MORPHING TECHNIQUES FOR ENHANCED SCAT SINGING

Jordi Janer

Technology Department
Universitat Pompeu Fabra, Barcelona, Spain

jordi.janer@iua.upf.es

Alex Loscos

Technology Department
Universitat Pompeu Fabra, Barcelona, Spain
alex.loscos@iua.upf.es

ABSTRACT

In jazz, scat singing is a phonetic improvisation that imitates instrumental sounds. In this paper, we propose a system that aims to transform singing voice into real instrument sounds, extending the possibilities for scat singers. Analysis algorithms in the spectral domain extract voice parameters, which drive the resulting instrument sound. A small database contains real instrument samples that have been spectrally analyzed offline. Two different prototypes are introduced, producing sounds of a trumpet and a bass guitar respectively.

1. INTRODUCTION

1.1. History of scat singing

Scat is defined as a vocal improvisation using phonetic sounds similar to the instrumental sounds of jazz. Although, scat is believed to have its origins in African American ring shout ceremonies, it was first popularized by Louis Armstrong in a 1926 recording, when he dropped the lyrics and spontaneously substituted *scat* for the words. This particular singing style was adopted by other great jazz singers such as Ella Fitzgerald or Sarah Vaughan. Today, *scat* maintains its relevance with performers like Bobby McFerrin, who exploits his vocal register producing a wide range of sounds.

1.2. Transforming the voice – Voice driven synthesis

In the context of spectral audio processing, *Morphing* [9] has appeared as a technique for generating new and never heard sounds. Actually, the concept of *Morphing* comes from the field of video processing, where an image is transformed sequentially into another. In audio processing, attempts for synthesizing sounds, whose acoustic characteristics lie between two real instruments sounds, have been carried out. However, in the presented approach, we make use of *Morphing* techniques for transforming the input singing voice into a musical instrument sound. In the literature, this technique is also referred as *Cross-Synthesis*.
A particularity of such a system is that it can be understood by means of two different perspectives: as a voice transformation effect, or as a voice-driven synthesizer. This issue can motivate a more thorough discussion that is beyond the scope of this article.

2. VOICE AS A CONTROLLER

2.1. Preliminary issues

Digital music synthesis permits a wide range of sounds while enabling a high degree of control over the sound generation process. It overcomes physical (in acoustic instruments) or electronic (analog synthesizer) limitations in terms of control. This flexibility of control has led to the emergence of numerous musical controllers. In this work, we use the singing voice as a muscial controller, making use of feature extraction algorithms that parametrize the input voice. The motivation for exploring the voice as a controller are various. It is common in a rehearsal situation that musicians employ the singing voice for expressing a musical concept or comment to another musician. Hence, it seems a straighforward way for transmitting musical information, while maintaining a high dimensionality.

On the other hand, in the past decades many controllers appeared with the aim to provide new forms of musical expression beyond the traditional instruments, and in particular, linked to computer music. One of these systems are gestural controllers, which rely on the idea that body gestures are an appropiate form of controlling the generation of music. However, we argue that the intened intimacy is not fully accomplished with such systems. Actually, the use of gestural controllers are often bound to experimental dance performances. Also, we shall not overlook other developements that use the voice as input device in other contexts. One application is video games, where a microphone replaces the traditional joystick. In [10], example games are presented where the pitch controls the player position.

The hypothesis in which this system stands is that voice can be successful in specifying main characteristics of a melody but also the expressivity nuances (such as attack sharpness / smoothness or dynamic envelope) of an instrument performance.

2.2. Voice-to-MIDI systems

Let us consider Voice to MIDI systems as a possible path for enhanced singing. There already exist several hardware and software commercial voice-to-midi converters. Among diverse softwares that offer real-time conversion we can find: MidiVoicer (www.e-xpressor.com/m_voicer.html), and Digital Ear (www.digital-ear.com/digital-ear/info.htm). Some hardware solu-

tions are: Vocal to MIDI (www.geocities.com/vocal2midi/), and MidiVox (www.healingmusic.net).

Figure 1: *MidiVox general view*

Some of these converters might be useful and give reasonable results in some cases but they generally lack of robustness. The problem comes from the singing voice real-time note onset detection. Such critical process is in charge of deciding at each analysis frame time if the current voice data belongs to a new note or if it has to be considered part of the note that was already being performed. This decision has to be taken with no frame delay and with no prior knowledge on the final melody (not even key or/and scale). The considerable complexity of this problem makes it nearly impossible to avoid the converter outcoming false notes.

