From musical analysis to musical expression

by

Oscar Mayor Soto

Submitted in partial fulfilment of the requirements for
the degree of Diploma of Advances Studies
Doctorate in Computer Science and Digital Communication
Department of Technology

Tutor: Dr. Xavier Serra

University Pompeu Fabra

Barcelona, September 2003

Index

Abstract

This document presents an overview of my research work in the field of computer music since I had my first contact with it a few years ago. In the first part, a state of the art of each of the topics I have been involved in is presented, from audio identification to musical instrument synthesis, including tempo tracking and rhythm perception research and some topics on musical performance analysis. Then some articles I have published are presented detailing my contribution and in the final part, some future directions for my ongoing thesis related with performance analysis of the singing voice are exposed.

Keywords

Audio identification, Song matching/alignment, Saxophone, Musical Expression, CBR, HMM, Spectral modelling, Speech synthesis, Instrument synthesis, Singing voice synthesis, Musical expression, Musical styles, Mood emotions, Musical Performance, Performance rating.

Acknowledgements

I would like to thank Professor Xavier Serra, my thesis director and also director of the Music Technology Group, I would like also to give an acknowledgement to all my workmates at the MTG, especially to Jordi, Alex, Lars, Oscar, Pedro, Eloi, Jaume, Jojo, Perfe, Emilia, Fabien, Xavier, Vadim, Enric, Cristina, Joana, Marteen, Martin, Jose, Teresa, Claudia, Carlos, Otto, Pau and people from Yamaha Corp. for their friendship and collaboration. My friends at lunatik and all the people that have played music with me also deserve a great gratefulness.

Chapter 1

Introduction

In this research project, the work already done and future directions to investigate are exposed. Since some years ago until now I have been involved in some fields, always inside the world of computer music. Before that, I'm a musician, I play guitar and saxophone and I've performed in several bands, I'm also a computer engineer. These are the reasons about my interest in getting inside the big field of computer music.

I have also studies of musical theory and harmony related to jazz and modern music and I play the electric guitar and the saxophone and I have performed live with my band in several places.

My first contact with the field of computer music was as a collaboration in a project called SaxEx [ALS97] consisting in giving expression to non expressive performances of jazz classics, performed by a saxophone player, using AI techniques. The method consisted in analyzing the non expressive performances, extract some features like pitch, loudness, attacks, releases and note durations and transform the input sound using Spectral Modelling analysis/synthesis algorithms [SS89] [SB97] to get an expressive performance. The decision of what transformations to apply where decided by a module that had a database with multiple expression performances labelled depending on the type of expression (sad, happy, tender, aggressive,…). Using Case Based Reasoning

techniques (CBR) the system decided which modifications to apply in each situation to give a certain expression to the synthesis. My collaboration consisted in improving the fundamental frequency detection, automatic segmentation into regions of the input sound (note attack, note transition, release) and integration between AI and spectral processing algorithms.

Another project that I have contributed to was about analysis and identification of audio using fingerprinting techniques. The project called RAA (Recognition and Analysis of Audio) consisted in recognizing the songs that are being broadcasted by the radio or TV in real-time and create play lists of the songs played everyday for royalty and property right purposes [BC00]. The system analyzed the songs that were being played in real-time, extracted some spectral parameters (audio fingerprint) and tried to match this extracted parameters with the extracted parameters of the songs in a database of commercial songs. My contribution in this project included the creation of graphical interfaces, integration of modules and development of string matching algorithms. As a continuation of this project I got involved in the rhythm perception and beat tracking field, to apply rhythm features to the identification system in order to search for similar songs in the database. I developed a real-time beat tracking system for polyphonic audio that used multiple hypotheses to decide the beat at each time [May01].

After finishing the beat-tracking system I have been involved until now in a different field, the spectral synthesis, firstly in the development of a wind instrument synthesizer, then in speech synthesis and now in singing voice synthesis.

The development of a real-time midi controlled wind instrument synthesizer, financed by the MOSART network, consisted in the implementation of a software spectral synthesizer that used a database of analyzed sounds that were concatenated and transformed to produce the output synthesis [Haas01] [MBL02]. The input of the system could be a midi wind controller or a midi keyboard for real-time synthesis or an xml file with the score for offline synthesis.

Another project where I got involved was the development of a speech synthesis engine using spectral models. This was a project done in collaboration with TELEFONICA where they gave us a database with recorded samples (phonemes), the prosody and intonation speaking rules and we adapted our singing voice spectral domain synthesis techniques to the case of speech. The results were quite successful.

Finally I've been involved in the field of singing voice synthesis, in the part of applying expressive changes in order to obtain naturalness and expressiveness syntheses [MCBL03]. These expressive transformations range from local changes for getting a more natural synthetic voice to more general and global changes that apply certain emotions (musical style or mood) to the synthetic performance.

My thesis will be oriented inside the field of musical expression for instrument synthesis, specifically for the singing voice, which is the most flexible musical instrument and so the most difficult to imitate. One title for the thesis could be something like "Analysis of the expression, naturalness and meaningful emotions of the singing voice" or "Performance analysis of the singing voice", I plan to improve local and global expression models already developed in the spectral singing voice synthesizer [MCBL03], and continue the research applying expression models and

performance rules like the ones developed in KTH [Fri91] to apply certain mood emotions like anger, fear, happiness or sadness, and also musically meaningful emotions, also called style or genre labels like jazz, blues, opera, rock or country to non-expressive performances. The goal is similar than in SaxEx [ALS97] but the procedure is quite different, here instead of applying previous expressive cases stored in a database, general rules or models are created and applied to the non-expressive performance to give a certain degree and kind of expression. I also plan to develop a singing performance rating scheme to determine how well a user performs a certain piece, based in a standard way of performing it or based in a comparison with a previously performance of a professional singer that the user has to mimic. A performance rating system can have numerous applications like judgement in singing contests, karaoke performance rating or virtual singing education. We will need to analyze the user performance, extract some parameters (expression, execution, accuracy) and compare these parameters with ideal performances. The evaluation of the accuracy of the results given by the performance rating system must be supervised in an early phase by professionals in order to validate the certainty of the report that the system gives to the user, and improve the results in future reports. The most relevant research in this field will be the analysis of the expression in the singer's performance in order to apply and recycle the investigation done in this field, the synthesis of the singing voice, in a coherent way. Once we know what rules and transformations to apply to express a happy or sad mood, we can determine if a singer is expressing happiness or sadness, just analyzing his performance execution. So the expressive transformations in the synthesis of the singing voice and the performance analysis of the singing voice are tightly related.

Chapter 2

Overview of relevant research

2.1 Analysis and Identification of Audio

2.1.1 Introduction

The analysis and identification of audio has taken an important role in the computer music field in the last few years. The analysis has been mostly used for data mining and feature extraction but also for data identification. One of the fields where data analysis is needed is the audio identification and due to the importance and presence that music piracy has acquired in the digital era, where recordings can be easily replicated and distributed, here a monitoring system able to automatically generate a play list of registered songs broadcasted live [CBMN02] or distributed via internet [NMB01] can be a very valuable tool for copyright enforcement organizations and also for record companies. Many techniques can be used to identify audio that is being broadcasted in real-time by a radio or TV station or distributed in a digital medium but the most important ones are Audio Watermarking and Audio Fingerprinting [PAK99]. We can see that each of these techniques has its advantages and its drawbacks.

2.1.2 Audio Watermarking

Audio watermarking is a technique that consists in embedding a mark into the audio signal [BTH96] [GCGBB02]; this mark is the so called watermark. This mark is an audio signal that carries data that can be retrieved from the watermarking signal. The watermark must be:

- Imperceptible: inaudible by the listener.

- Statistically invisible: robust to attacks by pirates resulting in an unauthorized detection or removal. Should be only detectable by the author of the piece or the subject that has introduced the watermark to the piece

- Have similar compression characteristics as the original signal to survive compression/decompression operations.

- Robust to distortion and standard signal manipulation and processing operations (filtering, resampling, compression, noise, cropping, A/D-D/A conversions).

- Embedded directly in the data, not in a header or similar.

- Capable to support multiple watermarking, so a signal can have multiple hidden data.

- Self-clocking: Able to detect the watermark in the presence of time-scale change operations.

The watermark can be exploited by a pirate in several ways:

- Manipulate the audio signal to make the watermark undetectable.

- Add false watermarks (inaudible jamming signals) to make the watermarking scheme unreliable.

The quality of a watermark can be measured by two factors:

- The probability to detect a watermark when one is present (probability of detection)

- The probability to detect a watermark when none is present (probability of a false alarm)

2.1.2.1 Embedding a Watermark

The generic embedding process consists in giving an audio (A), a mark (M) and a key (K), define a mapping function like:

$$A \times K \times M \rightarrow A'$$

In figure 1 we have a diagram of the generic embedding process.

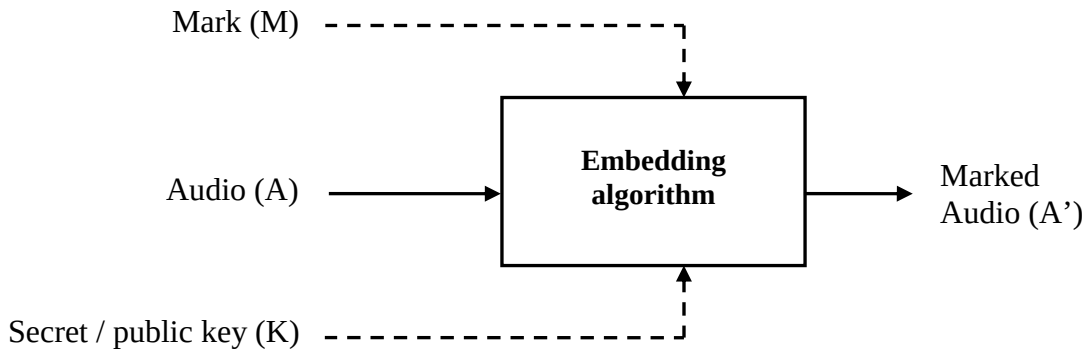

Figure 1: General digital watermark embedding scheme

2.1.2.2 Recovering a Watermark

The generic detection process consists in recovering the mark (M) or some kind of confidence measure which indicates how likely is for a given mark at the input to be present in the audio (A') under inspection.

Figure 2 shows a general diagram of the watermark recovering scheme.

The decoding process could have several utilities:

We take a test audio (A') (with the watermark embedded) and the key (K) and we extract the watermark (M) from the audio:

$$A' \times K \rightarrow M$$

This is the common use of this technique and is called public marking because we don't have to know any secret information like how it was the original audio (A). There is another use of the watermarking technique, commonly known as private marking or semi-private marking in which we take a test audio (A'), the key (K) and the mark (M)

and we try to know if the audio (A') has the mark (M) embedded or not, it's possible to have also the original audio (A) without the mark to help the decoding process:

$$A' \times K \times \{M, A\} \rightarrow \{0,1\}$$

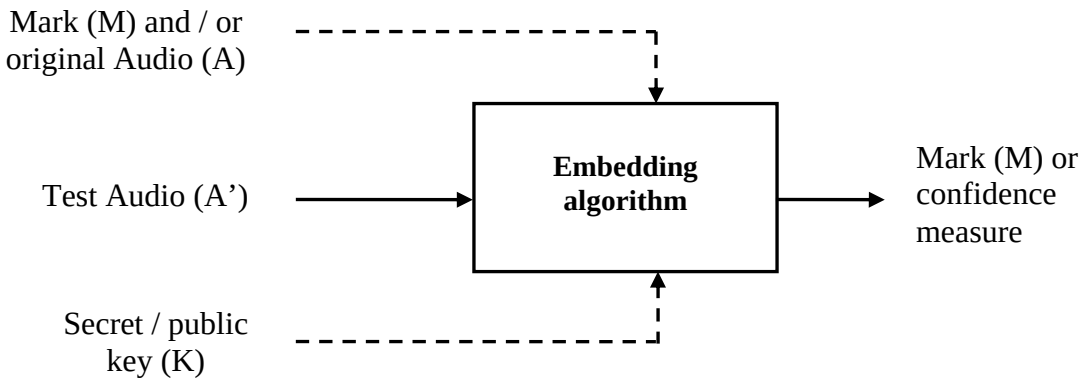

Figure 2: General digital watermark recovery scheme

2.1.3 Audio fingerprinting

An audio fingerprint is a content-based compact signature that summarizes an audio recording. One of the advantages that make fingerprinting a valuable technique is that allows the monitoring of audio without the need of meta-data or watermarks embedded into the audio [CBKH02] [CBGGB02].

Fingerprinting is commonly used instead of embedding a fingerprint in the audio, just as a watermark process, for analyzing the audio and extracting some relevant information (commonly perceptual information) that can be used to identify the audio. This information has not been inserted before in the audio, it consists in perceptual characteristics extracted by an analysis process. The main use of fingerprinting is content-based audio identification; this means to have a database with fingerprints of analyzed songs and when monitoring audio that wants to be identified, extract the fingerprint and try to match this fingerprint with the ones previously calculated in the database.

A system designed to calculate a fingerprint from an audio must be efficient and robust; the requirements depend heavily on the features and characteristics that are extracted from the audio. Here we present a list of some requirements for applications of audio identification:

- Accuracy: the number of correct, missed or wrong identifications (false positives).

- Reliability: In many cases, missing identifications are preferred instead of false positives.

- Robustness: It is important to identify the audio regardless of the level of compression, distortion, equalization, pitching or noise in the transmission channel.

- Security: Vulnerability to cracking or hacking in order to produce a fingerprint that yields to an identification error.

- Versatility: Identify audio regardless the format and the channel (radio, TV, internet) using the same database of fingerprints.

- Scalability: Allow big databases and concurrent identifications

- Complexity: Minimize the complexity of search in the database, extraction of the fingerprint, adding a new song to the DB.

- Fragility: Detect changes in the content but not produced by noise or compression /equalization algorithms.

- Compact: It's important to store the maximum perceptual information in the lower size possible.

2.1.4 Watermarking vs Fingerprinting

There are many applications to watermarking and fingerprinting, although depending on the application, one will be more suitable than the other [GCGB03]. In figure 3 we can compare fingerprinting and watermarking by their advantages and drawbacks.

As it is shown in figure 3 fingerprinting is less vulnerable to attacks and distortions, because these distortions should change the perceptual content of the audio to confuse the system. Fingerprinting requires no modification of the audio content, the algorithm extracts the fingerprint from the audio itself, the advantage is that we don't need to embed a mark in it but the complexity is quite higher and we need a database with a repository of previously analyzed songs, and we can only detect or identify audio that has a counterpart in this repository. Another drawback of fingerprinting is that two perceptual identical copies cannot be identified as different ones, so the message is not independent from the context as in a watermarking scheme. Applications of watermarking are wider than just identification or content integrity of audio, it can also be used to embed any information that we want to associate with a certain audio, so any metadata can be transmitted with a song.

	Watermarking	Fingerprinting
Vulnerability	High (attacks and distortions)	Low
Content modification	Yes	No
Complexity	Low	High
Database	No	Repository with analyzed audio
Identical Perceptual copies	Can be distinguished	Detected as the same
Use	Many uses	Least uses (identification and content integrity)

Figure 3: Advantages and drawbacks of watermarking and fingerprinting.

2.2 Beat tracking and rhythm analysis

2.2.1 Introduction

Musical beat tracking is needed by various multimedia applications such as video and audio editing and synchronization, tempo modification or stage lightning control. Beat tracking in the music field consists in recognizing temporal positions of quarter notes, just as people follows time in music by hand-clapping or foot-tapping. Many systems have dealt with midi as input [DH89] [AD90] [Ros92] [DC00], but the new technology evolution now allows to beat-track polyphonic audio in real-time [RGM94] [GM98] [May01]. In polyphonic audio is where a beat-tracking system is really valuable as it can be applied to the real world in real situations.

2.2.2 Beat-tracking (BPM detection)

Usually beats match with positions where the audio has the maximum energy but this is not always accomplished. Mainly in electronic music or rock music this is usually the case, but in jazz or classical, it is not always true and we have to look for onsets (beginnings of notes) instead of positions of maximum energy. This is more obvious in songs with no percussive instruments at all [GM97], for instance imagine a piano solo piece. First we will define some concepts to clarify more the explanation about beat-tracking:

Note Onset: A note onset corresponds with the beginning of a note, many times the onsets correspond to the beat of the song or multiples or submultiples of it (quarter note, eight note, sixteenth note).

Inter-Onset Interval (IOI): Is the distance between two consecutive note onsets.

Minimum Inter-Onset Interval (MIOI): In a whole piece or an excerpt of a song, it is the minimum distance between all pairs of consecutive onsets.

Foot-tapping: Is the act of tapping in the ground with the feet to mark the rhythm of a song, some listeners prefer to move the head instead of tap with the feet to mark the rhythm, it is commonly used with rock or electronic music, where the rhythm is more strong and becomes more noticeable.

Beats per Minute (BPM): It is usually the number of quarter notes per minute in a song. The value will be bigger in fast songs and lower in slow songs. Every beat matches with the movement of the foot or movement of the head when foot-tapping following the rhythm of a song.

So with these previous definitions we can say that the BPM detection consists is determining the number of beats per minute in a song, just as it would be to count the

number of times that the listener taps with the feet in the ground during a minute when listening to a song, but obviously this is not the way to calculate it. The process to detect the BPM, consists in firstly detect the position of beats in time and try to find the BPM value that best explain these beat occurrences in time. There are some techniques to calculate it, by doing a bank of energy filters and calculating the energy convolution of each filter or determining the inter-onset intervals between notes and extracting the BPM out of this:

- Bank of energy filters: The process consists in performing a bank of filters over the signal where the BPM is going to be extracted. The filters can be either calculated in time domain [Sch98] or spectral domain [May01] depending on other needs of the analysis. Over these filters, the energy along the time pattern of each one is calculated and also an overall value ponderating each energy-filter by a value usually giving more importance to low frequencies where the more rhythmic instruments are usually located (bass, drums, accompaniment chords). From these energy patterns we have to derive the BPM value, and it is easy to do it performing a convolution between the rhythm pattern obtained and a set of patterns representing all the possible tempos over a range. The pattern in the set that gives the higher value after performing the convolution will be selected as the BPM candidate.

- Inter-onset intervals: The process consists in, first of all, perform an onset detection over the piece that is being analyzed. Once the onsets are extracted we calculate all possible Inter-Onset-Intervals and the minimum value (MIOI). With these, it is possible to derive the tempo value (BPM). The MIOI will correspond to the minimum time-division in the score (eight-note, sixteenth-note…) and will help to quantize the onsets detected to this minimum time-division and try to find the BPM value that best explains the onsets pattern that has been extracted [Dix97] [Des92] [Dix99].

2.2.3 Beat representation

The beat information of a song can be represented in many ways. The most common representation consists in giving a number for the whole song, this number represents the BPM value, the number of beats per minute. In situations where the tempo has variations and does not remain constant for the whole song, we need information of this BPM value along the time, a simple function can represent the tempo evolution, but more sophisticated ways to represent the tempo and other rhythm information have been developed. Two graphical representation of rhythm, including The Beat Spectrum developed by Jonathan Foote and the BPM Spectrogram by Oscar Mayor are exposed here.

2.2.3.1 The Beat Spectrum

The Beat spectrum [FU01] is a new method for automatically characterizing the rhythm and tempo of music in audio. It is a measure of acoustic self-similarity as a function of time lag. Highly structured or repetitive music will have strong peaks at the repetition times. In figure 4 we can see a Beat Spectrum of a song.

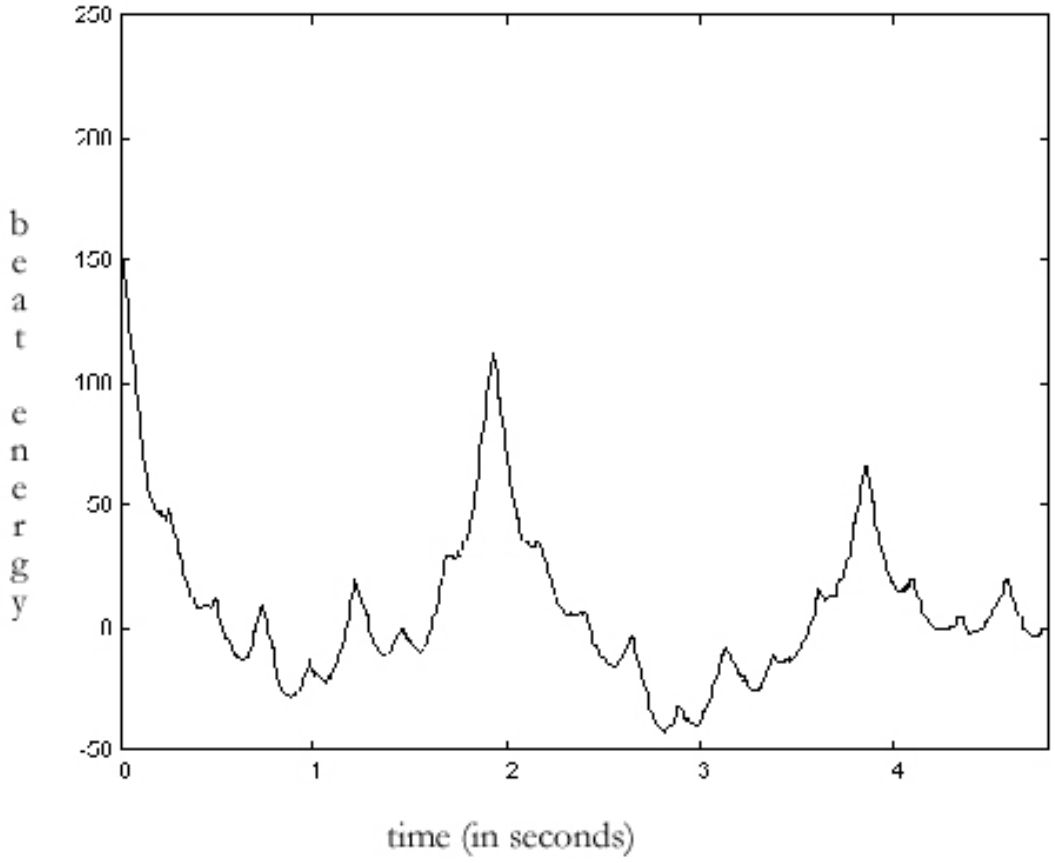

Figure 4: Beat Spectrum graphical representation.

The highest peaks in the spectrum represent high beats along the time, with the spectrum of the figure 4 we can extract that in this piece there is a strong beat every two seconds.

