DIGITISING THE SOUND OF THE ERHU

Minni Ang[1] & Huang Huey Shean[2]
Music Department
Faculty of Human Ecology
Universiti Putra Malaysia
43400 UPM Serdang
tel. 03-89486101 ext 1738
fax. 03-86561689
[1]minni@music.upm.edu.my
[2]shean@music.upm.edu.my

ABSTRACT

Sampling synthesis is one means of preserving authentic replicas of real-world sounds for use in computer generated music. While generally producing the most realistic sounds available through digital means, the technique is however criticised for producing non-controllable sounds which can only be triggered by MIDI messages but cannot be shaped or modified in real-time through controller devices. The digitisation of the sound of the erhu was undertaken using multisampling techniques for a variety of performance techniques, with the aim of creating a soft soundbank for use in MIDI sequence playback. Issues arising included accuracy of intonation and error-correction methods, optimisation of RAM usage, reduction of latency and realism of tone production. These difficulties were addressed and overcome, resulting in the development of a highly realistic soft soundbank of erhu sounds, usable for the playback of MIDI sequences orchestrated for this musical instrument.

Keywords: *Sampling Synthesis, Digital Audio, Waveform Analysis, Music Synthesis, Digital Music, SoundFonts*

1.0 INTRODUCTION

The process and techniques of sampling synthesis have been described in [1]. Briefly, it involves recording examples of real-world sounds, called “samples”, which are then manipulated through digital means to produce different pitches or notes that can be triggered or played back by musical instrument digital interface (MIDI) messages.

The digitisation of the sound of the erhu was undertaken using multisampling techniques for a variety of performance techniques, with the aim of creating a soft soundbank for use in MIDI sequence playback. The erhu is a Chinese musical instrument that is used widely within the Chinese orchestra as well as for solo performances, and is often heard in Chinese pop songs as an accompanying instrument. The aim of creating the soft soundbank of erhu tones is to make accessible to computer musicians the timbres of this ethnic instrument without having to have a real instrument and player present to produce the sounds. The format selected for this purpose was developed by Creative Labs for use with the company's Soundblaster sound cards. Known as SoundFonts [2], this format makes it possible to deliver sample data along with the MIDI notes that comprise a song. The use of MIDI files allows music data to be transmitted rapidly, as the file sizes involved are minimal due to the MIDI file containing only control information as to how the music should be played back [3]. The MIDI file format is thus highly suited to this age of Internet dominance as its bandwidth requirements are minimal [4]. The attaching of sample data to the MIDI file is a relatively new idea, formally defined in the new extended MIDI specification 2.0 [5]. This development ensures that the sounds heard by the end user are identical to what was heard by the content developer when the MIDI sequence was programmed.

2.0 DEVELOPMENT METHODOLOGY

The methodology employed in building the soft soundbank of erhu tones consisted of three main stages. The first stage involved the recording of the original erhu samples, the second stage involved the processing of these recorded samples and the final stage involved building the digital instrument patches and storing these patches in banks, which are directly loadable for playback. (Figure 1).

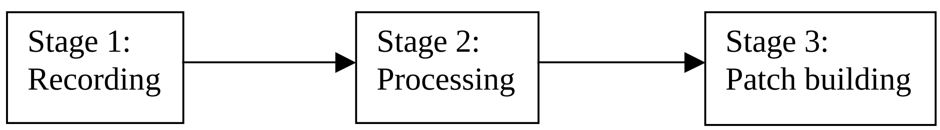

Figure 1. Main stages in the development methodology.

2.1 Recording the original erhu samples

The erhu samples were recorded using an identical pair of AKG C1000S condenser microphones, positioned approximately 15 centimeters behind and in front of the instrument, aimed directly at the sound box of the instrument. This double miking was employed so that two different sets of recorded data could be obtained from one recording session. The purpose of this exercise was to allow for maximum flexibility in later selection of suitable samples for the soundbank. The AKG condenser microphones were selected for their nearly flat frequency response, thereby introducing no artifacts or artificial resonances into the recorded samples. The signals were recorded directly onto the computer hard disk using the recording features available on the digital audio editing software. In order to eliminate any extraneous sounds such as background noise from the computer itself, several seconds of "silence" was recorded at the beginning and end of each recording session. This period of "silence" was used to obtain the background noise profile, which was then filtered out of the recorded samples. The erhu used in the recording session was made in China. Its construction included an authentic snakeskin top and a hexagonal resonance cylinder held by a stem made of rosewood. Its two steel strings were tuned a fifth apart.