Furthermore, when writing a song with a sequencer, either the composer records the score using a MIDI keyboard; either writes it by clicking the mouse. Clearly none of these solutions give intuitive and meaningful control over the final synthesis result and they usually require of a tedious test and trial tuning. This makes us think voice analysis and mapping to MIDI require from an excellence none of the available systems are offering.

3. MORPHING SYSTEM OVERVIEW

The limitations of MIDI in terms of musical representation led us to consider another method for controlling the sound synthesis. We propose to make a direct link of voice analysis parameters to synthesis parameters, without the restrictions of any protocol. This will certainly limit the integration with other systems, which is currently beyond our goals.
Similar to [5], here the processes can be decomposed in non real time processes, which are the ones that take place as a prelude, before the application runs, and the processes that take place in real time, which are the ones that occur while the user is performing.
The analysis used for both the instrument samples (trumpet and bass guitar) and the voice signal captured by the microphone is based is frame-based and uses spectral domain techniques that stand on the rigid phase-locked vocoder [6].

3.1. Non-real time processes: instrument database creation

The non real time processes focus on the wind instrument database creation. That is, on recording real performances, editing and cutting them into notes, analyzing and labeling them, and storing them as an instrument database. The samples are analyzed off-line in the spectral domain, and stored in the database in form of binary files containing spectral data (complex spectrum, harmonic peaks, estimated pitch, etc.).

3.2. Real time processes: Voice Analysis, Transformations and Synthesis

The real-time processes start with a frame based spectral analysis of the input voice signal out of which the system extracts a set of voice features. This voice feature vector is first used to decide which sample to fetch from the database, and later, as input for the cross-synthesis between the instrument sample and the voice signal.
The criterion that decides the trumpet sample to use at each voice frame is: take the nearest sample in pitch. From that sample, the trumpet frame is chosen sequentially taking into account loops. The frame is transformed so to fit the user's energy and tuning note specification, for which energy correction and transposition with spectral shape preservation is applied with similar techniques to those described in [2].
Finally, the synthesis is in charge of concatenating the synthesis frames by inverse frequency transformation and the necessary window overlap-add related processes.

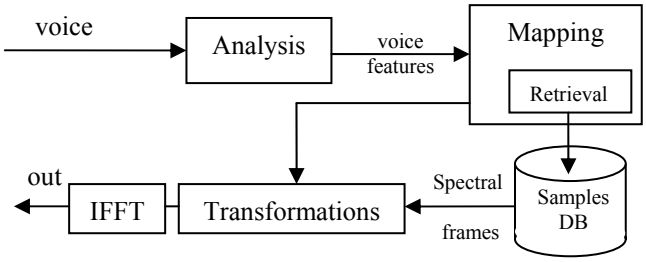

Figure 2: *Block diagram of the morphing processs.*

4. PROTOTYPES

4.1. Trumpet

The proposal is to use the pitch envelope obtained in the fundamental frequency analysis to fill the key number and its associated modulation along the note, the pitch bend; and to use the Excitation plus Residual (EpR) Voice Model excitation parameters [3] to fill the key velocity and its associated modulation, the aftertouch. Of course these voice attributes don't fulfill vocal expressivity space; however, when adequately mapped will allow a useful basic control.

In the current implementation, the database is made out of three trumpet notes at A3, A#5, and C5. For each sample the database contains a binary file in which the necessary data resulting from the analysis is stored. These data is pitch and energy envelope.
The mapping between dynamic related attributes and key velocity and aftertouch would be applied just the same. In this case, dynamic envelope is calculated out of the three EpR excitation parameters, that is, excitation gain, excitation slope, and excitation slope (see figure 3).
Obviously though, this process has to run on the fly and this means once a frame has been detected as the onset of a note, the converter takes the current pitch and dynamic values (possibly

averaged along some short past history) as the mean over the note. Thus, all following frames that are considered part of that note define aftertouch and pitch bend messages from the difference between its current values and the onset frame values.