2.2.3.2 The BPM Spectrogram

The BPM spectrogram [May01] represents the beat information along the time in three dimensions, the x-axis will represent time, the y-axis will represent the BPM value and the z-axis the importance of the BPM value in level of energy. To show the

representation of the BPM spectrogram projected into two dimensions, a scale of colours ranging from white to black to represent the z-axis is used. In figure 5, a BPM Spectrogram is shown for a piece with a BPM value of 99 BPM. The darker horizontal line represents the correct BPM (99) and there are other lines representing double and 2/3 tempo because the rhythm has some beats at the middle and at ternary positions of the beat. In this case, the tempo is constant along the time, in other examples with tempo variations, the straight darker line will have deviations to represent the correct BPM value

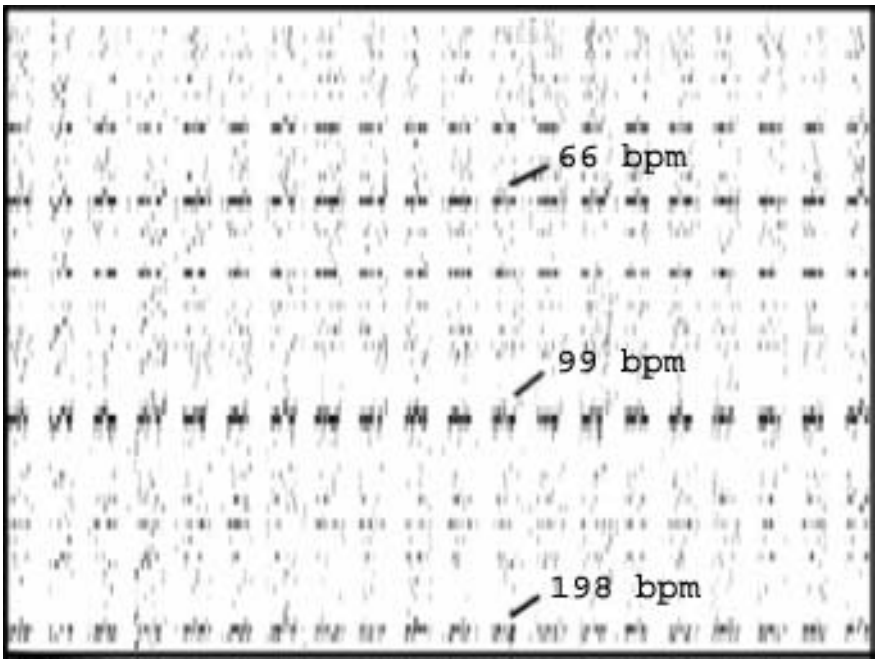

Figure 5: The BPM Spectrogram for a constant tempo song

2.3 Synthesis of musical sounds

2.3.1 Introduction

In the history of sound synthesis, there have been two different general methods for synthesis of musical instrument sounds. One approach is to look at the spectrum of a real instrument and try to recreate it. This includes methods such as *additive* or *subtractive synthesis* and *frequency modulation* (*FM*). With these methods we produce sounds with similar structure, but the parameters involved have no relation with the physical parameters of an instrument. Another approach is to use a recorded sample of the instrument sound, such as in *wavetable synthesis* and *samplers*. Another approach called *model based synthesis* consists in a method where we need first an analysis of the sound of the instrument that we want to synthesize and then these analyzed sound is transformed and resynthesized to create the output synthesis sound, these methods include the popular *Sinusoidal Plus Residual* modelling (*SMS*). In all these cases, you're creating sounds without any consideration of how the real instrument actually creates those sounds. When we don't want to create the sound directly but a process that creates and controls the sound we use the *physical modelling synthesis*. This process takes the sound synthesis up to a higher level, the idea is to define physically the process which defines the actual instrument, so when you play the synthetic instrument, you are specifying the physical parameters needed to make the sound. Another new method that appeared in the last years, called *granular synthesis* consists in creating complex timbres using banks of simpler elemental sounds, it's like taking additive synthesis to a higher level.

2.3.2 Additive Synthesis

Additive synthesis [Reid00] is based on the idea that complex tones can be created by the summation, or addition, of simpler ones. It is theoretically possible to break up any complex sound into a number of simpler ones, usually in the form of sine waves. In additive synthesis, we use this theory in reverse. This synthesis technique is also called Fourier Synthesis, the classical form of additive synthesis usually used in some synthesizers, may be called "harmonic synthesis", here the sources added together are simple sine waves and are in the simple harmonic ratios of the harmonic series. Over hundreds of harmonics may be used to create a complex sound and each of this harmonic is known as a partial.

Additive synthesis is a useful tool for periodic and harmonic sounds but noisy or chaotic ones are hard to generate. Creating the steady state of an instrument note is simple with additive synthesis (a few sine waves), but the attack of the note, usually noisy in many woodwind and brass instruments, is nearly impossible and we have to resort to other methods.

Additive synthesis can be implemented in time domain adding sinusoids to create the synthesis sound or in the frequency domain, filling an spectrum with harmonic peaks (at

the frequency desired) and synthesizing the spectrum just doing the inverse Fourier transform of it.

2.3.3 Subtractive synthesis

Subtractive synthesis [Rec98] is often referred to as analogue synthesis because most analogue synthesizers use this method of generating sounds. Subtractive syntheses is a very simple process consisting in creating a complex sound with an oscillator and filter it to modify the brightness of the sound and create a more suitable sound which is amplified to control the loudness of the synthesis over the time. In essence, you start with a sound and subtract out the undesired part and control its loudness over the time.

2.3.4 FM Synthesis

Frequency Modulation (or FM) synthesis is a simple and powerful method for creating and controlling complex spectra [Chow73]. It uses one periodic signal (the modulator) to modulate the frequency of another signal (the carrier). If the frequency of the modulator is in the sub-audio range (1-20Hz) it results in siren-like changes in the pitch of the carrier, but when we raise the frequency of the modulator to the audio range (>30Hz) we obtain a new timbre composed of frequencies called sidebands. The idea is to create a new sound with some desired sidebands, so we have to adjust the Carrier and Modulator frequencies (called C:M ratio) in order to add or not which sideband to the output synthetic sound. Each FM voice requires a minimum of two signal generators and sophisticated FM systems may use 4 or 6 signal generators per voice with adjustable envelopes which allow adjustment of the attack and decay rates of the signal. FM synthesis is very useful for creating expressive new synthetic sounds but when we need to recreate a sound of a real instrument it is recommended to use other more accurate digital techniques.

2.3.5 Wavetable and Sampler Synthesis

Digital Sampling systems store high quality samples of a recorded musical instrument, voice or whatever and then replay these sounds on demand. In order to reduce the amount of memory required to store all the possible sounds that we want to replay, some techniques, such as sample looping, interpolation, digital filtering or pitch shifting, are applied to transform these sample sounds.

2.3.6 Model Based Analysis/Synthesis

Model based Analysis/synthesis refers to all the synthesis methods that require a previous modelling of the sound that is going to be synthesized. We will analyze some synthesis methods that follow this technique, analyzing samples of the instruments to re-synthesize it: In frequency domain: Sinusoidal plus Residual synthesis (SMS) and Spectral Peak Processing and in Time Domain: OLA/PSOLA/TD-PSOLA synthesis.

2.3.6.1 Sinusoidal + Residual (SMS)

SMS is a spectral analysis/synthesis technique based in the decomposition of the sound in a harmonic part (sinusoidal) and a residual part [SS89] [Serra97]. This technique allows by means of an analysis process to extract a set of parameters and characteristics of the original sound, in order to be transformed to create the synthetic sound by re-synthesizing the original sound with the new desired parameters. These parameters pretend to be related with high level attributes of the sound to allow musically meaningful transformations [SB98].

In figure 6 the SMS analysis process is shown.

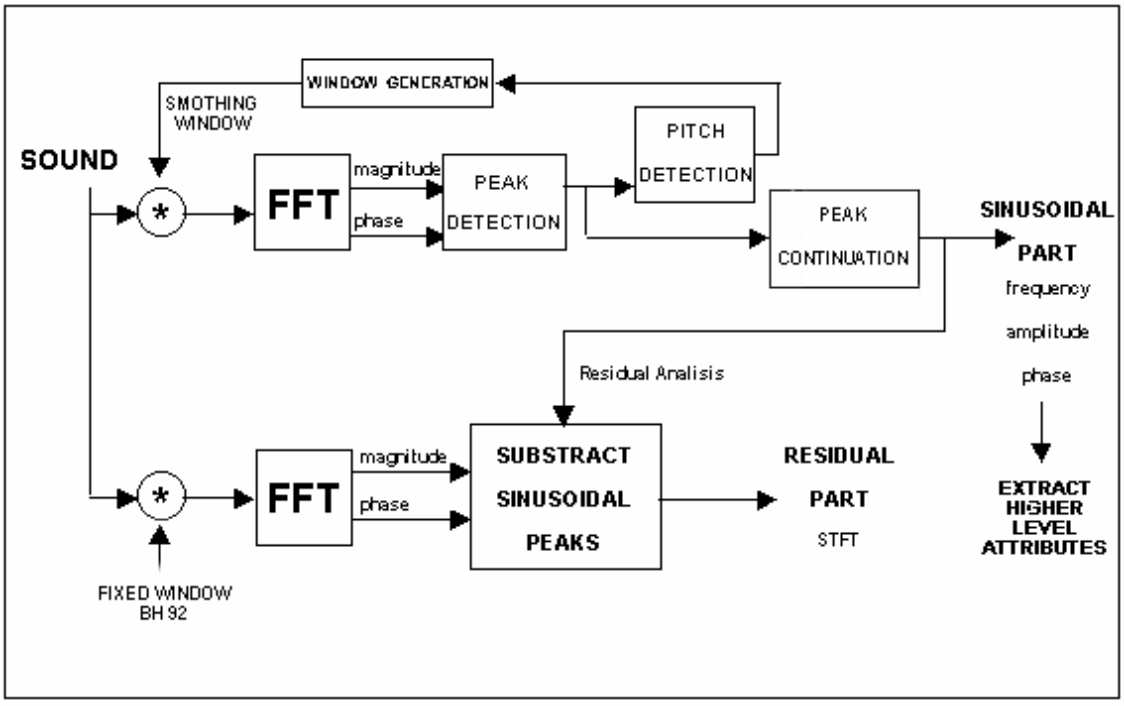

Figure 6: SMS analysis process

The basic idea is to analyze a sound, calculating the spectrum with the short time Fourier transform, detecting the harmonic peaks (in case of harmonic sounds), extract the pitch of the signal and with this information, subtract the harmonic peaks from the original sound to have separately the harmonic part of the sound (sinusoidal part) and the residual part of the sound (peaks that are not harmonic). It is also possible to extract higher level attributes with a more musically related meaning. With this representation it is possible to apply musically meaningful transformations to only the sinusoidal part and re-synthesize this part and add the residual part as it is. The synthesis consists in performing the inverse Fourier transform of the sinusoidal part adding the residual part in time domain or frequency domain depending on the application and the previous way of calculating the residual part.

2.3.6.2 Spectral Peak Processing

Spectral Peak Processing is an analysis by synthesis technique based on considering the spectrum as a set of regions, each of which belongs to one spectral peak and its surroundings [BL01] [Laro03]. The goal of such technique is to preserve the convolution of the analysis window after transposition and equalization transformations. The local behaviour of the peak region should be preserved both in amplitude and phase after transformations. To do so, the delta amplitude relative to the peak's amplitude and the delta phase relative to the peak's phase are kept unchanged after spectral transformations. The region boundary is set to be at the lowest local minimum spectral amplitude between two consecutive peaks or if there are no local minimums, at the middle frequency between two consecutive peaks. We can apply basically two kinds of transformations over the spectrum: equalization and transposition.

Equalization means a timbre change. From the SPP analysis we get a spectrum with the harmonic peaks and the regions. Then we want to change the spectrum to follow a desired envelope that will change the timbre of the sound. Each region is shifted up or down the amount needed so that the peak match with the target envelope and the phase is unchanged.

Transposition means to change the pitch of a spectral frame by multiplying the harmonic's frequencies by a constant value. In SPP, this operation can be done by shifting SPP regions in frequency [LD99] [Laro03]. The amount of frequency shifting calculated for each harmonic peak is applied as a constant to its whole region, so the linear frequency displacement for all the bins in a region will be the same. Therefore, the local amplitude spectrum of each region will be kept as it is, thus preserving the window convolution with each harmonic peak. When transposing to a higher pitch, the SPP regions are separated in the resulting spectrum with holes that are filled up with spectral amplitudes of constant amplitude -200dB and when transposing to a lower pitch, the SPP regions will overlap in the resulting spectrum. This overlapping is achieved by adding the complex values at each spectral bin. The phases of each region are changed to maintain a linear evolution.

2.3.6.3 OLA / SOLA / PSOLA

These methods of synthesis (not analysis) are usually known as time segment processing [DZP02] [Bon00]. They divide the sound in segments and reorder them in time. This processing mainly influences the pitch and the time duration of the audio signal.

OLA (Overlap and Add) technique is the simplest one and consists in the extraction of windowed segments of real audio to reorder them in time to time-expand or time-compress the sound. This reordering, without taking into consideration the phase relations between the segments, yields to pitch discontinuities and distortions that produce a bad quality output sound. The phase discontinuities that appear in the OLA technique are solved in SOLA (Synchronous Overlap and Add) minimizing distortions adding a fade-in and a fade-out between the segments and the overlap intervals are decided to achieve the maximum similarity between boundaries always preserving the time scale factor along the time.

PSOLA (Pitch Synchronous Overlap and Add) not only allows modifying the duration but also the pitch of the sound. It requires an analysis of the input sound where the pitch is calculated. This algorithm is commonly used in voice processing or monophonic musical instruments because it requires that the input sound is characterized by a pitch. In a first phase, the input sound is analyzed and segmented and the knowledge of the pitch is used to correctly synchronize the overlapping segments to avoid pitch discontinuities. In this method a robust pitch detection algorithm is very important to obtain good quality transformations.

2.3.7 Physical Modelling

Physical modelling [Cook95] comes from the problematic way to emulate the most expressive musical instruments that react to how they are played, to drive the synthetic instrument sharp or flat, just like the real thing and be able to change the timbre of the instrument and not be limited to use always the same sample over and over again, only being able to adjust the envelopes and adding vibrato. You are creating and controlling a process that produces a sound, and to control this process you have a set of adjustable parameters easy to understand because they have a real counterpart.

Physical Modelling is quite flexible and a single algorithm can achieve a wide variety of sounds, without requiring more memory. New sounds can be created just adapting the input parameters. The main drawback of physical modelling is that it requires quite a bit of processing power, so models that mimic accurately a real instrument have to take into consideration a lot of physical equations and need a powerful hardware to work in real-time. Another drawback is that you need a new model every time you want to emulate a new instrument to produce a realistic sound [Smith87] [Smith92].

2.3.8 Granular Synthesis

Granular synthesis could be seen as a form of additive synthesis but its approach and sonic results are quite different from additive synthesis [Truax93]. Until very recently the techniques and means associated with granular synthesis were not available to many people. This was due to the nature of granular synthesis, in that it can contain literally thousands of parameterized events just to specify one second of sound. Processing such large amounts of data required the work of mainframe computers equipped with a digital to analog converter (DAC). Granular synthesis makes an organization of music into "corpuscles of sound". Any sound could be synthesized with the correct combination of numerous simple sonic grains. The grain is a flexible representation for musical sound because it combines time-domain information (starting time, duration, envelope shape, waveform shape) with frequency domain information (the frequency of the waveform within the grain). In the years of the magnetic tape, granular synthesis idea was implemented by the laborious process of cutting and splicing hundreds of segments of tape for each second of music, an intimidating and time-consuming activity. It was not until digital synthesis that advanced composition with grains became feasible.

Granular synthesis is unique because it collapses the time and frequency domains within the concept of the grain [Clarke96]. This collapse in domain allows the limits of sonic perception to be exploited. The textures that can be created using granular synthesis are very exciting as they allow for a sound landscape that is totally controllable by the composer. The composer controls each timbre, each duration and each pitch within the piece and also how these parts interact with each other. This advantage of complete controllability of all aspects of the texture is carried out at the macro, as well as the micro level. Unfortunately such a large amount of control makes it difficult to keep control of all the levels. This is overcome by using statistical forms of control.

Granular synthesis is a very powerful means for the representation of musical signals and provides an opportunity for a composer to expand his or her sonic "palette". When granular synthesis techniques are used in conjunction with sampled waveforms, the possibilities for new sounds are infinite.

2.4 Musical performance analysis (Naturalness and Expressiveness)

2.4.1 Introduction

"For I consider that music is, by its very nature, essentially powerless to express anything at all, whether a feeling, an attitude of mind, a psychological mood, a phenomenon of nature, etc... Expression has never been an inherent property of music. That is by no means the purpose of its existence. If, as is nearly always the case, music appears to express something, this is only an illusion and not reality. It is simply an additional attribute which, by tacit and inveterate agreement, we have lent it, thrust upon it, and as a label of convention -- in short, an aspect unconsciously or by force of habit, we have come to confuse with its essential being."

-- Stravinsky, Igor. **An Autobiography.** orig. pub. 1936. London: Calder and Boyars, 1975, 53-54.

By musical expression, we understand the variations in tempo, timing, dynamics, articulation, etc. that performers apply when playing and interpreting a piece. By naturalness in our experience with synthesized sounds, spoken or musical, it is not difficult to find many synthesis systems to be inadequate or unconvincing. We can hear that something is wrong and we tend to call these artefacts, low quality or robotic interpretation under the common name of “unnatural” or “lacking in naturalness”. If you ask composers about how they feel about natural sounds, they will say that for them to sound natural is not the main objective. Usually Naturalness takes second place versus expressiveness, usually for a composer is more important to play an instrument with a high degree of expressivity but low synthesis quality than a very good quality but inexpressive sound. There are some exceptions in singing or speech synthesis, where intelligibility which is tightly and more related with naturalness, is often more important than expressivity. Realism which is another concept related with expression and naturalness, means to get a synthesis which mimics the reality.

Many researches have been done in several research centres all over the world. The work done in these centres is a good starting point for my research in the field of musical expression. In the following part, I summarize the work being done in some of these centres including ÖFAI in Austria, KTH in Sweden and IIIA and MTG in Spain.

2.4.2 Musical Expression Research Centres

- Austrian Research Institute for Artificial Intelligence (ÖFAI).
- Department of Speech, Music and Hearing (KTH) Rules for music performance.
- Artificial Research Institute (IIIA) & Music Technology Group (MTG) Case-based reasoning system for generating expressiveness musical interpretations.

2.4.2.1 Austrian Research Institute for Artificial Intelligence (ÖFAI)

By musical expression, it is understood the variations in tempo, timing, dynamics, articulation, etc. that performers apply when playing and "interpreting" a piece. The goal is to study real expressive performances with machine learning methods, in order to discover some fundamental patterns or principles that characterize "sensible" musical performances, and to elucidate the relation between structural aspects of the music and typical or musically "sensible" performance patterns. The ultimate result would be a formal model that explains or predicts those aspects of expressive variation that seem to be common to most typical performances and can thus be regarded as fundamental principles.

To achieve this, it is necessary to:

- obtain high-quality performances by human musicians (e.g., pianists)
- Extract the "expressive" aspects from these and transform them into data that is amenable to computer analysis (e.g., tempo and dynamics curves)
- Analyze the structure (meter, grouping, harmony, etc.) of the pieces and represent the scores and their structure in a formal representation language
- Develop machine learning algorithms that search for systematic connections between structural aspects of the music and typical expression patterns, and formulate their findings as symbolic rules
- Perform systematic experiments with different representations, sets of performances, musical styles, etc.
- Analyze the learning results with a view to both qualitative (are the discovered rules musically sensible? interesting? related to theories by other expression researchers?) and quantitative terms (how much of the variance can be explained? where are the limits?).

Some research done in ÖFAI include:

- Data Acquisition and Extraction of "Expression": Score extraction from expressive MIDI files (e.g., Cambouropoulos, AAAI'2000), score-to-performance matching and beat and tempo tracking in MIDI files (Dixon & Cambouropoulos, ECAI'2000) and audio data.
- Automated Structural Music Analysis: Segmentation (Cambouropoulos, AISB'99), Clustering and motivic analysis (Cambouropoulos & Widmer, J.New.Mus.Res. 2001)

- Musical category formation (Cambouropoulos, Music Perception 2001)

- Studying the nature of basic percepts related to expression: Experimental studies on the perception of tempo (changes) in listeners (Dixon & Goebl, 2002) and experimental studies on the perception of timing asynchronies (Goebl & Parncutt, 2002)

- Performance Visualisation: A software tool for animated visualisation of high-level patterns (Dixon, Goebl, & Widmer, 2002) and based on a visualisation idea by Jörg Langner (Langner & Goebl, 2002, 2003)

- Extensions to real-time tracking, smoothing, and animation (Dixon, Goebl, & Widmer, ICMAI'02): High-level visualization of performance patterns used by pianists (Pampalk , Widmer & Chan, 2003)

- Systematic Performance Analysis: Performance averaging (Goebl, SMPC'99), melody lead (Goebl, JASA 2001), articulation (Bresin & Widmer, 2000), relations between segmentation structure and low-level timing (Cambouropoulos, ICMC'2001) and systematic investigation of different tempi (Goebl & Dixon, 2001)

- Inductive Model Building (Machine Learning): Fitting existing expression models onto real performance data (Kroiss, 2000), looking for structure in extensive performance data (Widmer, ICMC'2000), inducing partial models of note-level expression principles (Widmer, JNMR 2002, Artif.Intell. 2003), inducing multi-level models of phrase-level and note-level performance (Widmer & Tobudic, J.New.Mus.Res, 2003)

- Characterization and Automatic Classification of Great Artists: Learning to recognize performers from characteristics of their style (Stamatatos & Widmer, ECAI'2002), discovering performance patterns characteristic of famous performers (Widmer, ALT'2002)

- A recent direction of research not directly related to expression (but with high practical potential): Music Information Retrieval (MIR): Organization and Visualization of Digital Music Archives (Pampalk et al., ACM Multimedia 2002; Pampalk et al., ISMIR 2003), rhythm detection and style classification (Dixon et al., ISMIR 2003)

2.4.2.2 Department of Speech, Music and Hearing (KTH). Rules for music performance

Combinations of performance rules and of their parameters can be used for synthesizing interpretations that differ in emotional quality (fear, anger, happiness, sadness, tenderness, solemnity) [ALS97] [BF00]. These rules and their parameters have to be selected so as to match previous findings about emotional aspects of music performance. The rules produce variations of the performance timing and dynamics with respect to a nominal performance. Some listening tests have been done were listeners are asked to classify some performances with respect to emotions, the results show that the listeners, with very few exceptions, recognize the intended emotions correctly, so the application of these music performance rules have a wide variety of meaning.

The idea is to apply the composer's mental representation to the score that is going to be performed in order to get a listener perception with the most resemblance to what the composer wants to express. The diagram in figure 7 illustrates this process:

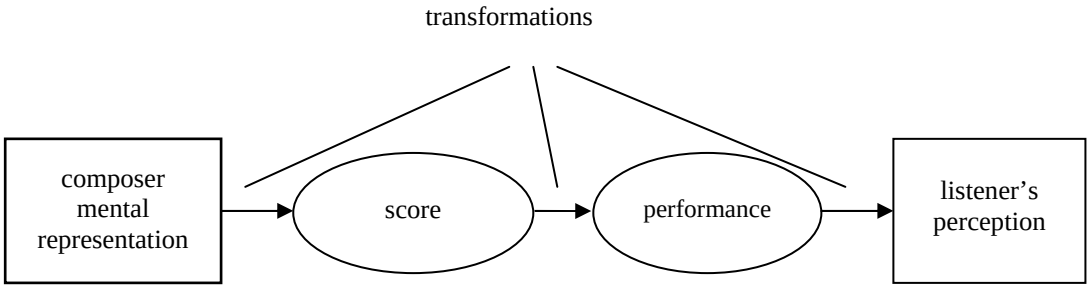

Figure7: Expression process from the composer to the listener

The developing and decision of where to apply these rules is driven by an analysis-by-synthesis process. The process starts with an idea which is formulated as a tentative rule in the computer. Then this rule is applied to a music example so that the result can be evaluated by listening. This offers an immediate feedback, often suggesting further modifications. The process is then repeated until a satisfactory performance is obtained (loop). Thus in a sense the system acts as a student acquiring some basic knowledge of music interpretation from an expert teacher.