The player was seated and the microphones mounted on stationery stands to ensure each note was recorded under uniform distances and conditions. The room in which the recording was conducted was quiet and relatively acoustically dry to ensure a clear recording with minimal reverberance. The player was requested to play all the notes within the range of the erhu using several different techniques (Table 1). To ensure clearly formed and distinguishable samples, the player was requested to pause briefly between each note articulation. A complete range of notes was thus recorded for each of the playing techniques listed in Table 1, and stored for further processing.

Table 1. The different techniques of playing the erhu.

Regular playing techniques	Up bow, down bow, detache, slur, staccato, vibrato, tremolo, trillo, pizzicato
Special effects techniques	Shifting, bird chirping, dog barking, horse neighing

2.2 Processing the recorded samples

The processing of the recorded samples involved several steps. The first step, noise reduction, was explained in section 2.1 above. Upon completion of the filtration of noise, the recorded samples were trimmed to remove unwanted sections at the beginning and end of the recorded sound file. The waveform was then normalised to obtain an optimum sample volume level. No signal processing was applied, such as equalisation or reverb effects, as the natural sound of the instrument was what was required. The samples thus processed were saved as wave files and stored for the purpose of patch building. For the ease of later retrieval and identification, these files were categorised according to playing technique and saved in different folders named accordingly. Each folder thus contained wave files named for their specific musical pitches, for example `C4.wav`[1].

2.3 Patch building

The individual raw samples in the form of wave files, obtained as described in Section 2.2 above, were imported into the SoundFont editing software for the purpose of building digital instrument patches. The steps in creating a digital instrument are described below.

1. The instrument name is defined and specific samples to be used in constructing this instrument are selected from the individual raw samples already imported into the SoundFont editor.

[1] C5 is defined here as the middle C on the piano keyboard, or MIDI note 60, with MIDI note 1 being defined as C1.

2. The range of MIDI notes to be assigned to each individual sample is set. For a perfect representation of the original musical instrument, each MIDI note should be assigned a correspondingly pitched raw sample, however this is not feasible due to memory constraints and in practise a range of notes are assigned to one sample, with a small number of samples of different pitches used to obtain the notes for the entire range of MIDI notes. The intervening notes are thus obtained through pitch shifting.

3. The MIDI root key is set based on the pitch of the imported sample. This step ensures the correct matching of triggered notes with corresponding MIDI notes.

4. Instruments requiring sustain capability, such as bowed erhu, are looped using the SoundFont editor. The waveform at the loop points must have similar peak amplitudes, as well as the shapes and phases, and must be joined smoothly to avoid discontinuities in the audible sustained sound. Furthermore, if the loop points are too close, there will be insufficient variety to make it sound like a real tone, and it may sound artificial. Thus, the loop points have to be as far apart as possible where the above criteria are still satisfied. For instruments not requiring this sustaining capability, such as pizzicato erhu, this step is omitted.

5. Digital effects are applied next. The most important of these, especially in the case of the bowed erhu sound, is vibrato. The SoundFont editor allows the application of vibrato through the use of a low frequency oscillator (LFO). Three parameters need to be set: `delay`, `frequency` and `to pitch`. `Delay` determines the time lapse between the triggering of the sample and when it starts to produce the vibrato effect. `Frequency` specifies the rate of the vibration and `to pitch` specifies its range.

   Other digital effects may also be applied. Besides vibrato, the SoundFont editor provides several different ways of modifying the raw sample sound, including reverb, chorus, pan, volume envelope, modulation envelope, modulation LFO and tuning. In the case of preparing the erhu SoundFont, these were not used as the original authentic sound of the instrument was what was required.

The next step is to define a new melodic preset. In this step, the name for the new melodic preset is specified, as well as its bank number and patch number. The digital instrument to be used under this preset is then selected from the instruments created, as described in the previous paragraphs.