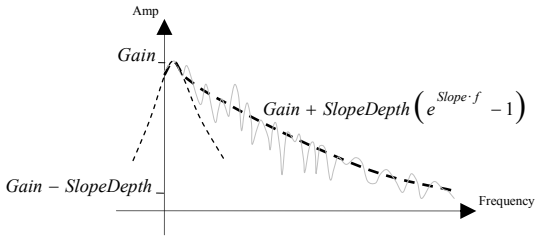

Figure 3: *EpR model voice excitation representation*

The cross-synthesis between wind instrument samples and voice utterances is a shortcut technique that avoids intermediate MIDI conversions. We define cross synthesis as the technique by which elements of one or more sounds combine to create a new one with hybrid properties. Taking profit of this technique capabilities, we can extend the voice control further than pitch, dynamics and their associated modulations and set off continuous control over, for example, the sharpness of the attack or the instrument timbre modulations.
In this cross-synthesis, wind instrument will take synthesis pitch and dynamics directly from the pitch and dynamics attributes of the input voice. More complex controls such as the ones previously mentioned can achieved by using, for example, voice excitation gain first derivative or spectral tilt voice attribute to drive the wind synthesis.

The prototype incorporates some additional transformations. One of them is a one octave up / down transposition. When turned on, it is applied to the voice analysis data before using it to fetch the trumpet frame. Another additional transformation is pitch quantification, implemented as in [7] with optional specifications of key and chord.
There is also an 'Extract Unvoiced' switch which can mute the unvoiced parts of the input. This allows the musician to use unvoiced allophones in the performance.
Finally, the prototype includes a morph feature by which the resulting synthesis timbre can defined as a balanced interpolation of both voice and trumpet timbre. Timbres are defined by means of the harmonic peaks magnitude data. The morph slider in the interface defines the morph interpolation value.

The trumpet prototype can be enhanced with some short term improvements. The current instrument database calls for many more samples, ideally several different dynamics (from pianissimo to fortissimo) for each of the notes the instrument can perform. The more samples the database contains, the less extreme transformations we will have to apply. Also, the only voice attributes used as cross synthesis controls are energy and pitch. Energy should be replaced by EpR voice excitation parameters and more complex controls such as the ones mentioned before could be achieved by using voice excitation gain first derivative or spectral tilt voice attributes. In synthesis, whenever the instrument sample is shorter than the note performed by the user, the system keeps looping it from its attack to its release. It is a must to have a 'loop region' attribute computed and stored in the database for each of the samples.

4.2. Bass guitar

Since the introduction of the jazz music, the bass, originally a double bass, has become an indispensable instrument of any rhythm section in popular music. Jointly with the drums, it provides the basis upon the other instruments play. The role of the bass continued evolving in new musical styles such as *rock* or *funk*, after the introduction of the electric bass by the Fender company in the 50's. Furthermore, other playing styles appeared, such as the "slap'n'pop" that was invented by Larry Graham in the 60's and often found in funk productions. The sound of a plucked bass guitar consists of impulsive notes that decay exponentially in time. Due to the long note decay time, bass players usually damp the vibrating string with the finger before plucking a new note. Usually, both the double bass and the electric bass have four strings tuned at E1 (41.3*Hz*), A1, D2 and G2. Therefore, the fundamental frequency remains below other instruments range, attaining a good timbre balance when playing together.

Essentially, vocal gestures can be of three different kind in the voice production mechanism, depending on whether its origin is either in the breathing system, in the glottis, or in the vocal tract. In this approach, the chosen features are classified based on control aspects: Excitation, Vocal Tract, Voice Quality and Context.

A more detailed description of the feature extraction algorithms, which derive from spectral analysis, appears in [8]. *Excitation* descriptors are probably the most elemental and intuitive for the user, since they are related to the instantaneous sung energy and fundamental frequency. In a further step, we find the voice color or voice timbre. For voiced sounds, the timbre is associated to a particular vowel. We assume that a vowel can be approximately determined by its two first formant frequencies. A very simple algorithm based on spectral centroid, *Dual-Band Centroid Estimation* [8], will estimate these frequencies approximately. In addition to the introduced features, we argue that the singing voice comprises other characteristics that might be also controllable by the singer, and thus useful in this kind of systems. Algorithms for estimating timbrical aspects (*Voice Quality*) such as *roughness* and *breathiness* were developed. Although these algorithms need further research, it is a good start-point for controlling other aspects of the transformed sound.

