t) = 2\sin(\omega t)\cos(\delta t) \quad (1)$$

3.1 The band-pass filter bank

Based on Dolson's findings, it is straightforward to apply amplitude modulations on each harmonic partial of the original vocal. The filter bank extracts 23 bands of the original vocal in the frequency domain as the outputs of the 23 band-pass filters. Each band-pass filter is a third-octave filter of order 10, with the center frequency ranging from 100 Hz to 16000 Hz, and overlaps another at the -3 dB cut-off frequencies.

Filter Designer [1] , a graphical user interface on MATLAB is used for the design of the filter bank. The magnitude response of a band-pass filter with a center at is shown in Fig. 2.

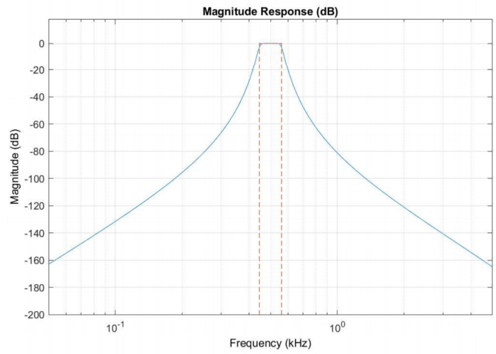

Figure 2: A third-octave filter with a center frequency of 500 Hz. Frequencies are in log scale.

The output of each band-pass filter will be multiplied with a simulated "beating", extracted from an independent white noise coming through the formant filter introduced in Section 3.2.

3.2 The white noise

Consider the case when two tones with very similar pitches are produced at the same time, in sine waves. Beating occurs as a periodic variation in amplitude whose rate is the difference of the two frequencies. The superposition of these two waves can be expressed as Eq. 1 in the time domain. For a small δ, the mix of two tones will be perceived as one with frequency ω, beating at frequency 2δ, while for a large δ, two tones with frequency $\omega \pm \delta$ are heard. The multiplication of the sine waves with frequency ω and a "beating" as a cosine wave with frequency δ is mathematically equivalent to the superposition of two sine waves on the left side of the formula (Eq. refeq:add).

Ternström et al. [1] explored the nature of choir sounds, and found that the randomness in pitch and amplitude is an essential condition for the perception of "ensemble". As each singer constantly makes subtle pitch and amplitude variations on every sustained tone, the ensemble sound obtains a rather random mix of beatings.

Thus, the Gaussian white noise is a reasonable generator of the random beatings on each harmonic partial of the single vocal. The white noise is passed through a bank of formant filters, with resonance frequencies in proportion to the center frequencies of band-pass filter bank.

The term formant filter stands for a second-order resonant low-pass filter, with a Q-factor of 5 (Fig. 3). A scaling factor is set in the formant filters in our program, with a default value of 1%, which is the ratio of the resonance frequency to the corresponding band center. As the beating frequency indicates the extent of pitch disagreement between singers in perception, the scaling factor controls the "size" of the simulated ensemble sound.

[1] A powerful graphical user interface (GUI) in Signal Processing Toolbox™ for designing and analyzing filters.

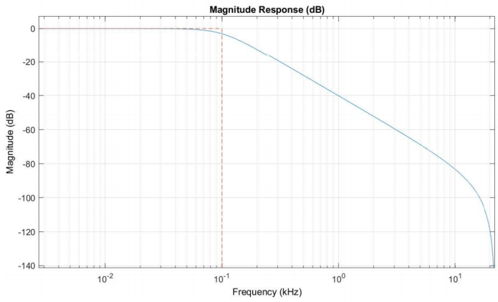

Figure 3: A second-order resonant filter with a resonance frequency of 5 Hz, and a scaling factor of 1%. Frequencies are in log scale.

4. RESULTS

As a result, ten seconds of vocal with choir effect was generated as an output. The output as same as the input had 4 pitches with one vowel and a sampling frequency of 44100 Hz (Sound Example 1). Another output with the same properties was also generated in which the size of the choir effect varied with time (Sound Example 2), with the scaling factor for the resonance frequencies increment by 0.5% per 2.5 seconds from 1% to 2.5%.

To make the simulation more realistic, we made a stereo presentation by modulating two channels of the audio in amplitude separately. The left channel and the right channel are multiplied with two sine waves in anti-phase, respectively, with a period of 2 seconds, so that the volume of two channels vary with time with a time delay. The simulated stereo choir sound is in Sound Example 3.

5. DISCUSSION

Following Dolson's investigations of the nature of ensemble sounds, Kahlin and Ternström invented three frequency-domain approaches to achieve chorus effects. These approaches are theoretically beautiful as well as giving convincing simulation results. In our project, we implemented also a simple frequency-domain algorithm for amplitude modulations of a single vocal to achieve a choir effect as an output.

We made practical attempts to produce more realistic and natural choir sounds by adding the stereo effect, and by changing the scaling factor to make the size of the choir vary with time. Theoretically, the scaling factor controls proportionately the speed of all fluctuations, and hence indicates the extent of disagreement among voices. A large scaling factor indicates a large ensemble of voices in the choir, but is restricted in a certain limit. When the scaling factor is beyond the limit, the choir appeared to be obviously out of tune. In our experiments, the limit is around 2.5%. This phenomenon agrees with the formula (Eq. 1) in which a large δ results in two separate tones. Thus, this frequency-domain approach is theoretically restricted since it is not able to flexibly increase the number of singers in the simulation, and it remains to explore a new method which keeps the choir sound in tune while enabling the variation of the size of the choir.

This project follows the study of Kahlin and Ternström in 1999, and various research projects in investigations of the ensemble effect have occurred since then. Jordi Bonada [5] proposed an approach based on morphing a voice solo with a unison choir recording. This is also an inspiring work since it combines the local spectral features of the choir (amplitude and frequency modulations) with the high-level features of the voice solo (pitch and timbre). Juhan Nam [2] re-explored approaches in the time domain, focusing on the uncorrelated variations of pitch and time delay, which are high-level features according to Bonada. Combining frequency-domain and time-domain approaches might possibly be a direction in future investigations in producing the choir effect, and we are interested in exploring the nature of choir sounds both theoretically and perceptually in the future.

Acknowledgments

This project is under supervision of Sten Ternström. Much appreciation to his guidance and providing the inspiration.

6. REFERENCES

[1] S. Ternstrom, A. Friberg, and J. Sundberg, "Synthesizing choir singing," *Journal of Voice*, vol. 1, no. 4, pp. 332–335, 1988.

[2] J. Nam, "The chorus effect-simulation of multiple sound sources playing in unison," 2009.

[3] M. B. Dolson, "A tracking phase vocoder and its use in the analysis of ensemble sounds," Ph.D. dissertation, California Institute of Technology, 1983.

[4] D. Kahlin and S. Ternstrom, "The chorus effect revisited-experiments in frequency-domain analysis and simulation of ensemble sounds," in *Proceedings 25th EUROMICRO Conference. Informatics: Theory and Practice for the New Millennium*, vol. 2. IEEE, 1999, pp. 75–80.

[5] J. Bonada, "Voice solo to unison choir transformation," in *AES 118th Convention, Barcelona*, 2005.