Journal of New Music Research
2003, Vol. 32, No. 1, pp. 95–114

0929-8215/03/3201-095$16.00

Content-based Transformations

Xavier Amatriain[1], Jordi Bonada[1], Àlex Loscos[1], Josep Lluís Arcos[2] and Vincent Verfaille[3]

[1]MTG-IUA, Universitat Pompeu Fabra, Barcelona, Spain; [2]Artificial Intelligence Research Institute (IIIA), CSIC, Bellaterra, Spain and [3]CNRS-LMA, Marseille, France

Abstract

Content processing is a vast and growing field that integrates different approaches borrowed from the signal processing, information retrieval and machine learning disciplines. In this article we deal with a particular type of content processing: the so-called content-based transformations. We will not focus on any particular application but rather try to give an overview of different techniques and conceptual implications. We first describe the transformation process itself, including the main model schemes that are commonly used, which lead to the establishment of the formal basis for a definition of content-based transformations. Then we take a quick look at a general spectral based analysis/synthesis approach to process audio signals and how to extract features that can be used in the content-based transformation context. Using this analysis/synthesis approach we give some examples on how content-based transformations can be applied to modify the basic perceptual axis of a sound and how we can even combine different basic effects in order to perform more meaningful transformations. We finish by going a step further in the abstraction ladder and present transformations that are related to musical (and thus symbolic) properties rather than to those of the sound or the signal itself.

1. Introduction content and transformations

The term "content processing" has already been around for a few years (Karjalainen, 1999; Chiariglione, 2000; Camurri, 1999) but its meaning is still unclear and a matter of controversy. When we talk about content analysis, content browsing, content indexing, content processing or content transformation we are usually addressing the higher-level information that a signal produced by an audiovisual source carries within.

Even though the previous pseudo-definition is conservative in its scope, it already includes a crucial and sometimes polemical term: higher-level. It is true that this label assumes that it is being compared to something else, and this something else is usually the signal processing level. Even so, what about semantic features that can (more or less directly) be extracted from the actual signal? Should we consider *pitch* as a higher-level feature as opposite to its signal-processing counterpart, *fundamental frequency*? How can we distinguish between the abstraction level implied by some perceptual feature like *loudness* and some other with more semantic load such as *genre*?

We will use the word "content" for any piece of information related to the audio source that is in any way meaningful (that it carries semantic information) to the targeted user. Thus, the description of that content can be thought of as a content hierarchy with different levels of abstraction, any of them potentially useful for some users. In that sense, think of how different would a content description of a song be if the targeted user was a naive listener or an expert musicologist. Even a low-level descriptor such as the spectral envelope of a signal can be thought of as a particular level of content description targeted for the signal processing engineer.

On the other hand, when we use the term *transformation*, we use it in a different way from how we would use the term *effect*. When we talk about an effect, we are focusing on the result of changing the sound in a particular way. However, when talking about a transformation, the strength is put on the change that a particular sound experiments, rather than on the result. Thus, not every sound can undergo a certain transformation, yet an effect can be applied on any source regardless its properties. That is the reason why we use the word *transformation* when addressing the "content" level.

Accepted: 9 May, 2002

Correspondence: Xavier Amatriain, Institut de l'Audiovisual, Universitat Pompeu Fabra, Pg. Circumval.lació 8, 08003 Barcelona, Spain. Tel.: +34 93 542 2199, E-mail: xamat@iua.upf.es

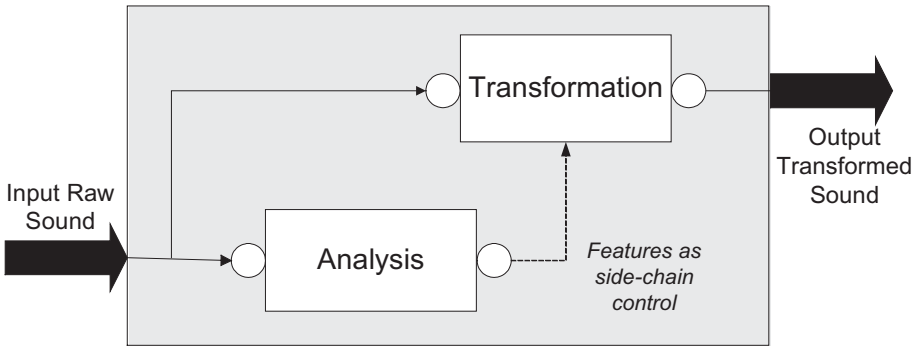

Fig. 1. Content transformation scenario. Analysis output is used as a control signal.