One requirement of this method is that everything must be quantified. A typical observation has been that the exact quantity of each parameter is crucial for a good performance. In determining the dependence of a rule on a certain parameter, such as note duration, it is generally helpful to find two extremes and then to interpolate linearly between them. If this does not yield an appropriate result a different function, e. g., a power function can be tried. In this way we can successively improve the rule step by step. Figure 8 shows this analysis-by-synthesis process:

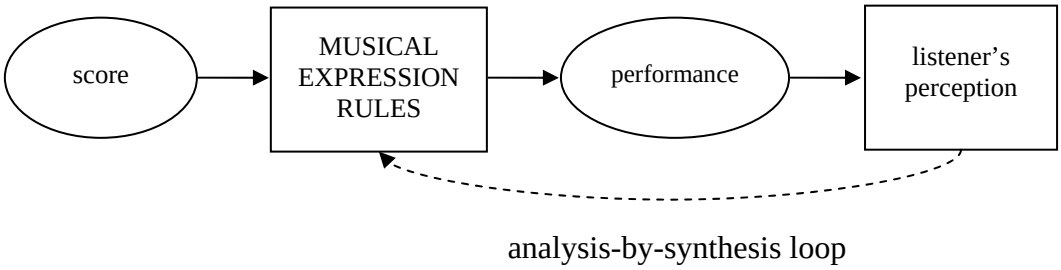

Figure 8: Analysis-by-synthesis loop for the application and development of the expressive rules

The rules created by this analysis-by-synthesis process loop can be grouped in several categories [Fri91]:

Differentiation of Duration Categories • Duration contrast • Double duration • Accents	**Differentiation of Pitch Categories** • High sharp • High loud • Melodic charge • Melodic intonation
Microlevel Grouping • Punctuation • Leap articulation (level envelope) • Leap articulation (micropause) • Leap tone duration • Faster uphill • Amplitude smoothing • Inégales • Repetition articulation (level envelope) • Repetition articulation (duration)	**Macrolevel Grouping** • Phrase arch • Phrase final note • Harmonic charge • Chromatic charge • Final ritard
Ensemble • Melodic synchronization • Bar synchronization • Mixed intonation • Harmonic intonation	

Differentiation of Duration Categories

These rules are related with changes in the duration of notes.

Duration contrast

The contrast between long and short note values, such as half note and eighth note, is sometimes enhanced by musicians. They play the short notes shorter and the long notes longer than nominally written in the score. This rule makes short notes shorter and softer.

Double duration

Tone groups in two note values having the ratio of 2:1 alternate (such as an eight note followed by a quarter note, followed by an eight note and so on) are not played according to their nominal durations. Instead, the short tone is lengthened at the expense of the long note.

Accents

Accents, which consist in an increase of the sound-level of a note are given to notes in the following contexts: a short note between long notes, the first of several short notes, and the first long note after an accented note.

Differentiation of Pitch Categories

These rules are related with changes in the pitch of notes.

High sharp

This rule plays high tones sharp and low tones flat, just like some musicians tend to do when they play. The pitch variation appear quite subtle to most listeners, in upgoing intervals the stretching can be increased up to 6 cent per octave.

High loud

This rule increases the loudness in proportion to the pitch height. The sound level of voice, brass and some woodwinds increases with pitch when a musician is instructed to play at a constant dynamic level but for keyboards, strings and plucked instruments the sound level is not affected by the pitch in the same manner.

Melodic charge

This rule accounts for the "remarkableness" of the tones in relation to the phrase or song harmony. For example, the root of the chord is trivial while the augmented fourth above the root is very special. We need an analysis of the harmony to apply this rule and it will not be applicable for atonal music. Sound level, duration and vibrato extent are increased in proportion to the melodic charge value.

Melodic intonation

The pitch deviation from equal temperament (ET) is made dependent on the note's relation to the root of the current chord. In some cases the pitch tends to be similar as in Pythagorean tuning. It is commonly applied to single voices, not to polyphonic music.

Microlevel Grouping

These rules refer to more general changes involving both, pitch and duration of notes based in a microlevel analysis of the melody or harmony.

Punctuation

The melody can be divided into small musical gestures normally consisting of a few notes. This rule tries to identify and perform these gestures. It consists of two parts: the gesture analysis and the application of these in the performance. Micropauses and a dip in the level envelope are inserted at the boundaries of these gestures.

Leap tone duration

The first note in an ascending melodic leap is shortened and the second note lengthened in duration if the preceding and succeeding intervals are by step (less than a minor third). In a descending leap the first note is lengthened and the second shortened. The amount in ms is only dependent on the interval size of the leap (unaffected by the duration).

Faster uphill

The durations in an ascending melodic line are shortened. Usually the performer tends to go fast to the target note to rest in it, and he does in an ascending line.

Amplitude smoothing

This rule smoothes out the level differences between subsequent notes by changing the level envelope linearly from onset to onset. This rule is intended for instruments with a continuous sound where the sound level can be changed, e. g. woodwinds, brass, voice. It is essential for a realistic performance on these instruments.

Inégales

This rule lengthens the stressed notes in sequences of notes having the same note value. All eighth notes appearing on a strong beat will be lengthened and all eighth notes appearing on a weak beat will be shortened. This applies also to all notes or rests starting or ending at a weak position. The rule reflects a convention used in Baroque music as well as in jazz commonly known as swing-feel. You can vary the duration relations between the stressed and unstressed notes.

Repetition articulation (level envelope and duration)

This rule applies changes to notes in a repetition. This rule inserts a dip in the level envelope between these notes and inserts a micro pause between notes of the same pitch.

Macrolevel Grouping

These rules refer to more general changes involving both, pitch and duration of notes based in a macrolevel analysis of the melody or harmony. This means that we need a previous analysis of the song to apply musical changes depending on this analysis.

Phrase arch

Music has a hierarchical structure, so that small units, such as melodical gestures, join to form subphrases, which join to form phrases, etc. When musicians play, they mark the endings of these tone groups. This rule marks the phrase and subphrase endings by creating accelerandos and decelerandos within phrases and subphrases according to a parabolic function. Thus it increases the tempo in the beginnings and decreases it towards the endings. The loudness is changed similarly creating crescendos and diminuendos. The phrases and subphrases have to be calculated by analyzing the score of the song being transformed

Phrase final note

This rule marks phrases on two hierarchical levels: phrase and subphrase. The last note in a phrase and the last note in the piece are lengthened. After the last note of a phrase or subphrase a micropause is inserted. In this rule we need also a phrase analysis of the piece. This rule is often applied to speech, where the last syllable of a sentence is lengthened.

Harmonic charge

This rule marks the distance (related to the distance on the circle of fifths) of the current chord to the root of the current key. Sound level, duration and vibrato frequency are increased in proportion to the harmonic charge value. The increases and decreases of these parameters are gradual with linear interpolation between chord changes. This rule requires a previous analysis of the piece to obtain the harmonic charge of chords in the song and the changes can not be applied in atonal music.

Chromatic charge

This rule increases the sound level and duration in areas where the intervals between the notes are small. This rule is applicable to atonal music or music where the harmonic analysis is hard to be obtained.

Final ritard

The tempo at the end of the piece is usually decreased according to a square-root function of nominal time (or score position).

Ensemble

These rules are related to complex harmonizations or polyrythmic situations. The harmony and the rhythm are taken into account.

Melodic synchronization

A new voice is constructed consisting of all new tone onsets from all voices. If several tones appear on the same onset, the one with the highest melodic charge value will be chosen. All duration rules are then applied to this new voice and the resulting timetable is transferred back to the original voices. This means that all simultaneous notes in all voices will be perfectly synchronized.

Bar synchronization

This rule synchronizes the onset times for the first note in each bar. The length of the voice with the most number of notes will be used as the bar length. The other voices will be adjusted proportionally to the same length. It is intended to be used in complicated polyrhythmic situations.

Mixed intonation

This rule is a combination of MELODIC and HARMONIC INTONATION, taking into consideration both the melodic strive to intonate minor seconds smaller than equal temperament and at the same time allow for beat-free chords. The initial pitch deviation will be set according to the melodic intonation. Slowly, the pitch deviation will change to a beat-free interval relative to the root of the chord.

Harmonic intonation

Every note is tuned so that the beats are minimized relative to the root of the current chord. This rule is not intended to be a stand-alone rule. It is used mainly for demonstrations of the effect of tuning each chord so that beats are minimized. A melody will normally sound out of tune. This is the target tuning for long chords in MIXED INTONATION.

2.4.2.3 Artificial Research Institute (IIIA) & Music Technology Group (MTG). A case-based reasoning system for generating expressiveness musical interpretations

Introduction

SaxEx is a system developed at the Artificial Research Institute (IIIA) in collaboration with the Music Technology Group (MTG) that is able to generate expressiveness musical interpretations, from non-expressive ones based in CBR (Case Based Reasoning) techniques [ALS97]. It Comes from the need of not only use the musical rules to generate music but also to use the human process as observation, imitation and experimentation. So SaxEx uses a database with a lot of real expressive interpretations, in order to create new ones with case based reasoning procedures. SaxEx has been developed and implemented using a new language, also developed in the IIIA, called Noos, an object oriented language designed to represent the knowledge in problem solving and learning. The aim is to get nice expressive results musically speaking, and to achieve this, SaxEx makes use of two general theories about perception and musical understanding and jazz theory knowledge, because the input of the system are melodies and chords taken from jazz standards.

In figure 9 a diagram of the complete system is shown, which shows the CBR and analysis/synthesis modules, the inputs and the output of the system.

The input of the system consists in:

- A musical phrase described by its *score*, a MIDI file with melodic and harmonic information.

- A recorded sound (a wav file with the musical phrase interpreted by a human playing with no expressivity).

- Values for the affective parameters (tender - aggressive, sad - happy, calm - restless)

The output of the system is:

- A set of new sound files obtained by doing transformations to the original, adding expressivity based on the values that the user has specified in the affective parameters.

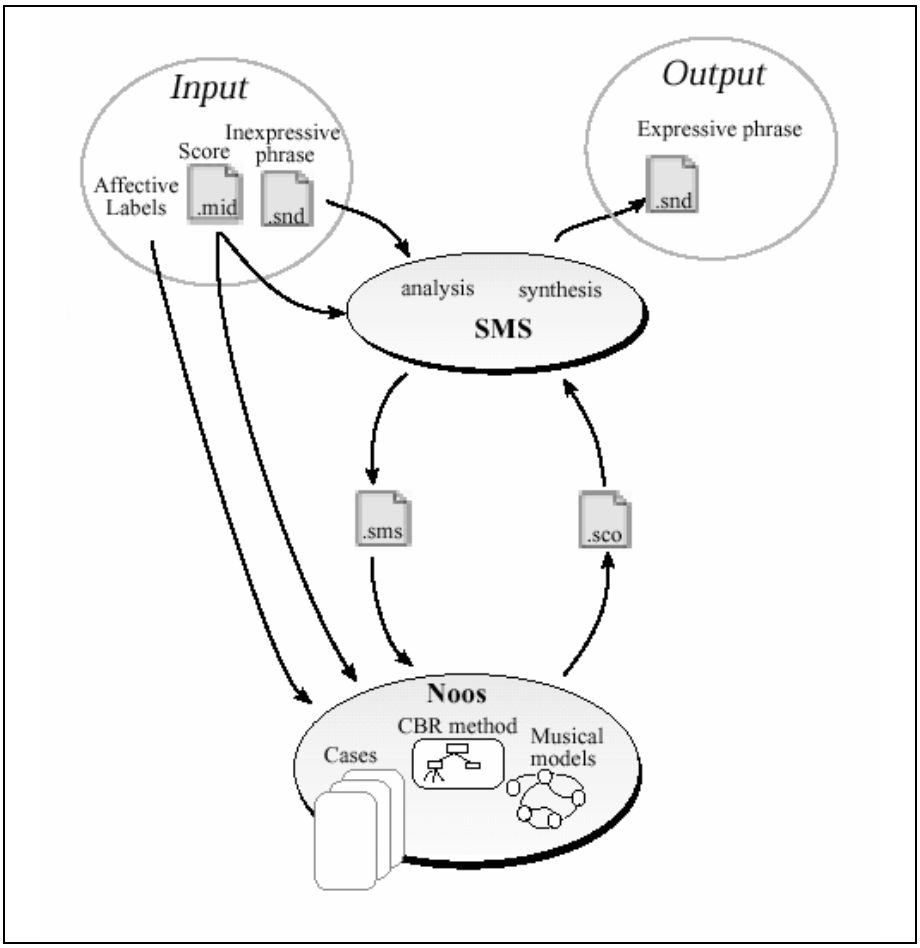

Figure 9: Block diagram of the SaxEx system

The SaxEx process can be divided in three phases:

- Sound Analysis (SMS)
- Case-Based Reasoning (CBR)
- Sound Synthesis (SMS).

The sound analysis and synthesis techniques use the Spectral Modelling approach (SMS) very useful to extract high level attributes from the real sounds that allow doing transformations in a meaningful way.

The CBR phase is the most important and integrates some aspects of Artificial Intelligence (Case Based reasoning Techniques) into the system.

Reasoning Process (CBR)

As we have said before, this is the most important part of the system, here is where the system finds the solution to the problem (musical phrases where we want to add expressivity).

The internal representation in SaxEx is divided in two models:

- The Domain Model, represents the musical knowledge

- The Problem Solving Model, which gives a sequence of expressive transformations of a musical phrase using the CBR module.

These two domains are tightly related because the domain model integrates all the musical knowledge that needs the problem solver to find a good solution. This model uses a general theory based in the musical perception called the *Narmour's implication/realization (IR) model*, another theory based in the musical understanding called *Lerdahl and jackendoff's generative theory of tonal music GTTM* and specific knowledge about jazz, because the system is intended to add expressivity to jazz standards.

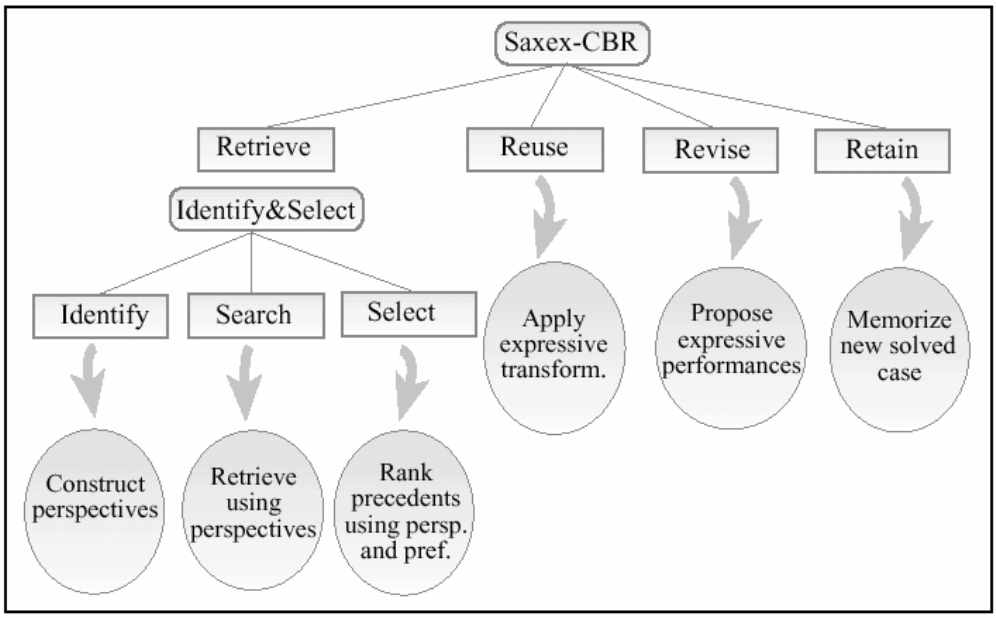

Figure 10: The CBR method in SaxEx

The problem solver includes the CBR, shown in figure 10, and the reasoning process which can be divided into four different stages:

- Retrieve: for each note of the melody, we choose from the case memory that contains the expressive interpretations of the melody, the set of similar notes to the problem. The values of the affective parameters (tender-aggressive, sad-happy, calm-restless) specified by the user will be decisive in the decision of the notes chosen. This retrieval can be subdivided into 3 sub processes:

- Identification: using the musical knowledge integrated in SaxEx, the system determines the important musical characteristics of each note.

- Search: with Noos methods, the system searches in the case memory for similar situations to the previous analyzed note.

- Select: The goal of the select task is to rank the retrieved cases using Noos preference methods, based in similarity in duration of notes, harmonic stability or melodic directions.

- Reuse: The goal of the reuse task is to choose a set of expressive transformations to be applied in the current problem from the set of more similar cases, the aim is to adapt the transformations of the most similar case.

- Revise: A set of solutions are presented to the user, so he can choose which one is better for him.

- Retain: When the user chooses a solution as good, this solution to the problem is automatically added to the cases memory so it can be applied to future problems…

Chapter 3

Recent and current research

3.1 Introduction

In this chapter an overview of the articles published by the author is presented as well as other research done but not reflected in published papers or articles. For each paper / article, the abstract and the contribution to it is presented and analyzed with some detail.

For further investigation on the research reflected in the papers, in the appendix there is a printed version of all the articles, as they have been published in the proceedings or journals. These articles reflect quite well my contribution to the computer music field and are the basis of my future research for my on-going thesis.

The articles are ordered chronologically by year of research, although contributions to projects not reflected in papers are at the end of the chapter.

Research and Contributions reflected in papers

2001

3.2 'Statistical Significance in Song-Spotting in Audio' [CKMB01]

Cano, P. Kaltenbrunner, M. Mayor, O. Batlle, E. (2001)
Proceedings of International Symposium on Music Information Retrieval 2001, Bloomington, Indiana (USA)

ABSTRACT

We present some methods for improving the performance of a system capable of automatically identifying audio titles by listening to broadcast radio [CBMN02]. We outline how the techniques, placed in an identification system, allow us detect and isolate songs embedded in hours of unlabelled audio yielding over a 91% rate of recognition of the songs and no false alarms. The whole system is also able of working real-time in an off-the-shelf computer.

CONTRIBUTION

In this project I was involved in several developer and research tasks. As a developer, in the integration of the different modules (analysis, recognition, database, acquisition, matching), the creation of a graphical user interface to interact with the user and the development of the acquisition module (audio driver). My research contribution consisted in the analysis and implementation of different dynamic programming algorithms for string matching [Gusf97] [NDR97] used in biology for DNA and RNA string detection [PL88] [KA90]. These algorithms (Smith-Waterman, BLAST, FASTA) are used in the audio recognition system to match the fingerprint of the song being detected (represented by an ascii string) with the previously calculated fingerprints of the songs in the database.

2001

3.3 'An adaptative real-time beat tracking system for polyphonic pieces of audio using multiple hypotheses' [May01]

Mayor, O. (2001)

Proceedings of MOSART Workshop on Current Research Directions in Computer Music. Barcelona

ABSTRACT

In this paper a method for extracting beat information of a piece of music is presented, a real-time analysis is performed while the music is played or recorded from any source and the system gives the beats per minute value at each moment and beat occurrences in time. It also becomes adaptative in case of a sudden or smooth change in the tempo. It deals with multiple hypotheses, and gives the most suitable results at each time. One of the most important applications linked to this work is the automatic classification of pieces in different musical genres or finding song similarities in terms of musical rhythm.

CONTRIBUTION

I developed this project alone with no collaborations, so everything was developed by myself. The project was started with the idea in mind to be integrated in the audio recognition system [CBMN02] reinforcing the recognition. Adding a fingerprint of the rhythm of the song can offer the user the possibility to search for a similar song of the one being played just because both songs have a similar rhythm pattern. I developed a real-time beat tracking system and then the idea was continued by other researchers to integrate it inside the big project. The project consisted in several modules: the real-time sound acquisition, the spectral analysis which consisted in calculating the total and average energy of a bank of filters [Sch98], the development of an algorithm to detect the BPM (beats per minute) calculating multiple hypotheses [RGM94], the synthesis of a perceptual model of the rhythm, also developed inside this project, and the graphical user interface, including the visualization of the BPM spectrogram which is a visual representation of the rhythm [FU01] also developed inside this project.

2002

3.4 'Real-time spectral synthesis for wind instruments' [MBL02]

Mayor, O. Bonada, J. Loscos, A. (2002)
MTG internal (to be published)

ABSTRACT

A real-time spectral domain monophonic synthesizer is being developed. The idea is to achieve the sound quality given by samplers and improve the flexibility of control and model transitions between notes. The output sound is generated concatenating, transforming and synthesizing attacks and stationeries previously recorded by a real player that have been analyzed and stored in a database. A spectral technique called SPP is used in order to get better sound quality than other spectral techniques, like the sinusoidal or sinusoidal plus residual model, when transforming the sound. The aim is to get a sampler-like quality but with a high degree of flexibility to transform the sound and to model note transitions.

CONTRIBUTION

This project was begun by Joachim Haas [Haas01] and I contributed with several tasks including the addition of a new instrument to the synthesizer (trumpet) [DD98], improvements in the interpolation between spectral samples [SRD90], modelling of note transitions some changes in the synthesis core, reorganization of the database and speed improvements and rewriting of the code. The addition of the trumpet consisted in record, analyze and add new trumpet samples to the existing saxophone samples in the database and adjust some parameters of the synthesis to work with this new instrument. The interpolation improvement consisted in the decision of the peaks that were going to be modified when interpolating between two spectrums with different timbres.

2003

3.5 'Sample-based singing voice synthesizer using spectral models and source-filter decomposition' [BLMK03]

Bonada, J. Loscos, A. Mayor, O. Kenmochi, H. (2003)

Proceedings of 3rd International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications. Firenze, Italy

ABSTRACT

This paper is a review of the work contained in the insides of a sample-based virtual singing synthesizer. Starting with a narrative of the evolution of the techniques involved in it, the paper focuses mainly on the description of its current components and processes and its most relevant features: from the singer databases creation to the final synthesis concatenation step.

CONTRIBUTION

This project consists in the implementation of a singing voice software synthesizer. My contribution can be viewed as a developer in the integration of the synthesis/database/expression module with the graphical interface managing the calls between the modules and also the construction and tuning of the expressive template database. As a researcher my contribution consisted in the development of the expression module for the singing voice synthesizer explained in more detail in the next paper [MCBL03].

2003

3.6 'Musical expression in a singing voice synthesizer' [MCBL02]

Mayor, O. Celma, O. Bonada, J. Loscos, A. (2003)
MTG internal (to be published)

ABSTRACT

This paper talks about different techniques that have been developed in the insides of a spectral singing voice synthesizer [BLMK03] in order to apply / modify some expression features of the synthesis. These techniques are based on templates extracted from recordings and mathematical models and pursue the idea of mimicking the voice behaviour of a professional singer. Some of these templates / models have a local influence inside the song since they only modify the attack or release of a note, the type of a note to note transition, the vibrato or the pitch contour in certain phoneme to phoneme transitions. But more general templates / models which apply changes that have influence all over the phrase or even all over the whole song are considered as well. These models are more related with prosody and intonation and are specific to each style of singing like blues, pop, opera, folk, jazz. All these expression features are controlled by the user as a system input.

CONTRIBUTION

This paper is focused in the expressiveness module of the singing voice software synthesizer. My contribution inside this module consisted and consists, because it's the field where I'm involved now, in the development of local expression models for note transitions, attacks, releases, vibratos and phoneme articulations and global expression models for mood singing or styles of singing. That is to give a certain kind of expression to a piece depending on the mood (sad, happy, tender, slow, fast) or also depending on the style of singing (rock, country, jazz, blues, opera, classical) [BF00] [ALS97]. Templates based on professional singer's recordings and mathematical models are used to apply transformations to pitch, loudness, roughness, breathiness, timbre of the non-expressive synthesis to get the desired expressive performance [Poli03].