Additional instrument patches are built, repeating the entire process described in this section 2.3. When all the required patches are complete, the entire soft soundbank is saved as a file with the extension `.sf2` which may be then attached to a MIDI file and used as a sound source for MIDI sequence playback. New patches may be added ot the soft soundbank (also known as a SounfFont) at any time through adding to the existing patches using the same procedure.

3.0 RESULTS

A total of 14 digital instrument patches were created for the erhu SoundFont, stored in two different SoundFont banks (Table 2). Various parameter details are in Table 3.

Table 2. Patches created for the Erhu SoundFont.

Bank Number	*Patch Number*	*Patch Name*
000	000	erhu up bow
000	001	erhu down bow
000	002	erhu detache
000	003	erhu pizzicato
000	004	erhu slur
000	005	erhu staccato
000	006	erhu tremolo
000	007	erhu trillo
000	008	erhu vibrato
001	000	erhu shifting
001	001	bird chirping
001	002	dog barking
001	003	horse neighing
001	004	all-in-one

4.0 DISCUSSION

Several technical issues arose during the course of the development of the erhu soft soundbank. These are discussed under the different sub-headings of accuracy of intonation and error-correction methods, optimisation of RAM usage and reduction of latency, and realism of tone production.

4.1 Accuracy of intonation and error-correction methods

One of the first problems faced in the construction of the digital instrument patches was the slight inaccuracies in pitching of the originally recorded samples. The precise nature of computer generated music means that each MIDI note produced must be 100% accurate in terms of its frequency, as the patches created may be used in harmony with other banks and patches. This computer-preciseness of pitching was however not achievable by our human musician, and though the original recording of the raw samples sounded in tune even to the trained ear of the musician, when closely analysed using computer software, the frequencies of the notes played were found to vary by up to 40 cents compared with their theoretical values[2].

[2] 1 semitone = 100 cents. According to the equal temperament scale used in most MIDI instruments, with 12 equal semitones in one octave, the frequency ratio of two adjacent notes must be exactly $2^{1/12}$. As A5 (MIDI note 69) is most often pitched at 440 Hz, this means that theoretical frequencies can be calculated for all MIDI notes.

Table 3. Erhu SoundFont Parameter Details.

Patch	*Sample*	*Root Key*	*Key Range*	*Looping*	
				Loop End	*Loop Start*
Up bow	UbA4.wav	81	G5 – G7	14180	12428
	UbB4.wav	71	C1 - D#5	21428	18484
	UbF#5.wav	78	E5 – F#5	7713	6034
Down bow	DbB4.wav	71	C1 - C#5	29230	24054
	DbD5.wav	74	D5 – G7	28013	23516
Detache	DetA#4.wav	70	C1 - A#4	-	-
	DetD5.wav	74	B4 – D5	-	-
	DetF#5.wav	78	D#5 – G7	-	-
Pizzicato	PizzA#4.wav	70	G4 – A#4	-	-
	PizzD5.wav	74	B4 – D#5	-	-
	PizzE5.wav	76	E5 – G7	-	-
	PizzF#4.wav	66	C1 - F#4	-	-
Slur	SlurC5.wav	72	C1 - C5	17858	14583
	SlurE5.wav	76	C#5 – E5	17656	14671
	SlurG5.wav	79	F5 – G7	18019	13393
Staccato	StacA4.wav	69	C1 - A4	-	-
	StacC5.wav	72	Bb4 – C#5	-	-
	Stac74.wav	74	D5 – G7	-	-
Trillo	MonoT1(C5–D5).wav	72	C1 - E5	-	-
	MonoT3(F#5–G#5).wav	78	F5 – G7	-	-
Tremolo	TremD5.wav	74	C1 - D5	-	-
	TremE5.wav	76	E5 – G7	-	-
Vibrato	DbB4a.wav	71	C1 - C#5	29230	24054
	DbD5a.wav	74	D5 – G7	28013	23516
Shifting	S1-up.wav	62	G#4 – F5	-	-
	S2-down.wav	79	F#5 – G7	-	-
Horse neighing	Neigh.wav	60	C1 - G3	-	-
Dog Barking	02(high).wav	60	C4 – D4	-	-
	01(low).wav	60	G#3 – B3	-	-
Bird chirping	Chirp.wav	60	D#4 – G4	-	-
All-in-one	S1-up.wav	62	G#5 – C6	-	-
	S2-down.wav	79	C#6 – G7	-	-
	Neigh.wav	60	C1 - G4	-	-
	02(high).wav	60	C5 - D5	-	-
	01(low).wav	60	G#4 – B4	-	-
	Chirp.wav	60	D#5 – G5	-	-