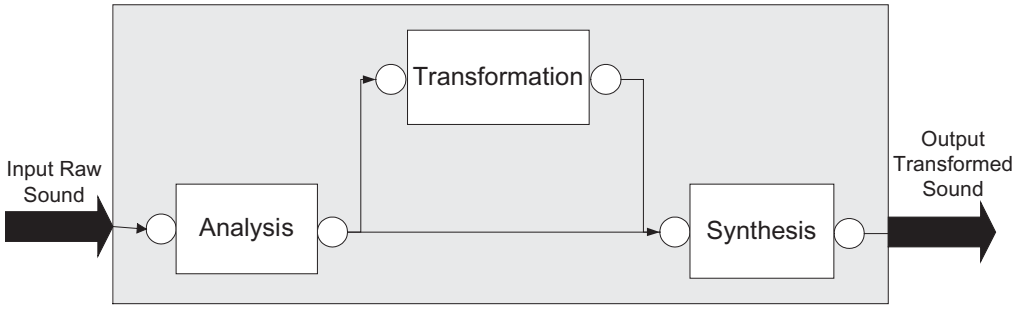

Fig. 2. Transformation process based on an analysis-synthesis model.

Throughout the previous discussion, it has been assumed that, in order to be able to apply some kind of content transformation, the signal must undergo a previous analysis step. The goal of this step is to compute features that will then be relevant in the transformation step.

The first possible scenario is the one represented in Figure 1. The output of the analysis is used as a control to the transformation block. The transformation is thus applied to the original sound directly. Note that, in this case, the user input is not used in the transformation chain so the scheme could be labeled as "unsupervised." The parameters of the transformation are dynamically adapted to the characteristics of the input signal.

A very basic example of this kind of signal processor would be an automatic gain control. Such a system can reduce or increase its gain depending on the relation between the input signal and a given threshold. When the signal exceeds that particular threshold the gain is reduced and the transformation is said to be a "compressor" (or a limiter if the slope is smaller than 1/10). On the other hand, if the signal is below the threshold, the gain is increased and the transformation is known as an "expander." One may argue that this sort of signal analysis is too "low-leveled" to be included in the category of content-based transformation but we refer again to the definition of "content" previously introduced. The content description of the signal is being reduced to just a very simple feature: its level. Anyhow, it is clear that the transformation depends on the analysis of that particular feature applied to the incoming signal. But the scheme here outlined can be applied to any kind of transformation and using any feature of the sound as a control signal. That is the idea behind Adaptive Digital Audio Effects (A-DAFx) (Verfaille & Arfib, 2001).

Most of the transformations implemented in the time-domain can fit quite well into any of the variations of the model presented up until now. The implementation of the processing algorithms is quite straightforward and based on a sample-by-sample process. Examples of transformations that can be effectively implemented using these techniques include those related to effects like delays, chorus, reverbs, dynamic processors, etc. More complex transformations such as those based on pitch-shifting or time-stretching may also be accomplished in the time domain using techniques like PSOLA (Pitch-synchronous overlap and add) (Dutilleux et al., 2002).

But sometimes the information that can be immediately gathered from the signal and its time-domain representation may not be enough in order to design a particular transformation. In a content transformation we want to analyze the signal and extract meaningful features from this analysis step and the time-segment processing is not well-suited for this sort of schemes.

In such situations, the analysis step must yield more than just a set of features to be used as control signals. Thus, in order to achieve "more interesting" transformations we need to find a model for the signal in such a way that this intermediate representation is more suitable for applying some particular processes. Thus, the signal is analyzed, trans-

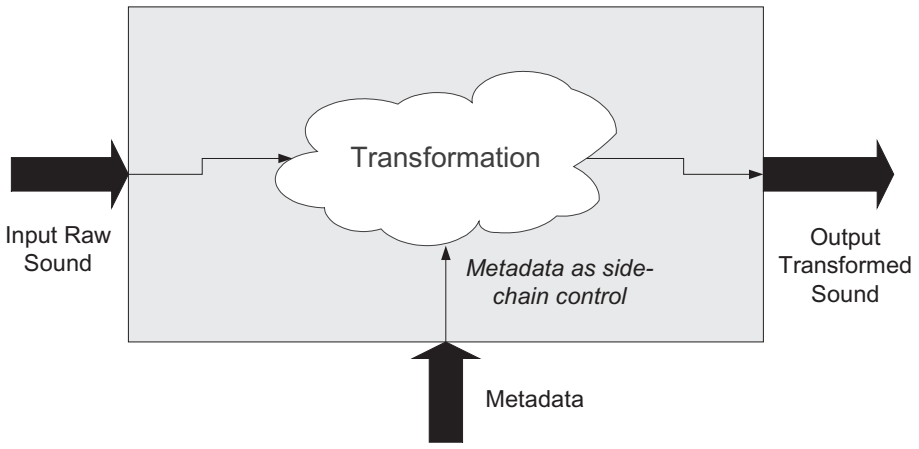

Fig. 3. Content description in the form of metadata can be a part of the incoming signal.

formed and then synthesized back (see Fig. 2) (Serra & Bonada, 1998; Amatriain et al., 2002).