Other contributions to projects not reflected in papers

2000

3.7 'SaxEx'

Mayor, O. Arcos, J. (2000)
Universitat Autonoma de Barcelona – IIIA

ABSTRACT

Saxex is a case-based reasoning system for generating expressive performances of melodies based on examples of human performances. Case-based Reasoning (CBR) is a recent approach to problem solving and learning where new problems are solved using similar previously solved problems. The two basic mechanisms used by CBR are (i) the retrieval of solved problems (also called precedents or cases) using some similarity criteria and (ii) the adaptation of the solutions applied in the precedents to the new problem. Case-based reasoning techniques are appropriate on problems where many examples of solved problems can be obtained---like in our case where multiple examples can be easily obtained from recordings of human performances

CONTRIBUTION

My contribution to this project solved some problems that did not allowed the system to work in an automatic way without the intervention of the user. Some features of SMS had to be improved or changed to adapt it to the SaxEx algorithm and behaviour. The fundamental frequency (pitch) detection was adapted to get more accurate results in monophonic woodwind instruments [Cano98], also an automatic region segmentation was developed to automatically detect every note in an audio file and separate it into musical regions (attack, steady state, release, silence). Also a graphical user interface was created to integrate SaxEx and SMS algorithms in only one system and allow the user to control all the process and synthesize the output expressive sound.

2002

3.8 'TVC: a text to voice conversor'

Mayor, O. Loscos, A. Garrido, J. Lopez M. (2002)
MTG internal

ABSTRACT

A text to voice conversor (TVC) is a system able to generate in an automatic way the sequence of sounds that a human would produce when reading a text. The automatic reading that the TVC system performs has to take into account some characteristics of the language that is going to “speak”, like orthography, Grammatik, phonetic and prosody. The development of a high quality text to speech conversor in Catalan will help to promote the diffusion of interactive systems based in speech technologies. Some systems that will use this will be email readers and billing information for mobile phones, customer support services, disabled people systems, etc. The TVC is based on the technique of generating the synthetic voice from the controlled concatenation of acoustical units. In order to generate a natural output signal the system needs not only to do the concatenation of the units following an input text but also to modelate some prosodic parameters (duration of each one of the sounds, pitch intonation and loudness).

CONTRIBUTION

My contribution to this project consisted in the automatic creation of the “multi-phoneme units” database, the integration of the prosody module with the synthesis module, and some tasks involved in the concatenation and transformation of samples to create the output synthetic phrase. With the prosody information, the values of stretch of each database unit had to be calculated depending on the synthesis time durations. The pitch shifting ratio had to be applied following a pitch curve decided by the synthesis intonation, also extracted from the prosody information. The concatenation and transformation of units was done in the spectral domain [BLK03], previously analyzing all the database samples using the Spectral Peak Processing Technique [BL01] [LD99].

Chapter 4

Future research goals and conclusions

4.1 Performance analysis of the singing voice

We use the voice when we speak or sing. To speak or to sing is to move one's lips, tongue, jaw, larynx, and so on, while an air stream from the lungs passes the vocal folds. In this way we generate sounds that we call voice sounds. These voice sounds may be either speech sounds or sung notes, depending on the purpose of producing them. The voice also generates other types of sounds, for example, hawking, whispering or laughing.

The different structures that we move when producing a voice sound can be called the voice organ. The voice organ, composed by the breathing system, the vocal folds, and the vocal and nasal tracts can be considered as a tool for making sound and the singer uses this tool as a musical instrument [Sun87].

If we consider the singing voice as a musical instrument, it is the most flexible and fascinating one and also the most difficult to synthesize. The voice source is how we call the sound that is generated when the vocal folds are set into vibration by an air stream from the lungs. To arrive to a good description of the voice source, we need to define it in at least three dimensions: fundamental frequency (pitch), amplitude

(loudness) and spectrum (timbral characteristics). Changing these three dimensions will change voice quality. Once we control these parameters in a proper way we can obtain a desired and good synthesized voice, but this is not enough to have a good singing voice, another aspect we also have to take into account is expressiveness. The emotional state of a speaker has a considerable effect on the way in which the voice is used, the way the speaker articulates the sounds, the intonation and also voice timbral characteristics.

Performance of the singing voice is a big field where many research is being done and many aspects are still to be investigated. My future work for my on-going thesis research will consist in the next topics, always related with expression and performance aspects of the singing voice.

- Improvements in local and global expression models for the singing voice
- Mood emotions (expressive labels)
- Musically meaningful emotions (style labels) (jazz, blues, opera, rock, country, funky)
- Expressive Singing Performance Rating

4.2 Improvements in local and global expression models for the singing voice

Different techniques for giving expression to synthetic singing voice performances have been developed in the insides of a spectral singing voice synthesizer [BLK03] [BLMK03]. These techniques are based on templates extracted from recordings and mathematical models and pursue the idea of mimicking the voice behaviour of a professional singer. These models/templates can be divided into two groups: templates/models that have a local influence inside the song, just modifying the attack or release of a note, a note transition or a vibrato (local expression models) and other model/templates more general which apply changes that have influence all over a phrase or a complete song (global expression models) [MCBL03].

Regarding Local Expression Models, future directions in this research will be to create more complex mathematical models for attacks, releases and vibratos with controllable parameters that would offer some degree of adjustment to cover several expression emotions. Global Expression Models can also be improved a lot and even we can create new models that apply changes to get convincing results for fast singing performances where a lot of notes have to be stretched or skipped to get a natural sounding synthesis.

4.3 Mood emotions (expression labels)

When adding expression to a piece we can apply some basic rules that add a certain kind of expression or naturalness in certain parts of a piece, but when we are in a real

situation it is more important to give a certain global emotional intention to the piece rather than locally improve the naturalness of a certain part of the song. Performers usually prefer a synthesizer with more degree of expression rather than good quality or natural sounding.

We can define some emotional/expression labels and analyze what kind of local rules have to be applied to achieve the emotional intention desired. These local rules to be applied will include changes in tempo, loudness, articulation, pitch deviation and time deviation.

We will start with a small set of expression labels [BF00] including: Fear, Anger, Happiness, Sadness, Solemnity and Tenderness. Experts usually say that it is not completely clear how these labelled emotions are defined because performers and listeners often use different terms in describing intentions and perceived emotions [Cana97] [BF98] [PRV98] [OC98]. Although we will use a standard definition of these labels that will be appropriate for many performers and listeners.

In order to achieve convincing results when adding a certain emotional colouring to a piece of music it is very important to have first a non-expressive performance of the piece. Applying the set of performance rules regarding a certain emotion to an expressiveness performance would yield to non-desired results with strange emotions, we first should detect the degree of emotion and subtract it from the piece and it would be an impossible task without feedback from the performer that played the piece.

Changes in tempo are quite substantial when professional performers give a certain degree of emotion when playing a piece, for instance an irregular tempo is present when expressing fear emotion, a fast tempo is used for transmitting happiness and anger, and the tempo becomes slow when expressing sadness, solemnity or tenderness. Obviously the sound level of a performance has a lot of crescendos and decrescendos but it is important to maintain a global low, moderate or high loudness to distinguish between different expressed emotions. For tenderness the sound level should be very loud, also for fear, performers maintain a low level and this level becomes the highest for expressing anger. When expressing happiness, sadness or solemnity the loudness maintains a moderate or loud sound level.

The way how performers articulate notes also become a crucial factor when expressing emotions. Mostly staccato and non-legato transitions are used to express fear and anger and legato transitions are commonly used when expressing sadness, solemnity or tenderness to give a relaxation sensation to the performance. For transmitting happiness emotions legato and non-legato articulations can be used together.

Pitch and time deviations can also be used to give a certain degree of expression to the performance of a piece. So the use of slightly out-of-tune notes can express anger or fear when the tuning is higher than natural or sadness when the tuning is lower. Rubattos and swinging time deviations are used to change the groove of the piece and also the emotional colouring of it.

4.4 Musically meaningful emotions (style labels)

There are some timbre and performance characteristics in the singing voice, which are tightly related with styles of music, for instance, the timbre of a soul singer is quite far in similarity to the timbre of an opera or rock singer. Also the way a rock singer articulates the notes or performs a vibrato is quite different from a jazz to a country singer.

In these differentiation characteristics we can distinguish between voice quality parameters related to the timbre of the singer and execution parameters related to the performance.

Voice quality parameters	**Execution parameters**
Brightness	Attacks (hard, soft, sexy…)
Breathiness	Releases (long, short, bluesy…)
Huskiness	Vibratos (regular, irregular, depth & rate…)
Hoarseness	Transitions (legato, bend, portamento…)
Nasality	Intonation (moderated, exaggerated…)
Deepness	Loudness (loud, sweet…)
Falsetto	Swing (time deviations)
Shout	Tuning (pitch deviations)
Roughness	

Figure 11: voice quality and execution parameters for applying musically meaningful emotions (jazz, blues, pop, opera…) to a performance.

As shown in the table in figure 11 voice quality parameters refer to techniques developed by professional singers to change their voice timbre to mimic a style of singing or a famous singer ideal voice. In the case of execution parameters, we refer to the way the singer performs an attack, a vibrato location and depth and rate parameters, or pitch deviations and time deviations during the performance.

So a combination of parameters of these two categories will be used to apply a certain degree of expression to a piece following a style of music. Many recordings of professional singers will be recorded including styles of singing like Jazz, Blues, Rock, Pop, Country, Opera, Reggae and Flamenco and analyzing this recordings some general performance rules will be created. These rules could be used to apply a certain style

expression to a neutral performance (a performance with no determined style) or to determine the style of music in a performance as in a genre classification system.

4.5 Expressive Singing Performance Rating

The idea of a performance rating system is to determine how well a user performs a certain piece, based in a standard way of performing it or based in a comparison with a previously performance of a professional singer that the user has to mimic. A performance rating system can have numerous applications like judgement in singing contests, karaoke performance rating or virtual singing education.

When we talk about expressive singing performance rating, we are putting emphasis in analyzing the expression of the singer when performing a song, so we are not only rating how well the user performs the song but also what degree of expression or in what mood a singer is performing. This is tightly related with the previous research directions and is one of the applications of it, so once we have established some performance rules about singing with expression, we can analyze a performance to determine which rules are following the executions to rate the degree of expression and mood of each performance.

To rate a performance, we need to analyze the singing-voice and extract some parameters of the voice. It is very useful to have a written score of the melody and chords that are being performed to help in a better analysis and a better rating of the performance:

- Pitch: Detect the pitch deviations over the melody to detect possible out-of-tune situations.

- Timing: Perform a note onset detection to evaluate how well the user follows the tempo of the song and analyze time and rhythm variations or deviations to decide if they are intended to give a certain degree of expression or if they are due to a bad performance.

- Dynamics: Analyze the continuous loudness along the time to detect if the performer is following a good intonation, marking the attacks, releases and vibratos correctly and not yielding in a monotonous interpretation.

- Timbre: Analyzing the user's singing voice timbre we can report the characteristics of the voice in several aspects like brightness, breathiness, huskiness, hoarseness, nasality, deepness, and falsetto. These characteristics will be rate as positive or negative, depending on the song being sung and the location where a certain timbre is used.

- Note articulation: The use of legatos, staccatos and bending between notes is an important factor to rate a performance, so we have to analyze the execution of the articulations that performs the user.

With these extracted parameters, we have to define a rating system that gives an idea of how well the user mimics a good performance and which are the weak points of the performance.

As an extension of the singing performance rating, the users can be supplied with some feedback about the type of expression or emotion that are giving to the audience, just to compare their intentional emotion with the one detected by the system, and some clues to improve their performance or their emotion.

The idea is that the system will give as a report a detailed description of the performance execution explaining how well or how bad the user has performed, and which are the weak points of the execution: where he is expressing in a certain mood or where the user is shouting or doing a falsetto, when is out of tune or performing a weird vibrato, etc. The aim is to get a recommendation as it would have been done by a singing professional teacher. With this report, the user is able to realize about his faults and mistakes and improve his singing skills.

Chapter 5

Appendix
- full papers -

Statistical Significance in Song-Spotting in Audio

Pedro Cano, Martin Kaltenbrunner, Oscar Mayor, Eloi Batlle
Music Technology Group
IUA-Pompeu Fabra University
{pedro.cano,modin,oscar.mayor,eloi.batlle}@iua.upf.es

ABSTRACT

We present some methods for improving the performance a system capable of automatically identifying audio titles by listening to broadcast radio. We outline how the techniques, placed in an identification system, allow us detect and isolate songs embedded in hours of unlabelled audio yielding over a 91% rate of recognition of the songs and no false alarms. The whole system is also able of working real-time in an off-the-shelf computer.

1. INTRODUCTION

A monitoring system able to automatically generate play lists of registered songs can be a valuable tool for copyright enforcement organizations and for companies reporting statistics on the music broadcasted. The difficulty inherent in the task is mainly due to the difference of quality of the original titles in the CD and the quality of the broadcasted ones. The song is transmitted partially, the speaker talks on top of different fragments, the piece is maybe playing faster and several manipulation effects are applied to increase the listener's psycho-acoustic impact (compressors, enhancers, equalization, bass-booster, etc…). An additional difficulty is that there are no markers in broadcasted radio informing when the songs start and end.

In this scenario, the article focus on the pattern matching techniques that, given a sequence of audio descriptors, are able to locate a song in a stream avoiding false alarms. Shortly the whole system works as follows, off-line and out of a collection of music representative of the type of songs to be identified, an alphabet of sounds that describe the music is derived. These audio units are modeled with Hidden Markov Models (HMM). The unlabelled audio and the set of songs are decomposed in these audio units. We end up then with a sequence of letters for the unlabelled audio and a database of sequences representing the original songs. By approximate string matching the song sequences that best resembles the audio the most similar song is obtained. We point out the importance of assessing statistical relevance on the best matching song found in order to avoid false positives. We end up explaining how these techniques can be applied to continuous stream of audio and commenting the results.

2. AUDIO PERCEPTUAL UNITS

From an acoustic point of view, music can be described as a sequence of acoustic events. To be able to identify titles it is relevant to extract information about the temporal structure of these sequences. The first step converts the acoustic signal into a sequence of abstract acoustic events. Speech events are described in terms of phones. In music modeling this is not so straightforward. Using, for instance notes would have disadvantages: Often notes are played simultaneously (accords, polyphonic music) and music samples contain additional voices or other sounds. The approach therefore followed is learning relevant acoustic events, that is, finding the set of "fundamental sounds" in which we can decompose audio and representing them with a letter. The alphabet of audio perceptual units is derived through unsupervised clustering using cooperative HMM from a database of several thousand titles [1].

3. SEQUENCE ALIGNMENT

Having derived HMM models for the audio perceptual units, we can decompose the songs into a symbolic representation. Instead of comparing raw audio, for identifying titles, we compare the sequence of letters of unknown audio against the sequences corresponding to all the songs to identify. The search for a sequence in a database similar to the query sequence is performed by approximate string pattern matching [2]. A measure of the difference between two sequences is the edit distance, defined as the minimum number of character insertions, deletions and substitutions needed to make them equal. An arbitrary weight can be associated with every edit operation, as well as with a match.

The dynamic programming algorithm is guarantied to find the best alignment between a pair of sequences given a particular choice of scoring matrix and gap penalties [3]. There are several variants of the dynamic programming algorithm that yield different kinds of alignments. The Neddleman and Wunsch is a global alignment, that is to say, it aligns the entire length of both sequences. For our particular case this is not suitable since it is typical that a song in the radio is broadcasted partially. The variant known as the Smith-Waterman algorithm yields a local alignment. It aligns the pair of regions within the sequencesIn our application, since the query audio sequence must be compared to several thousand titles, we run a heuristic approximation to the Smith-Waterman algorithm that allows us perform the matching much faster named FASTA[4].

3.1 The choice of substitution scores

The weighted scores for substitutions of the edit distance are calculated to account for bias in the replacement of symbols between the original and the broadcasted song sequences. A set of original CD and corresponding radio songs are selected and manually edited by cutting pieces so that the pieces of audio are synchronized. Then a similarity ratio, R_{ij} is computed for the symbols in the sequences

$$R_{ij} = \frac{q_{ij}}{p_i p_j}$$

where q_{ij} is the relative frequency with which the symbols i and j are observed to replace each other in the manually aligned sequences. p_i and p_j are the frequencies at which the symbols i and j occur in the set of songs in which the substitutions are observed. Their product, $p_i p_j$, is the frequency at which they would be expected replace each other if the replacements were random. If the observed replacement rate is equal to the theoretical replacement rate, then the ratio is one ($R_{ij} = q_{ij} / p_i p_j = 1.0$). If the replacements are favored with the manipulative effects above described the ratio will be greater than one and if there is selection against the replacement the ratio will be less than one. The similarity reported in the similarity score matrices S_{ij} is the logarithm to this ratio.

4. STATISTICAL SIGNIFICANCE

Considering the possible uses of the system, a great concern in the similarity searching above described is a false-positive error. We would not like to include in a play list for a copyright enforcement association a song that has not been played. Any two sequences composed of letters from the same alphabet can be aligned to show some measure of similarity. Typically alignment scores of unrelated sequences are small, so that the occurrence of unusually large scores can be attributed to a match. However, even unrelated sequences can occasionally give large scores in the local alignment regime. Although these events are rare, they become important when one attempts a search of a big and expanding sequence database. How often will an event at least as extreme as the one just observed happen if these events are the result of a well defined, specific, random process? It is imperative to understand the statistics of the high-scoring events, in order to estimate the statistical significance of a high-scoring alignment.

In the case of gapless alignment, it is known rigorously [6] that the distribution of alignment scores of random sequences is the Gumbel or extreme value distribution (EVD), which has a much broader tail than that of the Gaussian distribution. For the case of gapped alignment, there is no theory available to predict the distribution of alignment scores for random sequences. It has been conjecture that the score distribution is still of the Gumbel form. Also our tests on sequence of descriptors extracted from audio seem to show a good fit to the Extreme Value Distribution. The EVD is of the form:

$$E = Kmne^{-\lambda S}$$

where E is the expected number of hits with score >=S, m is the size of the query sequence, n is the size of the database. λ and K the are Gumbel constants and must be estimated from a large scale comparison of random sequences. The FASTA or various implementation of the SW algorithm, produce optimal alignment scores for the comparison of the query sequence to sequences in the database. Most of these scores involve unrelated sequences, and therefore can be used to estimate λ and K.

5. ON-LINE SYSTEM

We have then a method for comparing fragments of audio against a database of songs for a best match and statistical method for assessing its goodness. Both the symbolic extraction and the matching against the database run fast on a normal machine. The approach for, having a continuous stream of broadcasted audio, identify songs consists in sending hypothesis to match against the database every few seconds. That is, the superstring resulting from the conversion of the raw audio to symbols is windowed with overlap. So every 10 seconds, a sequence corresponding to two and a half minutes of sound is compared to the database. As a result of each comparison a set of candidates is shown along with its expectation (E-value). A candidate with sufficiently low E-value suggests that the query is related to that candidate sequence and therefore can be added to the play list. Along with the candidate sequence, an alignment with the query is provided. With the timing associated to the query sequence an estimation of the beginning and ending time of the song broadcasted can be obtained and printed in the play list.

6. RESULTS

The system has been tested with 24 hours of radio recorded from 10 different stations against a database of around 2500 songs of commercial music. The radio data contains among music, jingles commercials... 147 songs registered in the system (its original version is in the database). The system yields a result of 133 (little over a 91%) songs recognized and no false positive. By lowering the threshold of acceptance of a candidate raises the results to 135 correctly identified but false positives appear as well. When working on-line, the delay between the moment a song starts sounding and it is added correctly to the play list is about one minute as average. The system runs in more than real-time in a Pentium III 500Mhz.

7. REFERENCES

[1] Batlle, E., Cano, P., **"Automatic Segmentation for Music Classification using Competitive Hidden Markov Models"** Proceedings International Symposium on Music Information Retrieval (2000)

[2] Gusfield, D., **"Algorithms on Strings, Trees and Sequences"** Cambridge University Press (1997)

[3] Smith, T.F. and Waterman, M.S., **"Identification of common molecular subsequences"** Journal of Molecular Biology. (1981), 195-197.

[4] Pearson, W.R. and Lipman, D.J. **"Improved tools for Biological Sequence Comparison"** Proc. Natl. Acad. Sci. (1988) 85 : 2444-2448.

[5] Nicholas, H.B., Deerfield D. W., Ropelewski, A.J. **"A Tutorial on Searching Sequences Databases and Sequence Scoring Methods"** (1997)

[6] Karlin, S. And Altschul, S.F. **"Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes"** Proc. Natl. Acad. Sci. USA 87 (1990), 2264-2268.

An adaptative real-time beat tracking system for polyphonic pieces of audio using multiple hypotheses

Oscar Mayor
Music Technology Group, Pompeu Fabra University
oscar.mayor@iua.upf.es, http://www.iua.upf.es/mtg

ABSTRACT

In this paper a method for extracting beat information of a piece of music is presented, a real-time analysis is performed while the music is played or recorded from any source and the system gives the beats per minute value at each moment and beat occurrences in time. It also becomes adaptative in case of a sudden or smooth change in the tempo. It deals with multiple hypotheses, and gives the most suitable results at each time. One of the most important applications linked to this work is the automatic classification of pieces in different musical genres or finding song similarities in terms of musical rhythm.

1. Introduction

To extract rhythm information from a piece of music has become a difficult task in the scope of computer music research. To achieve a musical representation of the rhythm of a song in a polyphonic recording including not only percussive instruments is not giving good results yet, although some research has been done and there are satisfactory results for drum loops, in terms of music transcription. One of the most important topics related to this work is the automatic classification of pieces in different musical genres or finding song similarities in terms of musical rhythm. One of the first steps to achieve a high level of rhythm description is the beat level. It means to detect beat occurrences in time, the beats that a listener usually tap with his foot while listening to a song, and derive BPM (beats per minute) information from it. Beat occurrences commonly match with points of maximum energy of the sound or at least correspond with points of high energy. In this article a method for extracting BPM information of a song is presented and some conclusions about applications and future work are discussed.

2. Previous work

Previous approaches related to rhythm tracking and beat induction include several approaches.

Goto and Muraoka [1] [2] present a method, that works in real-time in a parallel-processing computer, that extracts drum patterns from a musical signal and uses a template-matching model to determine the beat of the song being analyzed. They also have presented a system that works with drum-less audio signals [3].

Scheirer [4] also presents a real-time beat-tracking system, which using a small number of band-pass filters and banks of parallel comb filters extracts the beat from musical signals of arbitrary polyphonic complexity. This system can be used to predict when beats will occur in the future.

Dixon [6] presents a bottom-up approach to beat tracking from acoustic signals deriving time signature and approximate tempo from the timing patterns of detected note onsets.

Desain and Honing [7] have made a lot of research in computational modeling of beat-tracking, their models begin looking for inter-onset intervals associating a rhythm pulse with the interval stream.

Gouyon [8] have proposed a method for classification of drum-loops detecting the minimum inter-onset interval which he calls the tick, and looks for percussive templates in the most relevant ticks of the drum loop.

3. System overview

The flow diagram in figure 1 shows the functionality of the system and can be followed to easily understand the processing of the system.