The raw sample frequencies thus had to be adjusted using pitch bending facilities provided by digital audio editing software. The frequency analysis facility provided by the audio editor was first used to identify the exact deviation of the frequency from its theoretical value. This amount was then adjusted through specifying the exact quantum of sharpening or flattening required to adjust the pitch to its theoretical value. In this manner, perfectly tuned samples were obtained.

4.2 Optimisation of RAM usage and reduction of latency

Optimisation of memory requirements is always an issue in sampling synthesis and has been discussed in the author's previous paper [1]. Measures taken in the production of the erhu soft soundbank included the use of mono samples (originally recorded stereo samples were converted to mono during the sample processing stage), use of the minimum number samples per digital instrument possible in order to produce a realistic timbre, and looping. It is taken note here that using too few samples per digital instrument is not advisable as the timbre of the sample changes when it is stretched for more than one octave either upwards or downwards, producing an artificial sounding instrument.

In order to make the created SoundFont accessible to even those whose computers may be short on RAM, Bank 000 and Bank 001 were also split into two different SoundFonts and saved as two separate SoundFont files. In fact, if only a subset of the patches created are required and additional RAM is not available, it is possible for the user with access to the SoundFont editing software to further split the SoundFont into different files, each storing only the required patches. The reason for this exercise is because attaching a particular `.sf2` file to a MIDI sequence means that the file will be loaded into the temporary memory of the computer, and although not all the patches available may be required for that particular MIDI file, the entire set will still be loaded. Thus, to avoid loading the RAM with additional unused data, SoundFonts may be created that include only the specific patches required for the piece.

4.3 Realism of tone production

The patches created for the different erhu playing techniques were aimed at creating a realistic simulation an authentic erhu sound. In real life, string players, including erhu players, use a variety of playing techniques within a single piece of music. To simulate this effect using a MIDI sequence requires rapid patch change messages to be sent to the synthesiser device, in our case the soft soundbank. However, it was found with existing computers of up to Pentium-III 450 MHz and 64 MB RAM, these patch changes were not always audible when applied to adjacent notes, resulting in some notes being "dropped" or not heard. However, a reasonably realistic tone was still achievable without articulating every single note the way a live musician would do. Perhaps with the advent of the new generation of superfast computers, a completely authentic computer-generated rendition will become achievable.

5.0 CONCLUSIONS

The digitising of the sound of the erhu produced a total of 14 different digital instrument patches, saved in the SoundFont format. Various issues arose, including accuracy of intonation and error-correction methods, optimisation of RAM usage, reduction of latency and realism of tone production. Of these, perhaps accuracy of intonation was the one most peculiar to the production of ethnic instrument soundbanks, as these instruments are traditionally played not according to western standards of tuning. However, all the difficulties were addressed and overcome, resulting in the development of a highly realistic soft soundbank of erhu sounds, usable for the playback of MIDI sequences orchestrated for this musical instrument.

REFERENCES

[1] Ang, Y.F. & Ang, M. (1999). "Sampling Synthesis Technique applied for the Digital Generation of Musical Tones of Malaysian Folk Instruments." *Malaysian Science and Technology Congress, MSTC'99, Symposium C.* Confederation of Science and Technology Associations in Malaysia (COSTAM), Johor Baru, 6-8 December 1999. Proc.Vol.I. pp. 366-373.

[2] Creative.com. (2000). "Basics of SoundFonts." *Soundblaster Website.* [Online]. Available: http://www.soundblaster.com/resources/read.asp?ArticleID=12

[3] Rumsey, F. (1990). *MIDI Systems and Control.* London: Focal Press.

[4] Ang, M. (2000). "Breaking Internet Audio into Basics." *Jaring Internet Magazine*. 3(8): 51.

[5] MIDI Manufacturers Association. (1999). "Downloadable Sounds, DLS Level 2 Specification." *MMA Website.* [Online]. Available: http://www.midi.org/ about-midi/ dls/ abtdls.htm