Different analysis-synthesis schemes and methods have been proposed and are used in different situations yielding different results. Most of these transformations are aimed at providing a representation of the signal in a different domain, namely the frequency domain. Among the most used transforms, there are STFT (Short-time Fourier Transform), DCT (Discrete Cosine Transform), Wavelets and Haar. The first of these is by far the most commonly used and will be referred to later with more detail.

In Verfaille and Arfib, 2001, several transformations were implemented taking into account frequency information as well as changes on the analysis and synthesis hop size and window length. The transformations (generically called "Adaptive Effects") are applied in the frequency domain through a simple phase vocoder implementation that is controlled by features such as the fundamental frequency, the spectral centroid, a voiced/unvoiced gate (computed using the autocorrelation of the frame) or the energy (RMS). Using this sort of transformation scheme, we can implement effects such as a voiced/unvoiced driven time stretch, or a robotization whose fundamental frequency is controlled by the energy of the input sound. All these transformations can yield a great change in the expressivity of a spoken or sung voice.

1.1 Content description

Sometimes, the analysis step may be skipped because the input stream already contains metadata that can be used for the transformation process. This metadata is called "content description." In Figure 3 we illustrate this situation.

An example of such a transformation would be, for instance, a genre-dependent equalization. By applying some of the existing genre taxonomies we could add metadata defining the genre of a given piece of music. The classification could be performed either manually or by using a combination of a previously existing metadata that included, for example, author and title. The transformation block would then implement a basic filtering process that loads different filtering function templates depending on the genre.

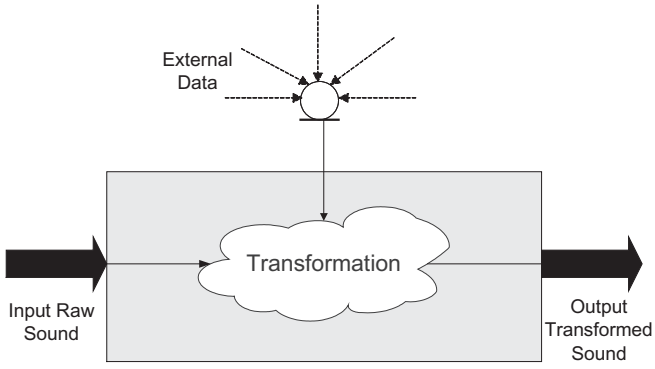

Fig. 4. Context awareness as a means of control.

This leads us to another discussion: are there appropriate formats for metadata description? The answer to this question is two-fold. On the one hand, some proprietary formats are available, most of them focusing on a particular application. On the other hand, some committees are also aiming at providing suitable standards for multimedia content description, and that obviously include audio and music. Maybe the most feasible effort is that put forward by the MPEG committee in its MPEG-7 standard (Manjunath et al., 2002). According to Martínez (2002): "MPEG-7, formally named "Multimedia Content Description Interface," is a standard for describing the multimedia content data that supports some degree of interpretation of the information's meaning, which can be passed onto, or accessed by, a device or a computer code. MPEG-7 is not aimed at any one application in particular; rather, the elements that MPEG-7 standardizes support as broad a range of applications as possible." Metadata will be available in a "readable" textual format, namely XML, and in a "more efficient" binary format.

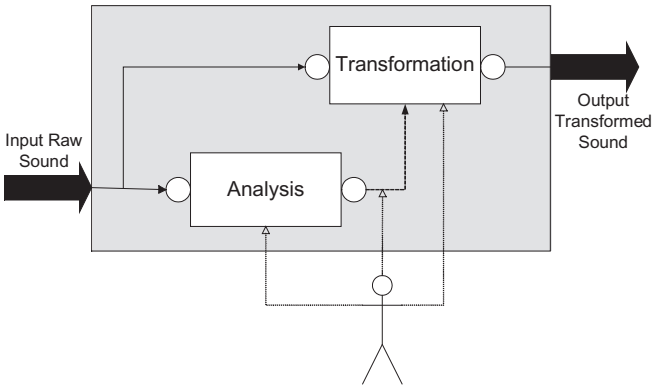

Fig. 5. User inputs to the transformation thread.

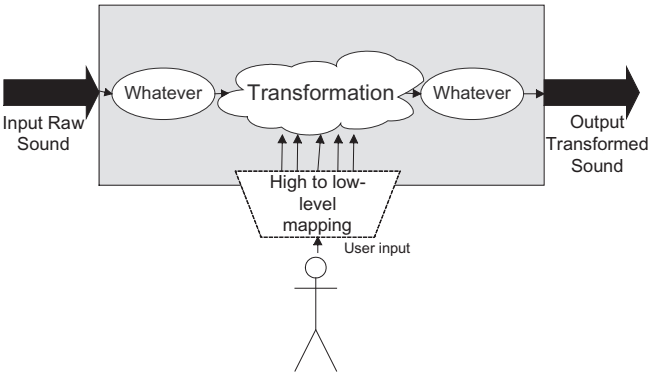

Fig. 6. High to low-level mapping at a control level.