Finally, we include the *Context* features, which give us information of the note's state. We included the *Attack unvoiceness* descriptor. Assuming that a note consists of a pitched sound, this descriptor attempts to determine whether the attack of the note was unvoiced or not. The formula derives from the accumulative zero-crossing rate of the signal weighted with the energy, which is evaluated just before a note onset. The initial motivation for this descriptor is to use it for determining the harshness of the synthesized note's attack. In our case of a bass guitar, it might be related to the differences between a soft fingered and a sharp slap electric bass.

In our case, from the *Energy* feature, we defined a note onset signal that triggers a note depending on the energy's derivative. The *Pitch* feature is transposed one octave lower and passed as continuous parameter, thus allowing pitch bend actions. A quantization function for simulating the influence of the frets is foreseen,

but not yet integrated in the current implementation. In addition, the two formant frequencies, which are related to the timbre, are also passed though as continuous parameters to the synthesis technique dependant layer. In the following sections, we address the mapping particularities of each synthesis technique.

In a further step, all tracks have been labeled by hand according to its characteristics. Currently, only three features were annotated: *Pitch*, *Dynamics* and *Attack Type*. The pitch values are specified in *Hz*, *Dynamics* and *Attack Type* range is [*0..1*]. In the case of Dynamics, *0* corresponds to a *pp* sound, and *1* to a *ff*. The attack type is a novel concept that we defined, and it is instrument dependant. Concerning bass sounds, we decided to classify two types of sounds depending on the plucking technique pick-plucked or fingered, whose sounds are primarily related to the attack.

For our implementation, we set up a very small database consisting of 12 template tracks (containing one note sample). A retrieval method calculates the minimum Euclidean distance between the Input Vector (*Pitch, Loudness and Attack Unvoiceness*) and the database elements. This outputs an ID corresponding to the selected track. Then, in a further step, we start reading the spectral frame of the track. Since we are dealing with a very small database, few combinations of loudness and pitches are available. Currently, the transformation process is reduced to a transposition.

Another factor that must be taken into account is the timing. The pre-analyzed template tracks have a certain duration. For each track, this corresponds to a certain number of spectral frames. In our system, though, the performer's voice controls the synthesized sound. Hence, is the input voice that decides the output duration. We analyze the track by hand and set two loop points (*sustained* and *release*) within a steady region of the original sound. When a note is triggered, we start reading the template track frame by frame. When the point sustained is reached, we repeat this frame continuously keeping the phase coherence in the generated spectrum. When the performer releases the current sung note, the last frames are played until the note's end.

5. CONCLUSIONS

With this work, we aimed to extend the possibilites of musical performances by using the voice as a controller. In particular, the prototypes presented target jazz singers eager to experiment news ways of scat singing.
Although the current prototypes are still in an early stage, we believe that using the voice as a musical controller brings an intersting perspective to the music generation. Further improvements will focus on the extraction of higher-level vocal features, and on mapping strategies that achieve a more intimate control of the synthesized sound.

6. ACKNOWLEDGEMENTS

This research has been partially funded by the EU-FP6-IST-507913 project SemanticHIFI.

7. REFERENCES

[1] Flanagan, J.L. and Golden, R.M., “Phase Vocoder”, Bell Systems Technology Journal, vol 45, 1966

[2] Laroche, J. and 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

[3] Bonada, J. and Loscos, A., “Sample-based singing voice synthesizer by spectral concatenation”, Proceedings of Stockholm Music Acoustics Conference 2003, Stockholm, Sweden, 2003.

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

[5] Cano P., Loscos A., Bonada J., de Boer M., Serra X. “Voice Morphing System for Impersonating in Karaoke Applications”, *Proceedings of International Computer Music Conference*, Berlin, Germany, 2000.

[6] Puckette M. S. “Phase-locked vocoder”, Proceedings of IEEE Conference on Applications of Signal Processing to Audio and Acoustics, Mohonk, USA, 1995.

[7] Zolzer U. DAFX – Digital Audio Effects. Wiley, John & Sons, March, 2002.

[8] Janer, J. “Feature Extraction for Voice-driven Synthesis”, Proceedings of the 118th AES Convention, Barcelona, 2005.

[9] Cook, P.R., “Toward the Perfect Audio Morph? Singing Voice Synthesis and Processing”, Proceedings of the 1st. International Conference on Digital Audio Effects (DAFX), Barcelona, 1998.

[10] Hämäläinen P. et al., “Musical Computer Games Played by Singing”, Proceedings of the 7th. International Conference on Digital Audio Effects (DAFX), Naples, 2004.