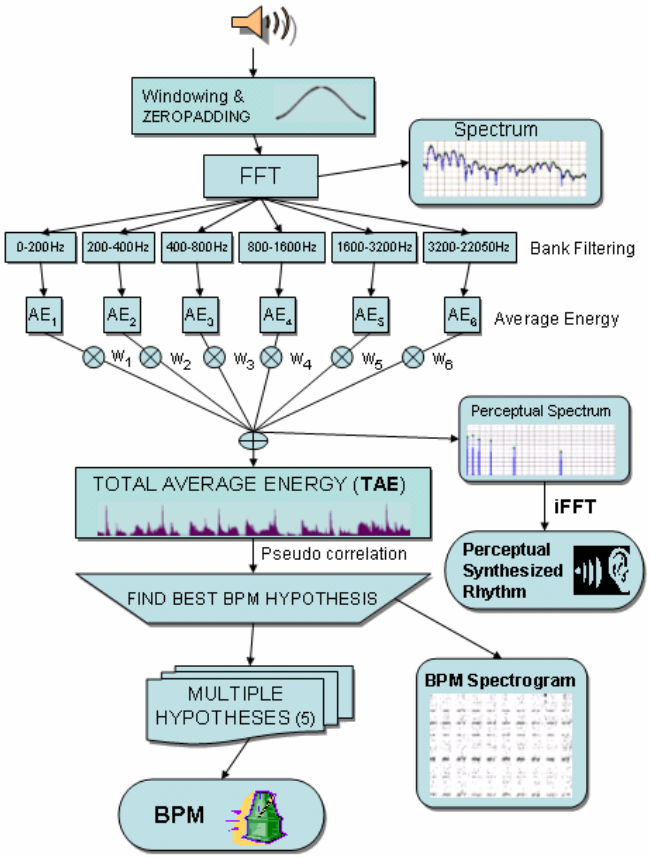

Figure 1: System overview of the beat-tracking system

First of all we apply a smoothing window and zero-padding to the sound source which is going to be analyzed and then we do the Short Time Fourier Transform that produces a Frame-by-Frame Frequency Spectrum of the input signal. Then we apply a bank of filters over the magnitude spectrum to calculate the energy evolution in time of each filter, and multiplying this energy evolution by an arbitrary factor for each filter we get a Total Average Energy that will be the basic information where to look for BPM candidates. A pseudo-correlation method that will be explained later in section five, searches for BPM candidates and adds them to a list of hypotheses from where we derive the most probable hypothesis at each moment.

We also synthesize a perceptual rhythm from an important simplification of the input audio data and we display what is called the BPM spectrum, that is explained in detail in sections four and six respectively.

4. Perceptual beat energy extraction

The system accepts any kind of audio data as input, including the compressed MP3 format, and this data is automatically converted to PCM raw audio with a quality that can be changed by the user: mono/stereo, 8/16 bit, 22050/44100Hz.

Good results are obtained dealing with mono, 16 bit, 22050 Hz, PCM raw audio, so there is no need to work with better quality because the calculation time will increase a lot and the results will be more or less the same.

Assuming that beat occurrences usually match with maximum energy points of the waveform or at least points with high energy, we can rely on energy to find beats in a musical excerpt. We perform a perceptual simplification of the audio data in order to work with few data but with enough perceptual information to perceive the rhythm of the song. The perceptual simplification is done in the frequency domain, so first of all an smoothing analysis window is applied to each frame of audio data (512 samples - 23 ms), with half overlap so we have a minimum time resolution of 11,5 ms, and then the Short Time Fourier Transform (STFT) of each frame is done applying zero padding to get an smoother spectrum with 1024 bins. Once in the frequency domain, a bank of filters like the one presented in [4] is calculated. One low-pass (0-200Hz), one high-pass (3200-22050Hz) and four band-pass filters (200-400Hz, 400-800Hz, 800-1600Hz, 1600-3200Hz) are created and the Average Energy of each filter (AE_j) is calculated in this way:

$$\sum_{j=1}^{nFilters} AE_j = \frac{\sum_{i=left}^{right} MagSpectrum[i]^2}{right - left}$$

where nFilters is the number of filters, in our case six, left and right are the left and right bins corresponding to the frequency cut-offs of each filter's spectrum and MagSpectrum[i] is the magnitude value of the i-th bin. If we listen to the synthesized sound calculated computing the inverse fft of an spectrum with six peaks, each one

representing the middle frequency of each filter, modulated to the average amplitude of the filter that it represents, we can perceive a sound that inherits a reliable representation of the rhythm of the original sound. So an important simplification of the original data has been done preserving the necessary rhythm information for later processing.

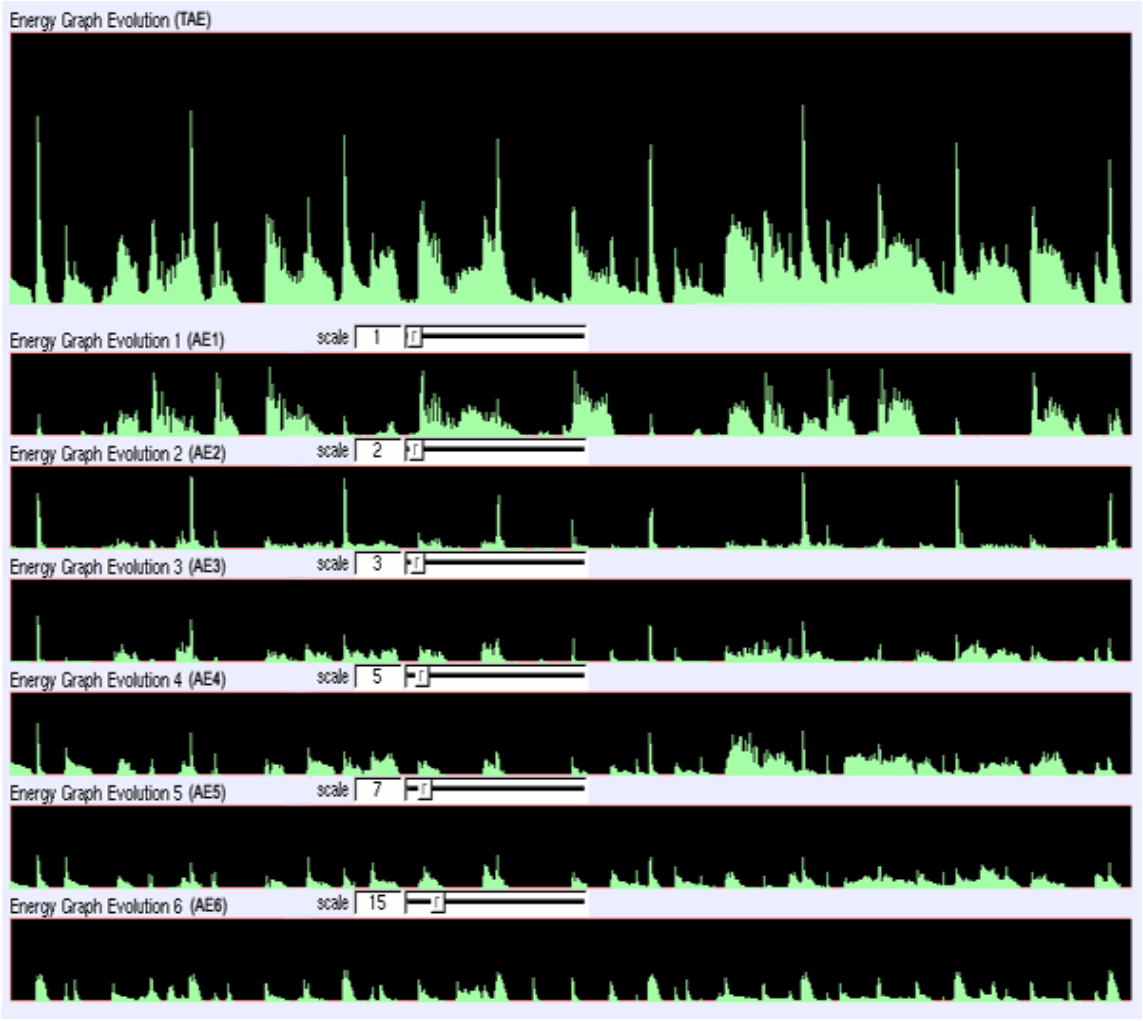

Figure 2: TAE and ponderated AE Graph Evolution for each filter, peaks in the TAE graph represent beats, we can see that high beats are equally separated, representing a constant BPM value.

In figure 2 we can see a graphical representation of the evolution of the energy for each filter along the time, so high peaks in the y-axis represent instants of time where the average energy of the filter is very high and these points are good candidates for beats.

The Total Average Energy (TAE) also shown in figure 2, is calculated as the ponderated sum of each filters' average, divided by the number of filters, in our case six.

$$TAE = \frac{w_1 \cdot AE_1 + w_2 \cdot AE_2 + w_3 \cdot AE_3 + w_4 \cdot AE_4 + w_5 \cdot AE_5 + w_6 \cdot AE_6}{\text{nFilters}}$$

In order to extract the BPM (Beats Per Minute) information, we will focus in the Total Average Energy evolution along the time.

5. Maximum Beat Correlation

Once we have the beat information along the time (energy graph evolution, see figure 2), we try to find equally distant beats in time in order to find the correct BPM for the part of the song being analyzed. We search for candidates only when a high energy beat value is detected, this will happen only when the energy of the current beat is at least twice the average of some previous frames (last 100 frames in this implementation). To find the BPM value we use some kind of time-domain Comb-Filtering algorithm over the energy graph evolution (beats along the time) to obtain high energy and equally distant beat repetitions that will determine the most adequate BPM of the musical excerpt being analyzed.

We calculate the sum of some equally separated time-positions' energy, and then varying the separation between points into a limited range, determined by the maximum and minimum BPM value that we accept (between 50 and 200 BPM), we obtain a list of values each one representing the score of each possible BPM, then we store these values in an array, that once sorted will show the more probable and less probable BPM values at each time. This method gives the number of frames between high beats, this value is converted to a BPM value with the next conversion:

$$Tempo\ (BPM) = \frac{1}{GAPFRAMES \cdot \frac{SIZE}{2} \cdot \frac{1}{SAMPLINGRATE}} \cdot 60$$

where GAPFRAMES is the number of frames between high beats, SIZE is the number of samples in a frame (512) and SAMPLINGRATE is the sampling rate of the sound (22050/44100 Hz).

6. BPM Spectrogram

In order to represent graphically the results of the previous analysis, so a user can derive BPM information from it, several graphics have been represented showing values of energy, time position and distance between beats. In a two dimensional space we can only represent the best BPM value calculated at each moment, so y-axis will represent the distance between beats (from where we can derive BPM), and x-axis will represent the evolution in time, the problem with this representation is that we are discarding BPM values calculated with lower score but very close to the maximum that could be the correct tempo value at this moment. So the best way consists in displaying three parameters in a 3-axis representation, similar to the Beat Spectrum idea presented by Foote [5]. X-axis will represent time, y-axis will represent the BPM value or distance

between high beats and z-axis, the importance of this BPM value in terms of correlation of an equally distant peaks template with the energy values at an interval of time assuming this BPM. (score of this BPM).

In figure 3 we can see this graphical representation in 2-D while representing the third axis with color depth as a gray-scale, where white represents the lowest value and black the highest value. We can see the darker horizontal line that represents the most probable BPM for the song; other lines represent half and double tempo or beats that follow a constant pattern of repetition but do not match with the multiples or submultiples of the main beat.

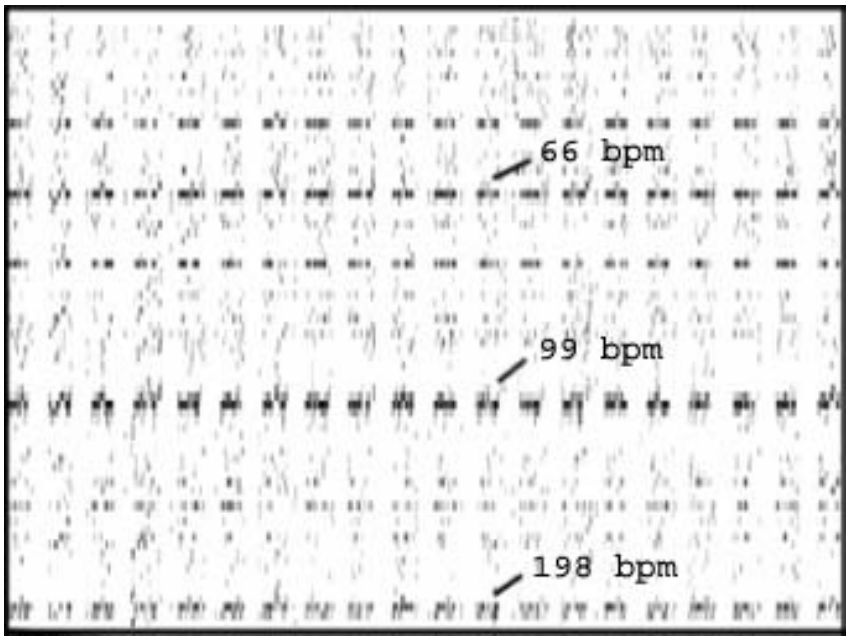

Figure 3: In this BPM spectrogram, we can see the darker horizontal Line representing the correct bpm (99bpm) and other lines representing double tempo (198bpm) and 2/3 of tempo (66bpm).

7. Multiple Hypotheses

With a graphical representation like the so-called BPM Spectrogram that we have presented before, it's quite easy for a human to decide what BPM mostly identifies a song, but for a computer it's a more difficult task, the idea is to work with multiple hypotheses during the analysis process, and decide which one of the hypotheses is the best at each moment, because we're trying to get the BPM of the song in real-time when the song is being recorded or played. We deal with multiple BPM hypotheses in order to give a reliable result. In case of a changing tempo in a small interval of time that should be considered as artefacts instead of a tempo change, the system is conservative and robust enough to consider the most possible hypothesis as the correct one, we use a different approach as the one presented in [2].

Each time that a high beat occurs, the system calculates the best BPM candidates with the method described before and new hypotheses are added to the system, that decides which hypotheses are consistent, or what hypotheses are no longer valid.

The algorithm used to decide what hypothesis should be considered valid or wrong can be easily adapted in order to make the system more robust or more flexible to tempo changes, and also to tune the accuracy of the system, but this is always a compromise between quality and functionality.

For each hypothesis we have six properties: The BPMvalue, Value, Hits, HowOften, Duration and Score. BPMvalue is the number of beats per minute, Value represents the average correlation value for all the times that this hypothesis has appeared, Hits is the number of total apparitions, HowOften is the frequency of appearance in the last 100 calculations, Duration is the time since the first appearance of the hypothesis and Score is the score given to the hypothesis calculated from the rest of properties.

When an hypothesis comes up, first of all it's compared with the existing and if it's similar to one in the list, its score is recalculated or if it's new, it's added to the list replacing the worst one. The Score of the hypotheses is calculated multiplying their average value (Value property) with the HowOften value. Other properties like Hits and Duration are important to know if an hypothesis is new or old or if it's active.

At the same time that we calculate the score of the hypotheses, we decide which one is the worst, that will be replaced when a new one comes out.

Some assumptions are made that affect the scoring of the hypotheses:

- A very recently added hypothesis is scored higher than an old hypothesis.
- A very active hypothesis is never discarded even if it gets a low score.

So we try to never reject the new or very active hypotheses, even if they get bad punctuation. In these cases, these hypotheses remain in the list, and the next bad hypothesis is replaced by the new one.

8. Beat occurrences in time

Once the tempo of the song is "correctly" detected, the next step is to extract higher level information about the rhythm patterns of the song. This is a difficult task because if we work with different genres, it's not helpful to try to find spectral shapes to discriminate an instrument because a snare drum or tom sound has a different sound across genres and also different spectral shape from one style to another.

We have to deal even with music with no drums or even without any percussive instrument, which doesn't mean that it has no rhythm, it's just that the rhythm is more present in the bass lines, guitar chord progressions or in any other instrument. This is one of the problems that systems based in pattern matching [2] [8] encounter when there is no snare or bass drum to search for.

In the actual implementation we have information about when the beat occurs, so the deviation between the predicted beat and the real beat can be easily calculated and derive the level of expressivity, in this case the rubato. We have also information about secondary beats, often with less energy than the main beats, which doesn't occur at the beat level but at half, quarter or third tempo. This information characterize the rhythm pattern of the song and is the basic information to extract high level attributes that will make it easy to find, with this rhythm information, similarities between songs.

9. Conclusions and future work

The beat-tracking system has not still been systematically tested. There are a lot of parameters that can be adjusted to tune the accuracy or functionality of the system like filters' width, scaling factors, high beat threshold, frame size, type of window, zero-padding and overlap factors, hypotheses decisions, etc. Many of these parameters can be adjusted in real-time by using a complete GUI.

With a large song's database including several genres and styles, the system will be fully tested and these parameters adapted to achieve the best results for all the songs or even adapt the system during the analysis process to improve the results depending on the song that is being tracked.

Once that the BPM of a song is correctly detected, the next step is to find more information about the rhythm of the song, rhythm descriptors, transcription, finding patterns of repetition, and another high level attributes that give information about the kind of rhythm that is being analyzed. It would also be important to focus in other aspects like chord detection or changes in the melody and harmony to help the system to find similarities between songs, and not only looking for rhythm similarity but also close harmonic or melodic structures. This is part of the current work, that will lead to an extensible content based analysis system.

10. References

[1] Goto, M., Muraoka, Y., **"An audio-based real-time beat tracking system and its applicatios"**, in *ICMC Proceedings 1998*.

[2] Rosenthal, D., Goto, M., Muraoka, Y., **"Rhythm tracking using multiple hypotheses"**, in *ICMC Proceedings 1994*.

[3] Goto, M., Muraoka, Y., **"Real-time rhythm tracking for drumless audio signals – chord change detection for musical decisions"**, in *IJCAI-97 Workshop on computational auditory scene analysis*, pp. 135-144, 1997.

[4] Scheirer, E., **"Tempo and beat analysis of acoustic musical signals"**, in *J. Acoust. Soc. Am.* 103(1), jan 1998, pp 588-601.

[5] Foote, J., Uchihashi, S., **"The beat spectrum: A new approach to rhythm analysis"**, in *IEEE International Conference on Multimedia & Expo 2001*, Tokyo, Japan.

[6] Dixon, S., **"A beat tracking system for audio signals"**, in *Proceedings of the Conference on Mathematical and Computational Methods in Music*, Vienna, Austria, Dec. 1999, pp 101-110.

[7] Desain P., **"A (de)composable theory of rhythm perception"**, in *Music Perception 9*, 439-454.

[8] Gouyon F., Herrera P., **"Exploration of techniques for the automatic labelling of embedded instruments in audio drum tracks"**, in *Mosart Workshop*, Barcelona 2001.

Real-time spectral synthesis for wind instruments

Oscar Mayor, Alex Loscos, Jordi Bonada
Music Technology Group, Pompeu Fabra University
{omayor,aloscos,jbonada}@iua.upf.es, http://www.iua.upf.es/mtg

ABSTRACT

A real-time spectral domain monophonic synthesizer is being developed. The idea is to achieve the sound quality given by samplers and improve the flexibility of control and model transitions between notes. The output sound is generated concatenating, transforming and synthesizing attacks and stationeries previously recorded by a real player that have been analyzed and stored in a database. A spectral technique called SPP is used in order to get better sound quality than other spectral techniques, like the sinusoidal or sinusoidal plus residual model, when transforming the sound. The aim is to get a sampler-like quality but with a high degree of flexibility to transform the sound and to model note transitions.

1. Introduction

In this article we describe the progress report of a current project and the future work that is being done. The project consists in a real-time monophonic spectral synthesizer that has been applied successfully to brass and reed instruments like the trumpet and the saxophone.

The synthesis process has been changed [1] from a sinusoidal plus residual model [2] to a pure spectral model, where the spectrum is divided in a set of regions, each region representing a harmonic spectral peak and its boundaries and we apply timbre and pitch transformations preserving the regions.

The synthesis is controlled with a set of parameters that can be changed via midi or with a complete graphical user interface.

There is a database with pre-analyzed samples that will be selected accordingly in the synthesis stage depending on the midi events that came in.

In figure 1 we can see a flow diagram of the synthesis process that is explained later in this article.

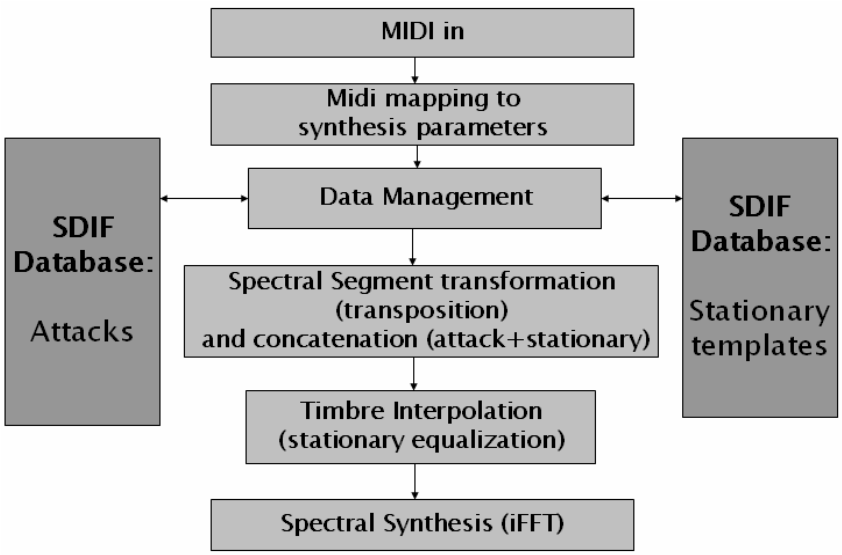

Figure 1: Flow diagram of the synthesis process.

2. Spectral Peak Processing

Spectral Peak Processing is a technique based on considering the spectrum as a set of regions, each of which belongs to one spectral peak and its surroundings [3]. The goal of such technique is to preserve the convolution of the analysis window after transposition and equalization transformations. The local behaviour of the peak region should be preserved both in amplitude and phase after transformations. To do so, the delta amplitude relative to the peak's amplitude and the delta phase relative to the peak's phase are kept unchanged after spectral transformations. The region boundary is set to be at the lowest local minimum spectral amplitude between two consecutive peaks or if there are no local minimums, at the middle frequency between two consecutive peaks.

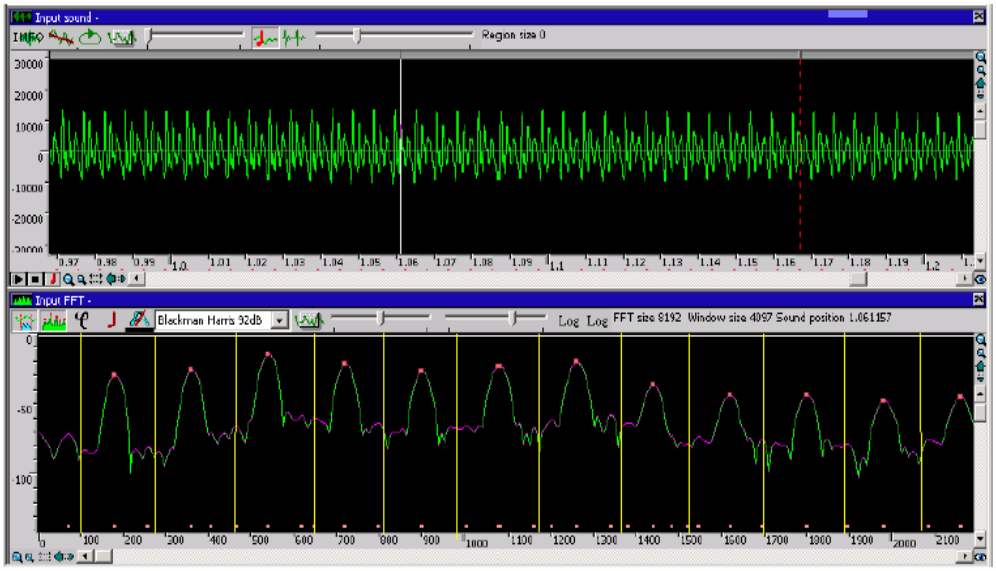

Figure 2: SPP analysis region segmentation.

In figure 2 we can see an example of a segmentation of one STFT spectrum. We can distinguish the little squares representing the harmonic peaks and the vertical lines representing the boundaries of each region.

We can apply basically two kinds of transformations over the spectrum: equalization and transposition.

2.1. Equalization

Equalization means a timbre change. From the SPP analysis we get a spectrum with the harmonic peaks (square points in the figure) and the regions. Then we want to change the spectrum to follow a desired envelope (transversal line in the figure) that will change the timbre to be adjusted to the target one.

The SPP equalization is done by calculating for each region the amount needed so that each harmonic's amplitude matches the desired envelope (in the figure we can see the displacement marked with arrows). Then this amount is added to all the bins that belong to a region (amplitude linear addition). Therefore, the spectrum amplitude of a region will be just shifted up or down and it will keep its local behaviour. The phase is unchanged.

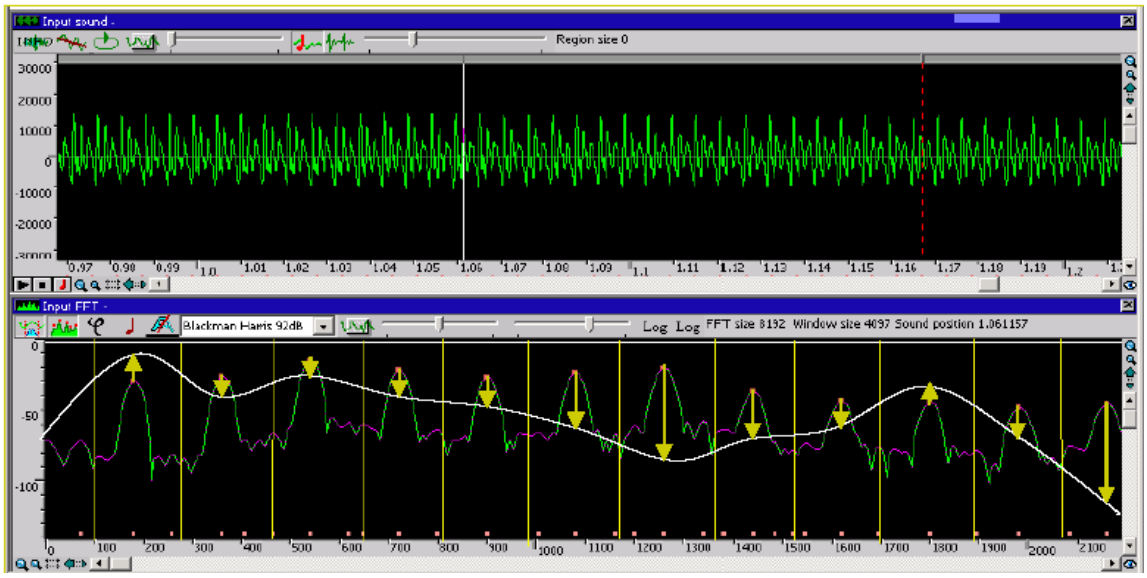

Figure 3: SPP equalization

2.2 Transposition

Transposition means to change the pitch of a spectral frame by multiplying the harmonic's frequencies by a constant value. In SPP, this operation can be done by shifting SPP regions in frequency.

In the figure 4 we can see graphically this process for transposing up. The arrows show the displacement in frequency that is applied to each harmonic. At the bottom we can see the resulting spectrum. As we can see, the amount of frequency shifting calculated for each harmonic peak is applied as a constant to its whole region, so the linear frequency displacement for all the bins in a region will be the same. Therefore, the local

amplitude spectrum of each region will be kept as it is, thus preserving the window convolution with each harmonic peak.

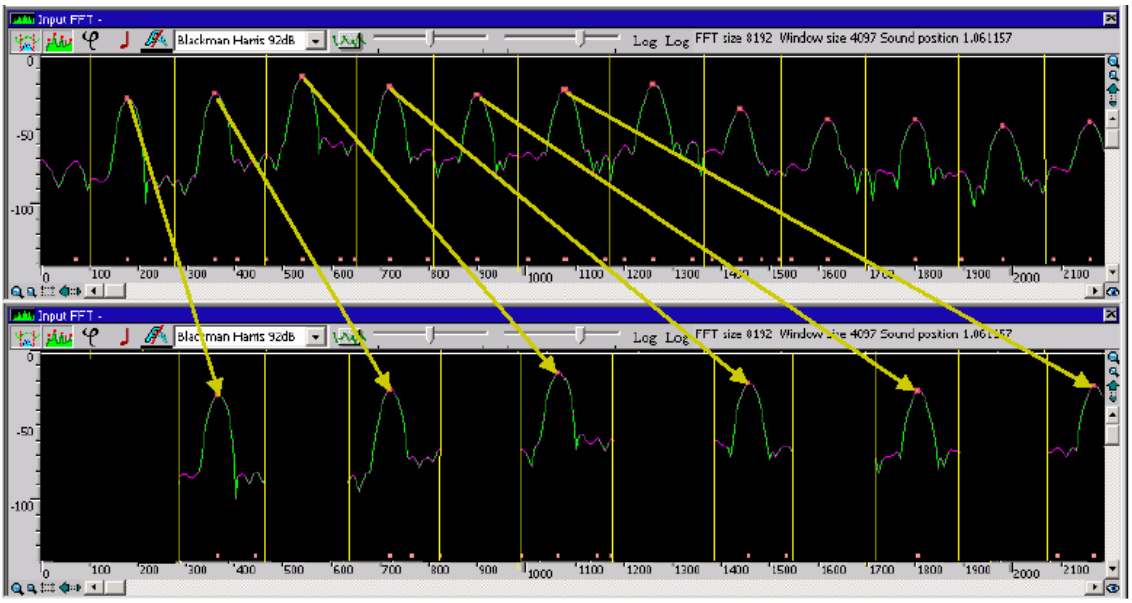

Figure 4: SPP transposition

The frequency displacement will be a non integer value in most cases. Therefore the spectrum should be interpolated. A spline interpolation of 3rd order to deal with that is a good compromise of quality versus computational cost.

In the previous figure we can see an example of transposition to a higher pitch. The SPP regions are separated in the resulting spectrum with holes that are filled up with spectral amplitudes of constant amplitude -200dB.

In the other hand, when we are transposing to a lower pitch, the SPP regions will overlap in the resulting spectrum. This overlapping is achieved by adding the complex values at each spectral bin.

3. Database

As we are not creating new sounds, but imitating a real instrument like the trumpet or the saxophone, we have a complete recording of these instruments performed by a professional player. These recordings are analyzed using the SPP technique and stored in a database that will be used by the synthesis process to synthesize the output sound.

For each instrument we want to synthesize, we will have in the database 4 different attacks recorded for each pitch from very soft to very hard. The attack corresponds to only the first tens of milliseconds of a note, usually less than 0.1 seconds.

The database also contains some stationary templates (two or three pitches for each octave) also with up to four different dynamics from very low/soft to very hard/aggressive. These stationary templates will be at least 1 or 2 seconds long and have to be recorded without any kind of expression like vibrato or reverb. As we don't have stationeries for all the notes in the scale, we will have to transpose the same stationary to achieve some different pitches, we will always use a lower pitch stationary than the

desired pitch to synthesize and will transpose it up, to get better results and avoid the overlap between regions when we transpose down.

Once these sounds are analyzed, the output of the analysis is stored in the database in an SDIF file [4] that will be loaded in memory at the beginning of the synthesis process.

4. Real-Time Synthesis and Processing

The synthesizer is controlled in real-time by midi, the idea is that the processing core receives the midi signals and chooses the segments accordingly to the processing algorithm and concatenates them to produce a natural sound.

4.1 Midi Control and Mapping

The current implementation of the synthesizer can be controlled with a complete GUI but also via MIDI. An input midi file will produce an output wave file with the melody synthesized, but the most interesting way of control is the implementation of real-time midi control with a simple midi keyboard or with the midi breath controller Yamaha WX5.

There are two kinds of incoming MIDI information, the initial controls and the continuous controls. In the next tables we can see this controls and who changes them in the case of the midi keyboard and the breath controller:

	MIDI keyboard	**Yamaha WX5**
Initial pitch	Key pressed	Fingering position
Attack velocity	Key velocity	Initial breath speed
Note On/off	Press/Release key	Fingering changes

Table 1: initial controls

	MIDI keyboard	**Yamaha WX5**
Note volume	After-touch value	Continuous breath speed
Pitch modulation	Pitch bend wheel	Lip pressure

Table 2: continuous controls

These controls are mapped to parameters of the synthesizer, following a divergent mapping that means that each midi control can be mapped to more than one synthesis parameter [5], in the next table we can see which parameters are controlled by which control.

MIDI controls	Synthesis Parameter
Initial pitch	Pitch (note)
Attack velocity	Attack type Synthesis volume
Note on/off	Note on/off Transition recognition
Note volume	Volume Stationary timbre interpolation Note off
Pitch modulation	Relative pitch modulation

Table 3: mapping controls/synthesis parameters

4.2. Attack Stage

The midi attack velocity value that arrives with a note on will decide which kind of attack will be selected from the database to synthesize the output sound; the higher the attack velocity the harder the attack will be selected. The attack is just synthesized by doing the iFFT of the database spectral segment chosen because we have attacks for all different pitches. As we are not modifying any characteristics of the original attack, we are not equalizing or transposing the spectrum, the output result after synthesizing will have "sampler-like" sound quality.

When the attack ends, a stationary template segment will be concatenated, accordingly to the note volume MIDI control at the end of the attack.

4.3. Stationary Stage and Interpolation

The stationary part is selected depending on the continuous note volume control at the end of the attack region, the attack and the stationary segment chosen will be concatenated.

During the stationary part, timbre interpolation between stationeries with other dynamics will be performed when the player reduces or increases the volume [6]. It is not enough to decrease or increase the amplitude of the spectrum to achieve a natural sound quality because in wind instruments like the trumpet or the saxophone, when the dynamic changes, the timbre also changes, so the spectrum's envelope has to be equalized from the base spectrum, which is the stationary segment being synthesized, to the target spectrum which is the spectrum of the template with a higher or lower dynamic in case that we are increasing or decreasing the volume.

This equalization has to be applied gradually, assigning weights to the base and target spectrum envelope depending on the continuous note volume, so we obtain a continuous timbre space. The idea is that we will have 4 quantized points in the timbre space, each one assigned to a note volume level, so when the note volume is near one of this points the target spectrum is changed to the desired template and the nearer the value is from the quantized point, the higher weight is given to the target template between 0 and 1 and the lower weight is given to the base template also between 0 and 1, then the

average envelope is calculated following a weighted linear interpolation between base and target. The base spectrum which is always given by the template chosen after the previous attack will be equalized to the average envelope calculated.

In case of synthesizing stationeries that are longer that the recorded ones we will loop in between the same stationary until the note ends or a note transition arrives.

4.4. Release Stage

The release stage is realized with a simple 3 frames fade-out at the end of the stationary just to avoid clicks in the output sound, the effect is quite similar to the sound produced when a saxophone player stops the reed with the tongue.

5. Transition modelling

One of the most important features that makes a spectral synthesizer more attractive than a sampler or other time domain synthesis methods based on wave tables [7] is the possibility to model note transitions.

A transition is detected when the incoming MIDI produces a new note on before the note off of the previous note on. This is realized always in the breath controller Yamaha WX5, because when you change the fingering position, if the breath pressure is bigger than zero, the controller sends the new note on for the new fingering and immediately the note off of the previous fingering. In case of the midi keyboard we can simulate a note transition, pressing the new key before releasing the previous one.
To model these note transitions, from real recordings we extract the pitch and amplitude envelope during the transition and then build a model that will be applied in the synthesis process when concatenating two stationeries at different pitches.

In the next figures we can see two examples of note transitions up (figure 5) and down (figure 6), the pitch and amplitude evolution is quite similar in both cases. From these real recordings analyzed we can build a model where the pitch and amplitude variation is realized during five frames (~30ms) like in most of the examples and the lowest point of the amplitude variation coincides with the middle of the pitch transition. The amplitude variation is for most of the cases around a decrease of 5 to 10 dB depending on the overall amplitude of the notes involved in the transition.

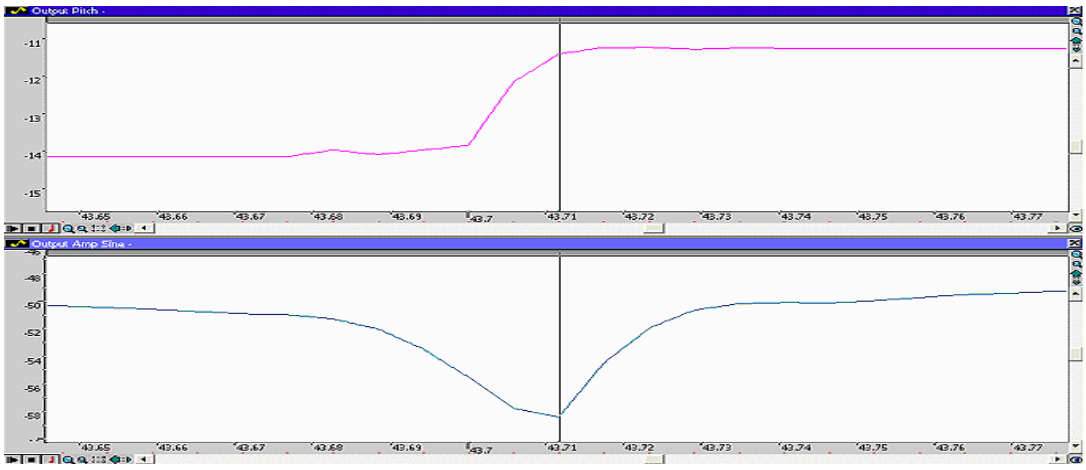

Figure 5: Transition up (3 semitones)

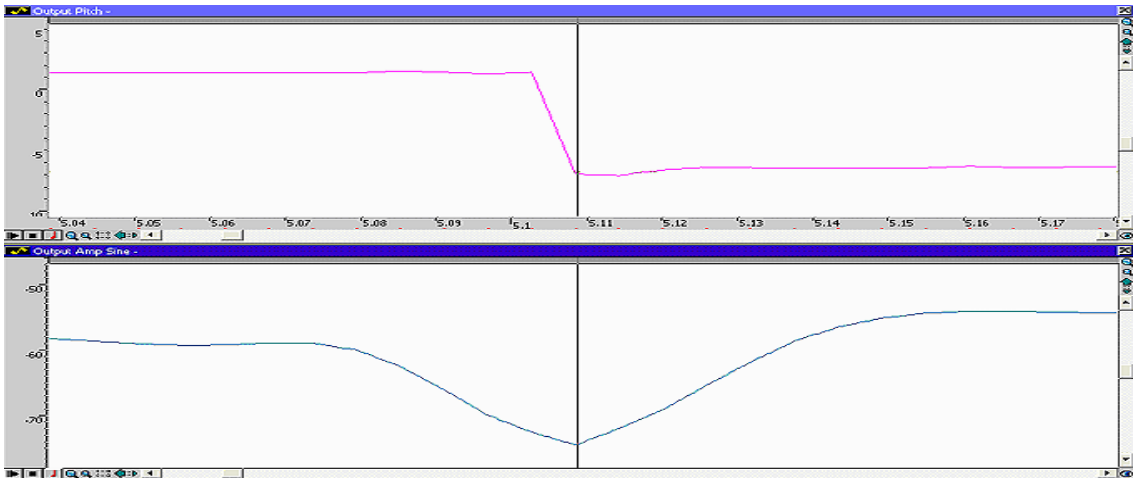

Figure 6: Transition down (1 octave)

In figure 7 we can see the model created, that as a first approach will be the same for all the transitions up or down without taking into account the number of semitones of the interval or the notes location in the register. We should control the number of frames of the transition depending on the dynamic of the notes implied. Usually for piano dynamics, the number of frames for the transition will be bigger and for forte dynamics or when the previous notes are very short, the number of transition frames should be shortened to avoid too long transitions that modulate the whole target stationary producing a vibrato effect.

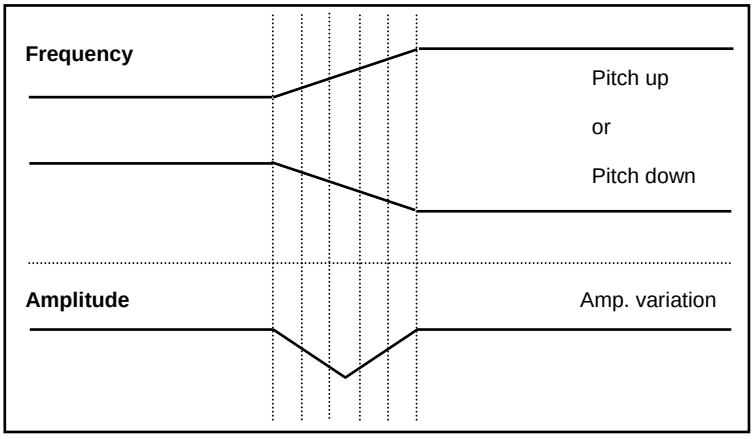

Figure 7: Pitch and amplitude model for transitions (up or down)

6. Conclusions and Future Work

Implementing the new SPP synthesis technique the sound quality achieved is quite better than in the previous implementation that used SMS. The main features that have been improved in this implementation are the timbre interpolation between different dynamics and the memory saving because we need less spectral segments in the database to do the synthesis. The next steps in the project will focus in the implementation of polyphonic synthesis and harmonization, based on to combine various instruments to synthesize a complete wind section that harmonizes a melody being played by a single player.

7. Acknowledgments

The authors of this article would like to thank Joachim Haas who was the main developer in the first implementation of the synthesizer.

8. References

[1] Haas, J., **"SALTO: A Spectral Domain Saxophone Synthesizer"**, *Proceedings of the Mosart Workshop*, Barcelona 2001.

[2] Serra X., Bonada J., et al., **"Integrating Complementary Spectral Models in the Design of a Musical Synthesizer"**, *Proceedings of the ICMC*, Thessaloniki 1997.

[3] Bonada, J., Loscos, A., **"Spectral Peak Processing Documentation"** *Internal MTG paper*, 2001.

[4] Wright M., Chaudhary A., Freed A., Wessel D., **"New Applications of the Sound Description Interchange Format"**, *Proceedings of the ICMC*, 1998.

[5] Rovan J.B., Wanderley M.M., Dubnov S., Depalle P., **"Instrumental Gestural Mapping Strategies as expressive Determinants in Computer Music Performance"**, *AIMI International Workshop*, Genova.

[6] Serra, M.H., Rubine, D., Dannenberg, R., **"Analysis and Synthesis of Tones by Spectral Interpolation",** *Journal of the Audio Engineering Society*, 38(3) (March 1990), pp. 111-128.

[7] Derenyi, I., Dannenberg, R. **"Synthesizing Trumpet Performances"** in *Proceedings of the International Computer Music Conference*. San Francisco (1998).

Sample-based singing voice synthesizer using spectral models and source-filter decomposition

J. Bonada[1], A. Loscos[1], O. Mayor[1], H. Kenmochi[2]
[1] Music Technology Group, Audiovisual Institute, Universitat Pompeu Fabra, Barcelona, Spain
[2]Advanced System Development Center, YAMAHA corporation, Hamamatsu, Japan

ABSTRACT

This paper is a review of the work contained in the insides of a sample-based virtual singing synthesizer. Starting with a narrative of the evolution of the techniques involved in it, the paper focuses mainly on the description of its current components and processes and its most relevant features: from the singer databases creation to the final synthesis concatenation step.

1. INTRODUCTION

The voice generation is typically explained as a source/filter system, in which a voiced/unvoiced excitation is filtered by the vocal tract resonances. The voiced excitation corresponds to the glottal pulses that originate the vocal fold vibrations whether the unvoiced excitation corresponds to the turbulent airflow that arises from the lungs. The voice filter is characterized by a set of resonances called formants that have their origin in the voice organs lengths and shapes (trachea, esophagus, larynx, …). This filter modulates the timbre of the sound by dynamically changing the amplitude, center frequencies and bandwidths of the resonances by moving the voice organs.

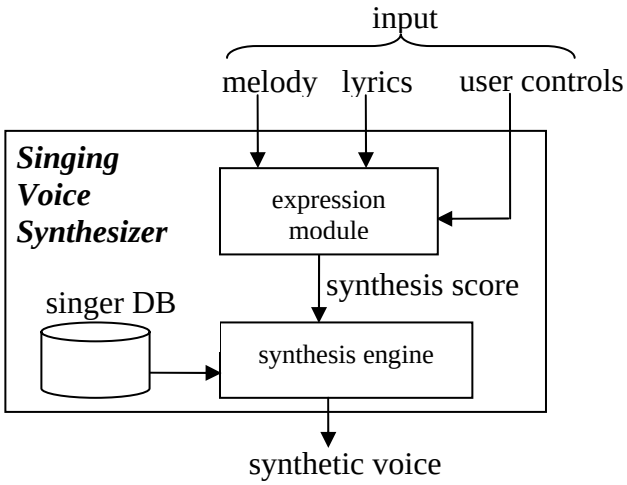

Figure 1: General system diagram

Some of the singing synthesizers developed since the beginnings of such discipline have focused in the source/filter decomposition (physical models based); others, rather than

focusing on how the sound is produced, have focused on the perception of the sound (spectral models based); and others, such as the synthesizer we present in this paper, have tried to combine both models.

The system can be roughly described by a singer database, an input, an expression module and a synthesis engine (see Fig. 1). The input contains the melody and the lyrics of the song plus some expression controls. The expression module converts this input into an internal low-level synthesis score, and the synthesis engine reads this synthesis score, fetches the required samples from the singer database and transforms and concatenates them to obtain the synthetic output signal.

2. VOICE AND SPECTRAL MODELING

Since our system is a sample based synthesizer in which samples of a singer database are transformed and concatenated along time to compose the resulting audio, we have always considered the task of finding the most appropriate and the highest quality transformation techniques a crucial issue.

We initially used SMS [1] as the basic transformation technique with the addition of a time domain delta-based excitation to mimic the singer's voiced excitation [2]. SMS had the advantage of decomposing the voice into harmonics and residual. Both components were independently transformed, so the system yielded a great flexibility. But although the results were quite encouraging in voiced sustained parts, in transitory parts and consonants, especially in voiced fricatives, harmonic and residual components were not perceived as one.

Intending to improve our results, we moved to a spectral technique based on the phase-locked vocoder [3] where the magnitude spectrum is segmented into regions, each of which contains a spectral peak and its surroundings. These regions can be then freely shifted in amplitude and frequency. Regarding the phase spectrum, the relation between harmonics found at the beginning of each glottal period is kept after transformations [4]. On top of this technique we developed a frequency domain voice model that consists of an excitation curve, a set of resonances and a residual envelope. We call it EpR (Excitation plus Resonances) [2]. The excitation curve models the voiced source using an exponential decay function and a low frequency resonance. The vocal tract is modeled using the rest of the resonances and the residual envelope stores the differences between the model and the spectral shape defined by the harmonics (see Fig. 2).

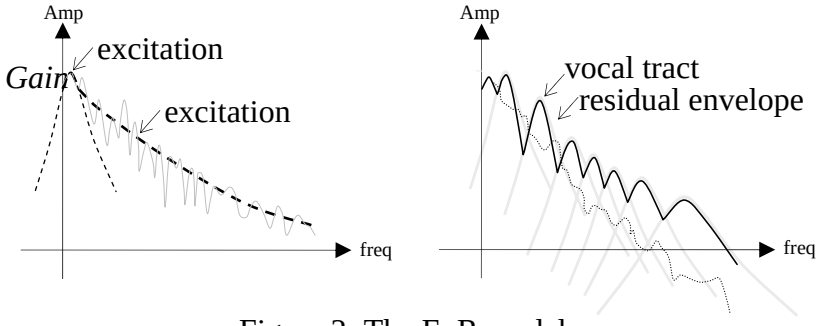

Figure 2: The EpR model

3. SINGER DATABASE

About two hours of dry singer performance recordings are required to build a database. The singer is asked to follow a detailed recording script that covers most possible phonetic contexts and several expression aspects [5]. These recordings are then segmented and analyzed using the spectral analysis algorithms. In order to speed up this process two free software toolkits [6, 7] are used as phonetic aligners between the audio files and the recording scripts. The resulting data fills the phonetic and the expression libraries and is stored in a set of files organized in tree structured folders.

The phonetic database contains timbres, steady-states and articulations. The timbre section stores the voice model (EpR) of different vowels at different pitches and dynamics, the steady-state section contains long sustained vowels at different pitches, and the articulation section contains an organized list of diphonemes samples at different pitches.

The expression database contains note and vibrato templates intended to keep some basic expression aspects of the singer's voice and therefore increase the naturalness of the synthesis. Note templates model singer's attacks, releases and transition behaviors in different musical and intentional contexts. These contexts are described by a set of meaningful labels, like sharp attack, legato transition or sexy release. Each template stores a set of controls (pitch, loudness, EpR excitation curve, breathiness, roughness) obtained from the analysis of the sample, each of which can be later used in synthesis to reproduce the voice excitation changes for each expressive context. Vibrato templates store the singer's excitation behavior for different types of vibrato and tremolos; basically they keep the pitch and the EpR excitation curve. Each template is segmented into attack, body and release parts. The body segment is mirror-looped at synthesis if needed.

4. INPUT SCORE

The input score is an ASCII text file based on the METRIX control language [8] that contains the score of the song. Not only lyrics and notes can be specified, but also high level controls and all the possible music information that the system is capable to interpret. To achieve naturalness in the synthetic voice, the system defines some musically meaningful controls [5]. The idea is to cover the maximum situations that can appear in a real singing performance in order to avoid a lack of expression control that could bring about non-natural results.

The input score contain the so-called note parameters and control parameters. The note parameters refer to a specific note of the score and describe note attributes such as pitch, duration, loudness, lyrics, dynamics, vibrato, attack / release types, roughness, etc., while the control parameters refer to the whole song and describe song attributes such as singer, tempo, etc. Below you can see an example of input score where the lyrics are *fly me*.

```
Score_Info
{
   Tempo: 90
   Meter: 4/4
}
InstrumentInfo { Robert }

begin
  t1    Robert     NoteNumber:   Ab2
                                 Duration: t0.5
                                 Lyrics: "f l aI"
                                 Loudness: 0.6
                                 AttackType: "soft"

  t1.5  Robert     NoteNumber:   G2
                                 Duration: t1
                                 Lyrics: "m I"
                                 Loudness: 0.3
                                 VibratoType: "wet"
                                 VibratoRate: [(0,1)(1,0.6)]
                                 VibratoDepth: [(0,0) (0.5,1)(1,0.7)]
                                 ReleaseType: "long"
end
```

5. BUILDING THE SYNTHESIS SCORE

The expression module generates an internal low-level score (*synthesis score*) out of the input METRIX. This score is structured into several tracks and control envelopes, some of which are shown in Fig. 3.

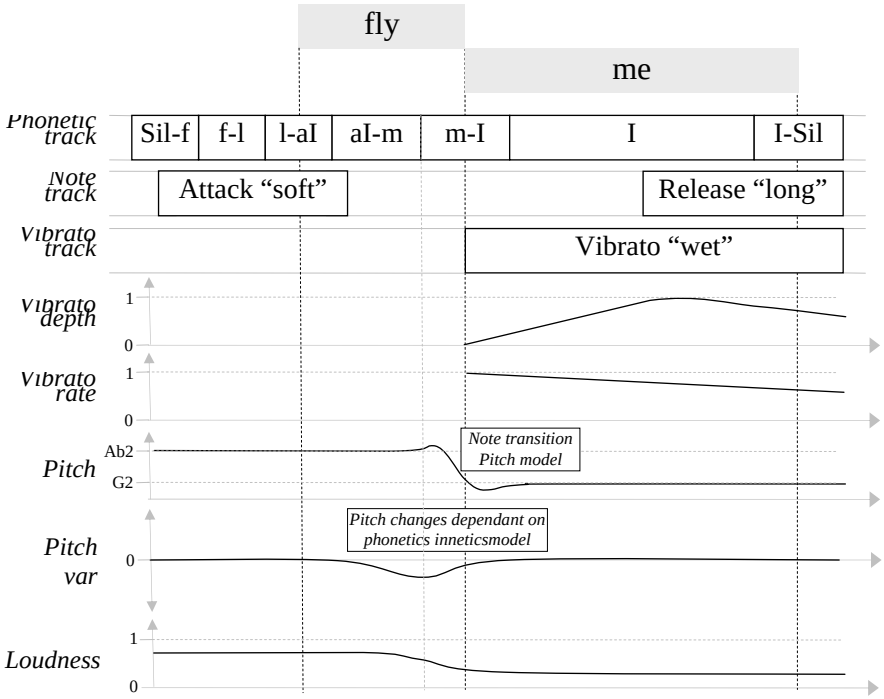

Figure 3: Synthesis score

The phonetic track shows the articulations and steady-states to be fetched from the DB and their corresponding durations, which are calculated trying to make them as close to the original database sample durations as possible. The note and vibrato tracks contain information on the note and vibrato templates that must be applied at synthesis and their corresponding durations. The envelope controls (*vibrato depth and rate, pitch, pitch var, loudness, etc*) express their behavior along the performance with a time-varying function.

In addition to the note and vibrato templates, several models have been created to cover a wide variety of possibilities. However, templates extracted from real recordings are preferable to get a more authentic expressivity, although they may not sound natural when the synthesis context in which they are applied is far from the template context.

The phonetic track is filled out taking into account that the vowel onset should match the begin time of the note. Besides, as already mentioned, taking the original sample duration is preferable since this way we avoid time-scaling transformations, but this is not always possible because all required articulations must fit into the note segment. On the other hand, whenever the added duration of the articulations is less than the target note segment duration, a steady-state is added to fill out what is left.

In the synthesis score there are two envelope controls that specify the output synthesis pitch. The first envelope (*Pitch*) stores the absolute pitch values that come out from the notes specified by the input score. On the other hand *Pitch var* stores relative pitch variations due to changes originated by some phonetic combinations, such as certain voiced consonant - vowel combinations (b-a) in which the pitch decreases during the consonant sound.

In synthesis, the relative values of the *pitch var* envelope and the expression templates are added together to the absolute pitch values. In the case that an attack or release template is specified, the pitch variations of this template are applied when synthesizing to obtain a pitch curve similar to the one in the template. In the case of note transitions, the process is the same but whenever no template is specified, a pitch model is applied that overwrites the absolute pitch track of the score, like shown in Fig. 3, so to avoid pitch discontinuities. This pitch model has to be carefully generated to obtain a natural sounding pitch curve in the output synthesis. A mathematical model has been designed to produce smooth pitch transitions between notes and allow the control of some parameters like duration, shape and synchronization to phonetics and musical rhythm. This synchronization is basically attained by reaching the target pitch at the onset of the vowel of each syllable. In Fig. 4 we can see a more detailed drawing of this pitch model. The distance between *begin pitch* and *max pitch*, as well as between *min pitch* and *end pitch*, depends on the note interval (the bigger the interval, the bigger the distance, but with some limitations for big intervals). On the other hand, the transition curvature depends on both the note interval and the transition duration and its slope is restricted to a maximum value in order to guarantee smooth pitch variations in short transitions.

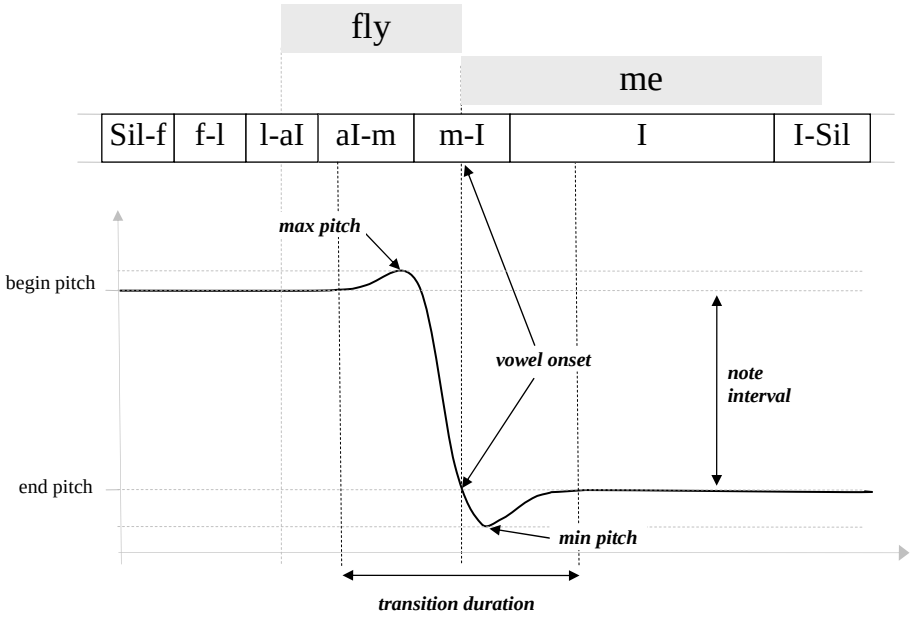

Figure 4: Pitch model for note transitions

6. SYNTHESIS ENGINE

6.1. Sample transformations

The synthesis engine reads the synthesis score and retrieves the required samples and templates from the singer database selecting those units that are closer to the synthesis context (mainly pitch is considered). Once we have retrieved the samples, some transformations [4] are applied to match the synthesis score: transposition, equalization, time-scaling, loudness modification, vibrato and voice excitation based transformations. Finally, the transformed samples are concatenated to compose the resulting synthetic performance**.**

Transposition is applied to match the synthesis score pitch. Therefore, the transposition factor is calculated as the synthesis pitch divided by the sample pitch. This factor is calculated frame by frame. In terms of the spectral technique, harmonic peak's regions are shifted in frequency and harmonic peak's phases are corrected without altering the phase synchronization between harmonics.

Equalization is used to obtain transformations on timbre. When transposing, it is used to keep the original timbre but it can be applied as well to get generic timbre transformations. Equalization is achieved by shifting in amplitude the harmonic peak's regions so to match the desired timbre envelope.

Time-Scaling is applied to samples in order to match their durations with the synthesis score durations. The time-scale ratio is sometimes applied in a non-uniform way so that the synchronization between control parameters, phonetic and note tracks is not altered. For example, the phonetic articulation that contains the vowel onset should not change

the timing of the vowel onset. Besides, in the case of abrupt phonetic changes, these should not be smoothed so not to degrade the intelligibility. The transformation is obtained by repeating or dropping some frames and interpolating them [9].

For loudness modification, database samples are considered to be sung at normal loudness, unless otherwise specified. Thus, sample loudness is changed to match the synthesis score value. The transformation can be achieved by applying an equalization filter obtained from a recorded template where the singer sang a crescendo or a decrescendo. This filter represents the timbre envelope differences between the sample estimated loudness and the target loudness.

For vibrato transformation, the pitch and EpR excitation changes enclosed in the vibrato template are applied to the audio samples. The little nuances of the singer's vibrato are kept even after altering its depth and rate, and the EpR voice model allows the harmonics to follow the resonances as their frequency is modified, thus emulating the real situation.

Besides, some voice excitation based transformations can be applied to improve the naturalness and expressiveness of the synthetic voice, such as roughness, whisper and breathiness. Roughness is obtained by adding sinusoids to the spectrum in a way that glottal periods become irregular. Whisper comes out of equalizing an unvoiced excitation with the timbre envelope. Finally, breathiness is succeeded by adding together whisper and equalization effects and lowering the harmonic's peak adjoining bins.

6.2. Sample concatenation

The last step in the synthesis engine is the concatenation of samples. Once we have transformed the samples, we have to deal with the spectral shape and phase discontinuities that appear when connecting them. With the aim of minimizing such discontinuities, amplitude and phase corrections are spread out along a set of transition frames that surround the boundary [4]. The results are quite smooth and good enough in most cases. Sometimes, however, a gap in brightness can be heard, especially when connecting samples that have been transposed with rather different factors, due to the fact that although there are no harmonic peak's amplitude or phase discontinuities, there do exist harmonic peak's regions amplitude and phase shape gaps. This problem is inherent to only consider harmonic peak's discontinuities when connecting samples, thus our algorithm should be expanded to consider inside region characteristics.

7. CONCLUSION

The system we present is able to generate synthetic performances with quite successful results. However, the more different from the database the synthesizer is asked to sing, the more artificial synthesis gets (it is difficult to make the system sing hip-hop using an opera singer database). Some of this difficulty arises from the fact that the synthesizer has been thought to preserve not only the timbre personality of the singer from which the database is created but also his/her expressivity.

In this sense, work has to be done to improve transformations naturalness, especially when the synthesis context is far from the original context in which the sample that is being transformed was recorded.

Other improvements directions include working on expression dependent timbre transformations and getting into a higher level transformation description in which the system could generate an expressive performance automatically out of the melody, the lyrics, the singer, and an expressive label such as sweet or aggressive.

8. REFERENCES

[1] Serra, X, **"A system for sound analysis-transformation-synthesis based on a deterministic plus stochastic decomposition"** *PhD thesis,* CCRMA, Dept. of Music, Stanford University, 1989.

[2] Bonada, J., Loscos, A., Cano, P., and Serra, X, **"Spectral Approach to the Modeling of the Singing Voice"** *Proceedings of the 111th AES Convention,* New York, USA, 2001.

[3] Laroche, J. and Dolson, M., **"New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects"** *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, New York, 1999.

[4] Bonada, J., Loscos, A., and Kenmochi, K, **"Sample-based singing voice synthesizer by spectral concatenation"** *Proceedings of the Stockholm Music Acoustics Conference SMAC03,* Stockholm, Sweden, 2003.

[5] Bonada, J., Celma, O., Loscos, A., Ortolà, J., and Serra, X., **"Singing Voice Synthesis Combining Excitation plus Resonance and Sinusoidal plus Residual Models"** *Proceedings of the International Computer Music Conference,* Havana, Cuba, 2001.

[6] Akinobu Lee et al, **"Julius - An Open Source Real-Time Large Vocabulary Recognition Engine"** *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 1691-1694, 2001.

[7] Huang, X., Alleva, F., Hon, H., Hwang, M., and Rosenfeld, R., **"The SPHINX-II speech recognition system: an overview"** *Computer Speech and Language*, vol. 7, no. 2, pp. 137-148, 1993.

[8] Amatriain, X, **"METRIX: A Musical Data Definition Language and Data Structure for a Spectral Modeling Based Synthesizer"** *Proceedings of 98th Digital Audio Effects Workshop DAFX98,* Barcelona, Spain, 1998.

[9] Bonada, J., **"Automatic Technique in Frequency Domain for Near-Lossless Time-Scale Modification of Audio"** *Proceedings of the International Computer Music Conference*, Berlin, Germany, 2000.

Musical Expression in a Singing Voice Synthesizer

Oscar Mayor
Music Technology Group, Pompeu Fabra University
omayor@iua.upf.es, http://www.iua.upf.es/mtg

ABSTRACT

This paper talks about different techniques that have been developed in the insides of a spectral singing voice synthesizer in order to apply / modify some expression features of the synthesis. These techniques are based on templates extracted from recordings and mathematical models and pursue the idea of mimicking the voice behaviour of a professional singer. Some of these templates / models have a local influence inside the song since they only modify the attack or release of a note, the type of a note to note transition, the vibrato or the pitch contour in certain phoneme to phoneme transitions. But more general templates / models which apply changes that have influence all over the phrase or even all over the whole song are considered as well. These models are more related with prosody and intonation and are specific to each style of singing like blues, pop, opera, folk, jazz. All these expression features are controlled by the user as a system input.

1. Introduction

The human voice is considered to be the most flexible and fascinating musical instrument and maybe the most difficult to emulate with a computer as well. However, whenever the quality of a vocal synthesizer is assessed, the only parameter in which most listeners mainly concentrate is naturalness.

Previous research have been done in the field of musical expression for musical performance including the quantitative rule system for musical performance in the Department of Speech Music and Hearing of the KTH in Stockholm, which is a set of complex rules which can be used in combinations to give a certain degree of expression to a midi file using a software called Director Musices [1]. Another research done in the Austrian Research Institute for Artificial Intelligence (ÖFAI) in Vienna applies Machine Learning methods to the discovery and analysis of principles of expressive music performance to obtain a deeper understanding of the complex domain of human competence and formulate theories about musical expression [2]. Also as a collaboration between the Artificial Intelligence Research Institute (IIIA) and the Music Technology Group (MTG) in Barcelona, a system capable to apply expression to real non-expressive performances using Case Based Reasoning techniques (CBR) has been developed in the context of saxophone melodies [3].

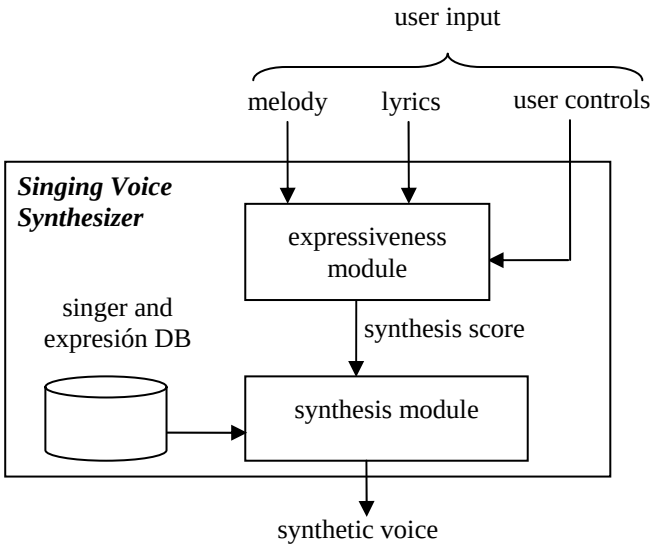

Figure 1: General system diagram

In this paper we present the work done in the framework of a sample-based singing voice synthesizer [4, 7], illustrated in figure 1, in order to give natural expressiveness to the resulting synthesis.

2. User input

The user input is given by the score of the song written using the METRIX control language [5] that contains the melody and lyrics of the song that is going to be synthesized and some expression labels and controls which will be interpreted by the expressiveness module. The idea is to cover the maximum situations that can appear in a real singing performance to avoid a lack of expression control that can result in non natural performances. These expression labels that the user can specify can refer to individual notes (note parameters) or to more general controls that refer to the whole song (control parameters). These controls are shown in figures 3 and 4 and explained in more detail in section 5.

3. The Singer and Expression Database

To build a singer database for the singing voice synthesizer, about two hours of dry singing performance recordings are required. The singer is asked to follow a detailed recording script that covers most possible phonetic contexts and several expression aspects. These recordings are then automatically segmented using two free software toolkits [8, 9] and analyzed using spectral analysis algorithms. The results of the analysis process are then stored in several categories (timbres, steady states and diphoneme articulations) to be used in the synthesis step.

The expression database contains templates intended to keep some basic expression aspects of the singer's voice and therefore increase the naturalness of the synthesis.

Note and vibrato templates are represented by a set of meaningful labels and are used to model singer's attacks, releases, vibratos and transition behaviours in different musical contexts. Each template is represented by a name (soft attack, portamento transition, sexy release, irregular vibrato …) and stores information about some time-varying controls like pitch, loudness, excitation, breathiness and roughness. This information is obtained by analyzing the template sample and these controls will be used later in the synthesis to apply the expressive changes to the non-expressive performance.

4 The Synthesis Module

The synthesis module is able to generate a voice that sings a song taking samples previously analyzed from a database of diphoneme articulations and vowel stationeries at different pitches and transform and concatenate them in the spectral domain to obtain the desired singing phrase. Some transformations applied in the synthesis are controlled by the expression module, so we take non-expressive samples stored in the database and transform it to sound like an expressive template specified by the expressiveness module. In figure 2 a graphical representation of the internal synthesis score is shown, this score is divided in several tracks (phonetic, note, vibrato, pitch, pitch var, loudness) that are created or modified by the expressiveness module from the input score applying the general expression rules and templates specified by the user to achieve naturalness. The creation of this score is explained with more detail in section 5.

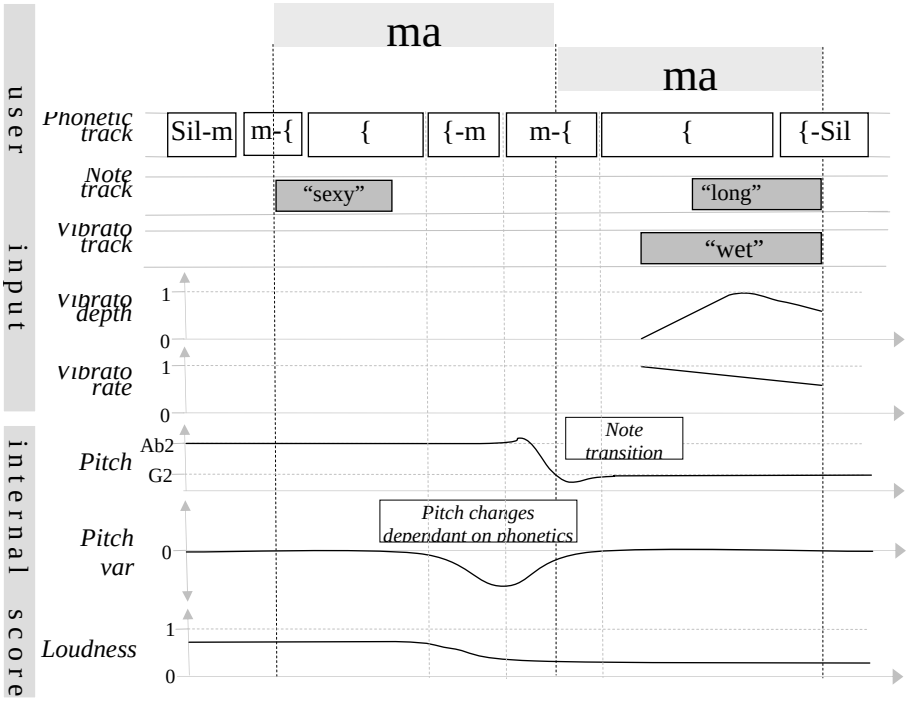

Figure 2: Synthesis Score

The synthesis engine takes as input this internal score and follows three steps to create the output sound: database retrieval of samples, sample transformation and sample concatenation [4]. The idea is to retrieve from the database the samples that are closer to the synthesis context, transformation of this samples include transposition of the

samples to match the output desired pitch, equalization to apply timbre changes and time-scaling to match the durations specified in the synthesis score. At the end of these transformations the samples are concatenated to build the output phrase, we have to deal with differences in spectral shape and phase between samples spreading them out along the sample boundaries to minimize discontinuities.

5. The Expressiveness Module

To achieve naturalness in the synthetic voice, the system defines some musically meaningful controls. The idea is to cover the maximum situations that can appear in a real singing performance, to avoid a lack of expression control that can result in non natural results. In the table below we have a list of specific controls related to individual notes (note parameters, figure 3) and more general controls that refer to the whole song (control parameters, figure 4) [6]. All these controls are specified in the input score using the METRIX control language [5].

These expressive controls are hardly related with mathematical models created to apply transformations in the synthesis to add the desired expression characteristics. As it happens with the controls, these models can affect to the whole song as themes or styles (global expression models) or to concrete parts of the song like the attack of a note, the release or the transitions between notes (local expression models).

Note Parameters	Description
Pitch	Midi number (0-127)
Begin time	Of the note, in milliseconds
Duration	Of the note, in milliseconds
Loudness	Normalizad value [0,1]
Lyrics	Syllable associated to the note
Dynamic envelope	(Time, normalized value)
Pitch envelope	(Time, cents)
Attack type	Normal, sharp, soft, high, …
Attack duration	Normalizad value [0,1]
Release type	Normal, soft, high
Release duration	Normalizad value [0,1]
Transition type	Glissando, marcatto, portamento, …
Vibrato type	Regular, irregular, cobra, …
Vibrato depth	Envelope (time, cents)
Vibrato rate	Envelope (time, Hz)
Opening of vowels	Envelope
Hoarseness	Envelope
Whisper	Envelope
Roughness	Envelope

Figure 3: Note parameters

Control parameters	Description
Singer type	Change singer of the DB
Gender type	Change gender (male/female/child)
Transposition	Transpose the input melody up or down
Style	Blues, Rock, Soul, Jazz, Opera, …
Mood	Sad, Happy, Tender, Aggressive, …

Figure 4: Control parameters

Regarding the degree of control that the user has over these models, we can classify them in another category distinguishing between three kinds of models:

- Models controlled in an automatic way by the system, do not depend on the context and are not related to styles of singing but naturality or better understanding.

- Models controlled in a semi-automatic way depending on some controls specified by the user, related to styles of singing or moods.

- Models controlled directly by the user, like the type of attack, release or vibrato for each note or the type of transition between two notes.

5.1. Mathematical Models vs Templates

In the case of attacks, releases, note transitions and vibratos, several mathematical models have been created to cover a wide variety of possibilities but also some templates extracted from real recordings can be applied to obtain a more real expressivity, although it could sound less natural in some cases that the context in where the template was recorded is very far from the output synthesis context. When we use the word models, we refer to models based in a math function or in a template model, so models is a general word not only referring to mathematical models.

5.1.1 Local expression models

When we talk about local expression models we refer to models that only apply to certain parts of the song or phrase and that are independent of the context where they appear. These models include:

- Attacks
- Releases
- Vibratos
- Transitions Pitch Model
- Pitch Model based on phonetics

To model attacks, releases and vibratos, the system is currently using some recorded templates that are labelled with a description name and, depending on what is specified

in the input score, the synthesis process retrieves the adequate template sample from the database as well as the note sample that is being transformed and the pitch, loudness, excitation, breathiness and roughness of the synthesis is taken from the template and applied to the non-expressive database sample trying to minimize problems due to the different context in where both samples where recorded. The idea is to apply the delta variations of the feature envelopes (pitch, loudness, breathiness, roughness) instead of the absolute values trying to mostly separate the template from its context.

For transitions between notes, when a template is specified, the process is the same as above but in cases that no template is specified, we have developed a mathematical pitch model that overwrites the absolute pitch track of the score to obtain a natural sounding transition avoiding abrupt pitch changes which may result in discontinuities.

Figure 5 shows a graphical example of this model where the curve model is generated between two notes, the first corresponding to the syllable “fly” and the second corresponding to “me” with a lower pitch. The mathematical model created allows to control some parameters like duration, shape and synchronization of phonetics with musical rhythm.

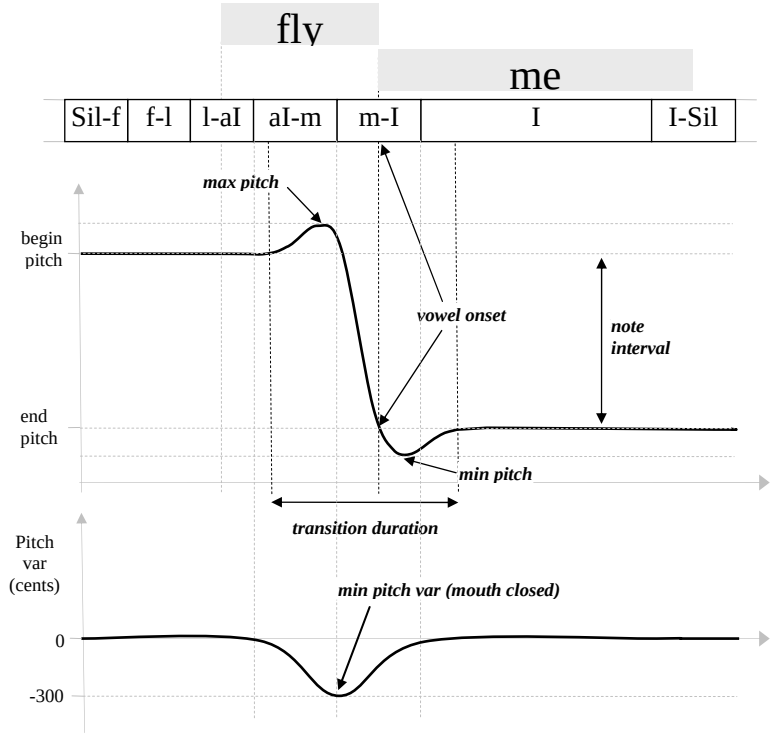

Figure 5: Pitch model for note transitions and pitch variation dependant on phonetics

This synchronization is basically attained by reaching the target pitch at the onset of the vowel of each syllable, and after reaching this onset, the pitch follows a smooth curvature to match again the target pitch at the end of the transition. The behaviour at the begin of the transition is similar, it is clear if we look at the detailed drawing in figure 5. The distance between *begin pitch* and *max pitch*, as well as between *min pitch*

and *end pitch*, depends on the note interval (the bigger the interval, the bigger the distance, but with some limitations for big intervals). The pitch transition slope depends on both the note interval and the transition duration and is restricted to a maximum value, so in short transitions a smooth pitch curve is always guaranteed.

Another local expression model applied to phoneme articulations depends on the combination of phonemes that is being synthesized. A singer closes his mouth when some voiced consonants are pronounced and it causes, depending on the context, to low the pitch during the consonant pronunciation. This situation is accomplished when a vowel follows a voiced consonant ([b]-[V], [m]-[{], [D]-[I]) or when a voiced consonant follows a vowel and is located at the end of a word or followed by another vowel ([e]-[m], [A:]-[g]-[A:], [I]-[m]). In figure 5 (down) we can see the model created to modify the pitch in these cases. The pitch curve that is drawn represents the delta pitch changes (negative values) that will be summed to the absolute pitch values given by the pitch model to create the final output synthesis pitch. These deltas correspond to decrease about 3 semitones (300 cents) at the middle of the voiced consonant.

5.1.2. Global expression models

These models are more general than the previous ones and refer to a bigger context:

- Mood singing (style)
- Naturalness model (loudness and pitch)
- Fast singing and slow singing

Mood singing is a global expression model consisting on specifying a style or mood and to apply some local models in certain parts of the song to make the output to sound with a certain style (blues, opera, rock, pop, soul, jazz) or mood (sad, happy, aggressive, tender). The user will specify using a label the mood and the system will be in charge of applying the needed attack templates, releases, transitions, vibratos and variations of loudness to get the desired expressive synthesis.

The naturalness model consists in applying some pitch and loudness transformations to a global phrase in order to make it sound more natural. The idea is to give to the synthesis a neutral expression that makes the synthesis more understandable and natural. Some changes include to increase the loudness in the strong syllable of each word and decrease it slightly at the begin and the end of words, creating a general pitch model following general rules that professional singers follow when performing a song.

When synthesizing a fast song with very small transitions and very short stationeries, some problems arise; a fast singing model has to be developed to decide which articulations should be stretched more than others in order to get a natural synthesis. The idea is to minimize stretches because in excess causes an annoying synthetic metallic sound. The same problem appears if we synthesize very slow transitions and expand too much the diphoneme articulations.

The synthesis phonetic track is filled out taking into account that the vowel onset should match the begin time of the note. Taking the original sample duration is preferable since

this way we avoid time-scaling transformations, but in the case of fast singing this is not always possible because all required articulations doesn't fit into the note segment and solutions based in what happens in real performances have to be used. When the added duration of the articulations is less than the target note segment duration (slow singing), a steady-state is added to fill out the note segment so the articulations durations of the database are not modified.

6. Conclusions

To give a synthetic singing voice the naturalness that characterizes the human voice is a very difficult task. In this paper we have presented some changes that can be applied to a non-expressive performance to get this naturalness but we have to consider that even with these changes and transformations it is still easy to distinguish between a synthetic and a real performance especially in examples where the singer sings fast. Anyway these methods are a good starting point to follow, to create a fully expressive singing voice synthesis.

7. References

[1] Friberg, A. **"Generative Rules for Music Performance: A Formal Description of a Rule System"** *Computer Music Journal,* 15 (2), 56-71, 1991.

[2] Dixon, S., **"On the Analysis of Musical Expression in Audio Signals".** *Prodeedings of the Conference on Storage and Retrieval for Media Databases, SPIE/IS&T Annual Symposium on Electronic Imaging,* Santa Clara, CA, 2003.

[3] Arcos, J.L., Lopez de Mantaras, R., Serra, X., **"SAXEX: A Case-Based Reasoning system for generating expressive musical performances"** *Proceedings of the International Computer Music Conference,* Thessaloniki, Greece, 1997.

[4] Bonada, J., Loscos, A., and Kenmochi, K., **"Sample-based singing voice synthesizer by spectral concatenation"** *Proceedings of the Stockholm Music Acoustics Conference SMAC03,* Stockholm, Sweden, 2003.

[5] Amatriain, X., **"METRIX: A Musical Data Definition Language and Data Structure for a Spectral Modelling Based Synthesizer"** *Proceedings of 98th Digital Audio Effects Workshop DAFX98,* Barcelona, Spain, 1998.

[6] Bonada, J., Celma, O., Loscos, A., Ortolà, J., and Serra, X., **"Singing Voice Synthesis Combining Excitation plus Resonance and Sinusoidal plus Residual Models"** *Proceedings of the International Computer Music Conference,* Havana, Cuba, 2001.

[7] Bonada, J. Loscos, A. Mayor, O. Kenmochi, H., **"Sample-based singing voice synthesizer using spectral models and source-filter decomposition"** *Proceedings of 3rd International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications,* Firenze, Italy, 2003.

[8] Akinobu, Lee. et al., **"Julius - An Open Source Real-Time Large Vocabulary Recognition Engine"** *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH),* pp. 1691-1694, 2001.

[9] Huang, X., Alleva, F., Hon, H., Hwang, M., and Rosenfeld, R., **"The SPHINX-II speech recognition system: an overview"** *Computer Speech and Language*, vol. 7, no. 2, pp. 137-148, 1993.

Chapter 6

References

[ALS97] Arcos, J.L., Lopez de Mantaras, R., Serra, X., **"SAXEX: A Case-Based Reasoning system for generating expressive musical performances"** *Proceedings of the International Computer Music Conference*, Thessaloniki, Greece, 1997.

[SB97] Serra X., Bonada J., et al., **"Integrating Complementary Spectral Models in the Design of a Musical Synthesizer"**, *Proceedings of the ICMC, Thessaloniki*,1997.

[SS89] Serra, X. Smith, J. **"Spectral Modeling Synthesis"** *Proceedings of International Computer Music Conference, Ohio, USA*, 1989.

[BC00] Batlle, E., Cano, P., **"Automatic Segmentation for Music Classification using Competitive Hidden Markov Models"** *Proceedings International Symposium on Music Information Retrieval*, 2000.

[May01] Mayor, O. **"An adaptative real-time beat tracking system for polyphonic pieces of audio using multiple hypotheses"** *Proceedings of MOSART Workshop on Current Research Directions in Computer Music, Barcelona*, 2001.

[Haas01] Haas, J., **"SALTO: A Spectral Domain Saxophone Synthesizer"** *Proceedings of the Mosart Workshop, Barcelona,* 2001.

[MBL02] Mayor, O. Bonada, J. Loscos, A. "**Real-time spectral synthesis for wind instruments"** *MTG internal (to be published),* 2002.

[MCBL03] Mayor, O. Celma, O. Bonada, J. Loscos, A. "**Musical expression in a singing voice synthesizer"** *MTG internal (to be published),* 2003.

[Fri91] Friberg, A. **"Generative Rules for Music Performance: A Formal Description of a Rule System"** *Computer Music Journal,* 15 (2), 56-71, 1991.

[CBMN02] Cano, P. Batlle, E. Mayer, H. Neuschmied, H. **"Robust Sound Modeling for Song Detection in Broadcasted Audio"** *112th AES Convention, Munich, Germany*, 2002.

[NMB01] Neushmied, H. Mayer, H. Batlle, E. "**Identification of Audio Titles of the Internet"** *International Conference on Web Delivering of Music*, 2001.

[PAK99] F.A.P. Petitcolas, R.J. Anderson, and M.G.Kuhn **"Information Hiding – A Survey"** *Proceedings of the IEEE, special issue on protection of multimedia content, 87(7):1062-1078*, July 1999.

[BTH96] Boney, L. Tewfik, A. Hamdy, K. **"Digital watermaks for audio signals"** *International Conference on Multimedia Computing and Systems, Hiroshima*, June 1996.

[GCGBB02] Gómez, E. Cano, P. Gomes, L. Batlle, E. Bonnet, M. **"Mixed watermarking-Fingerprinting Approach for Integrity Verification of Audio Recordings"** *International Telecommunications Symposium, ITS2002, Natal, Brazil,* 2002.

[CBKH02] Cano, P. Batlle, E. Kalker, T. Haitsma, J. 2002. **"A Review of Algorithms for Audio Fingerprinting"** *Proceedings of 2002 International Workshop on Multimedia Signal Processing. St. Thomas, Virgin Islands,* 2002.

[CBGGB02] Cano, P. Gómez, E. Batlle, E. Gomes, L. Bonnet, M. **"Audio Fingerprinting: Concepts and Applications"** *Proceedings of 2002 International Conference on Fuzzy Systems Knowledge Discovery, Singapore,* 2002.

[GCGB03] Gomes, L. Cano, P. Gómez, E. Bonnet, M. Batlle, E. **"Audio Watermarking and Fingerprinting: For Which Applications?"** *Journal of New Music Research Vol.32 .1,* 2003

[DH89] Desain, P. Honing, H. **”The Quantization of Musical Time: A Connectionist Approach”** *Computer Music Journal, vol 13, no.3, pp56-66,* 1989.

[AD90] Allen, P.E. Dannenberg, R.B. **“Tracking Musical Beats in Real Time”** *Proceedings of the 1990 International Computer Music Conference, pp.140-143,* 1990.

[Ros92] Roshenthal, D. **“Intelligent Rhythm Tracking”** *Proceedings of the 1992 International Computer Music Conference, pp.227-230,* 1992.

[DC00] Dixon, S. Cambouropoulos, E. **”Beat Tracking with Musical Knowledge”** *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000), ed. W.Horn, IOS Press, pp 626-630, Amsterdam,* 2000.

[RGM94] Rosenthal, D. Goto, M. Muraoka, Y. **“Rhythm tracking using multiple hypotheses”***, in ICMC Proceedings,* 1994.

[GM98] Goto, M., Muraoka, Y. **“An audio-based real-time beat tracking system and its applicatios”**, in *ICMC Proceedings*, 1998.

[GM97] Goto, M. Muraoka, Y. **“Real-time rhythm tracking for drumless audio signals – chord change detection for musical decisions”**, *in IJCAI-97 Workshop on computational auditory scene analysis, pp. 135-144,* 1997.

[Sch98] Scheirer, E. **“Tempo and beat analysis of acoustic musical signals”***, in J. Acoust. Soc. Am. 103(1), jan 1998, pp 588-601*, 1998.

[May01] Mayor, O. **”An adaptative real-time beat tracking system for polyphonic pieces of audio using multiple hypotheses”,** *Proceedings of MOSART Workshop on Current Research Directions in Computer Music, Barcelona,* 2001.

[Dix97] Dixon, S. **“Beat Induction and Rhythm Recognition”**, *in Proceedings of the Australian Joint Conference on Artificial Intelligence, pages 311-320,* 1997.

[Des92] Desain P., **“A (de)composable theory of rhythm perception”**, *in Music Perception 9, 439-454,* 1992.

[Dix99] Dixon, S. **“A beat tracking system for audio signals”**, *in Proceedings of the Conference on Mathematical and Computational Methods in Music, Vienna, Austria, pp 101-110,* Dec. 1999.

[FU01] Foote, J. Uchihashi, S. **"The beat spectrum: A new approach to rhythm analysis"**, *in IEEE International Conference on Multimedia & Expo, Tokyo, Japan*, 2001.

[Reid00] Reid, G. "**Introduction to Additive Synthesis",** *Synth Secrets on Sound on Sound Magazine, June*, 2000.

[Rec98] Reck, E. "**Computer Sound Synthesis for the Electronic Musician",** *Focal Press, Oxford, United Kingdom*, 1998

[Chow73] Chowning, John. **"The synthesis of complex audio spectra by means of frequency modulation"**. *Journal of the Audio Engineering Society*, 21(7):526-534., 1973. Reprinted in Curtis Roads and John Strawn, eds. *Foundations of Computer Music*, Cambridge, MA: MIT Press, 1985.

[Serra97] Serra, X. **"Musical Sound Modeling with Sinusoids plus Noise"**. *G. D. Poli, A. Picialli, S. T. Pope, and C. Roads Ed., Musical Signal Processing, p. Swets & Zeitlinger Publishers*, 1997.

[SB98] Serra, X. Bonada, J. **"Sound Transformations Based on the SMS High Level Attributes"** *Proceedings of COST G6 Conference on Digital Audio Effects. Barcelona*, 1998.

[BL01] Bonada, J. Loscos, A. **"Spectral Peak Processing Documentation"** *Internal MTG paper*, 2001.

[LD99] Laroche, J. Dolson, M. **"New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects"**, *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1999.

[Laro03] Laroche, J. **"Frequency-domain techniques for high quality voice modification"** *Procedings of the 6th international conference on digital audio effects DAFX-03, London, UK*, 2003.

[DPZ02] Dutilleux, P., De Poli, G., Zölzer, U., **"Time-segment Processing"** *Udo Zölzer Ed., DAFX: Digital Audio Effects, John Wiley & Sons Publishers*, 2002.

[Bon00] Bonada, J. **"Automatic Technique in Frequency Domain for Near-Lossless Time-Scale Modification of Audio"** *Proceedings of International Computer Music Conference 2000. Berlin, Germany*, 2000.

[Cook95] Cook, P.R. **"Integration of physical modeling for synthesis and animation"**, *in* Proceedings of the International Computer Music Conference, Banff. *1995, pp. 525-528, Computer Music Association*, 1995.

[Smith87] Smith, J.O. "**Music applications of digital waveguides",** *Tech. Rep. STAN-M-39, CCRMA, Music Department, Stanford University*, 1987.

[Smith92] Smith, J.O. "**Physical modeling using digital waveguides"**, *Computer Music Journal, vol. 16, no. 4, pp. 74-91 Special issue: Physical Modeling of Musical Instruments, Part I,* 1992.

[Truax93] Truax, B. **"Time-shifting and transposition of sampled sound with a real-time granulation technique"** *Proceedings of the 1993 International Computer Music Conference, San Francisco: International Computer Music Association, p. 82-85,* 1993.

[Clarke96] Clarke, M. **"Composing at the Intersection of Time and Frequency"** *Organised sound: An international journal of music technology,* vol. 1, no. 2, Aug 1996 p. 107-117, 1996.

[Dix03] Dixon, S. **"On the Analysis of Musical Expression in Audio Signals".** *Prodeedings of the Conference on Storage and Retrieval for Media Databases, SPIE/IS&T Annual Symposium on Electronic Imaging,* Santa Clara, CA, 2003.

[BF00] Bresin, R., Friberg, A., **"Emotional Coloring of Computer-Controlled Music Performances"** *Computer Music Journal, 24:4, pp. 44-63, MIT,* Winter 2000.

[CKMB01] Cano, P. Kaltenbrunner, M. Mayor, O. Batlle, E. "**Statistical Significance in Song-Spotting in Audio"** *Proceedings of International Symposium on Music Information Retrieval. Bloomington, Indiana (USA),* 2001.

[Gusf97] Gusfield, D. **"Algorithms on Strings, Trees and Sequences"** *Cambridge University Press, 1997. Smith, T.F. and Waterman, M.S., Identification of common molecular subsequences, Journal of Molecular Biology. 195-197,* 1981.

[NDR97] Nicholas, H.B. Deerfield, D.W. Ropelewski, A.J. **"A Tutorial on Searching Sequences Databases and Sequence Scoring Methods"**, 1997.

[PL88] Pearson, W.R. Lipman, D.J. **"Improved tools for Biological Sequence Comparison"** *Proc. Natl. Acad. Sci. 85: 2444-2448*, 1988.

[KA90] Karlin, S. Altschul, S.F. **"Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes"** *Proc. Natl. Acad. Sci. USA 87, 2264-2268,* 1990.

[DD98] Derenyi, I. Dannenberg, R. **"Synthesizing Trumpet Performances"** *in Proceedings of the International Computer Music Conference. San Francisco,* 1998.

[SRD90] Serra, M.H. Rubine, D. Dannenberg, R. **“Analysis and Synthesis of Tones by Spectral Interpolation”** *Journal of the Audio Engineering Society, 38(3), pp. 111-128.*, March 1990.

[BLMK03] Bonada, J. Loscos, A. Mayor, O. Kenmochi, H. **”Sample-based singing voice synthesizer using spectral models and source-filter decomposition”** *Proceedings of 3rd International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications, Firenze, Italy,* 2003.

[Poli03] De Poli, G. **“Expressiveness in Music Performance: Analysis and Modeling”** *Proceedings of the Stockholm Music Acoustics Conference (SMAC 03), Stockholm, Sweden,* 2003.

[Cano98] Cano, P. **“Fundamental Frequency Estimation in the SMS analysis”** *Proceedings of COST G6 Conference on Digital Audio Effects, Barcelona,* 1998.

[BLK03] Bonada, J. Loscos, A. Kenmochi, K. **"Sample-based singing voice synthesizer by spectral concatenation"***Proceedings of the Stockholm Music Acoustics Conference SMAC03, Stockholm, Sweden,* 2003.

[Sun87] Sundberg, J. **“The Science of the Singing Voice”** *Northern Illinois University Press,* 1987.

[Cana97] Canaza, S. et al. **“Sonological Analysis of Clarinet Expressivity”.** *In M. Leman, ed. Music, Gestalt, and Computing: Studies in Cognotive and Systematic Musicology.Springer Verlag, 431-440, Berlin,* 1997.

[BF98] Battel, G.U. Fimbianti. **“Adding Expressiveness to Automatic Musical Performance.”** *In A. Argentini and C. Mirolo, eds. Proceedings of the XII Colloqium on Musical Informatics. Udine: AIMI, 71-74,* 1998.

[PRV98] de Poli, G. Rodà, A. Vidolin, A. **”A Model of Dynamic Profile Variation, Depending on Expressive intention, in Piano Performance of Classical Music.”** *In A. Argentini and C. Mirolo, eds. Proceedings of the XII Colloqium on Musical Informatics. Udine: AIMI, 79-82,* 1998.

[OC98] Orio, N. Canazza, S. **“How Are Expressive Deviations Related to Musical Instruments? Analysis of Tenor Sax and Piano Performances of ‘How High the Moon’ Theme.”** *In A. Argentini and C. Mirolo, eds. Proceedings of the XII Colloquium on Musical Informatics. Udine: AIMI, 75-78,* 1998.