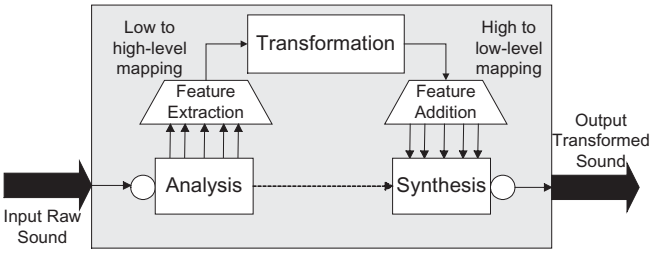

Fig. 7. High to low-level mapping at the analysis step.

Arguably, even another form of content transformation is that based on context awareness (see Fig. 4). By context awareness we mean the ability of a particular system of becoming "aware" of its surrounding world. In that sense, a dynamic processor whose threshold depends on the noise-level of the room would be an example of such a scheme.

Furthermore, context awareness is very much related to user profiling. A transformation system can respond differently according to the loaded user model. This user model can include information about user preferences as well as contextual information such as whether the user is happy or not (Chai & Vercoe, 2000).

1.2 User interaction

Even in such a simple example as the one of the automatic gain control, the "user input" must somehow be taken into account (the threshold and the slope must somehow be set). In that sense, the previous scheme must be modified in order to include this new input. A first version of the new scenario feeds this information directly into the analysis process so the user can control the settings of this particular step. The influence of the user's actions is directly on the features extracted from the signal.

Furthermore, the user may be able to directly interact with the output of the analysis process and so change the characteristics of the sound features before using them as a control of the actual transformation. Now, the influence of the user's actions is on the mapping function between the features extracted from the signal and the transformation control parameters. For example, we can take into account N features to control M parameters of the transformation, or more simply (using some sort of linear combination) take into account N features to control a single parameter of the transformation process. This way, the behavior that a given transformation will have on a particular sound is much more predictable and coherent to the characteristics of the sound itself. Yet another example of the interaction of the user in the transformation process is at the previously introduced stage of linear mapping between features and transformation control. Non-linearities, such as smoothing to avoid rapid transitions or truncation of the feature curve in order to select only the part of interest, may be introduced and directly controlled by the user's input.

The user input can also be directly fed to the transformation block in order to change the parameters of the actual transformation process. The influence of the user's action is now on the transformation controls (which will be generally different from those controlled by the extracted features). The following diagram illustrates the different possible user-inputs to the transformation thread.

But, as we already mentioned, when we talk about content processing, our focus is somehow shifted towards the final user of the system. The scenarios and examples of user input seen up until now suppose the user is still interacting with the transformation at a low-level. Thus, the user is seen more as an algorithm tweaking signal engineer than as a musician or artist.

But, in most cases, when we talk about content-based transformations, we imply that some sort of mapping between low-level parameters and higher-level ones is being performed. The aim of such a mapping is to group and relate features in such a way that they become meaningful for the targeted user. Still, the level of abstraction of the final controls has a lot to do with the profile of that targeted user. An expert user may require low-level, fine-tuning while a naive user will prefer high-level, easy to grasp parameters.

In the simplest case, the mapping between low and high-level parameters is done at the control level. The user input

is processed and mapped to the low-level parameters affected by that particular control (see Fig. 6).

But this mapping can already be performed at the analysis stage. Thus, these higher-level features are analyzed and extracted from the sound in such a way that the user can interact with them in a meaningful way (see Fig. 7). We will see this process in more detail in the following sections.

It is clear that the choice of a good mapping strategy is absolutely necessary if we aim at providing a user-oriented content transformation. Many studies have focused on mapping human gestures to low-level synthesis parameters (see Butch et al., 1997; Schoner et al., 1998; Todoroff, 2002; Wanderley et al., 2000, for example). Our focus here may seem different (because we are not dealing with physical gestures) but it is not so. The *intention* of a sound designer or musician using a transformation from a high-level approach can in many ways be seen as a musical *gesture*. Indeed, it is also a so-called haptic function, that is a low-frequency (compared to the frequencies in the sound signal itself) change in the control values.

2. Spectral models

The frequency domain representation of a sound is in many cases closer to our perceptual understanding than its time domain counterpart. Of the four main perceptual features of a sound (pitch, timbre, loudness, and duration), the first two are better interpreted in the frequency domain.

Many interesting content-based transformation frameworks or applications are based on one of the different existing spectral model (see Rodet et al., 1995; Arfib et al., 2002, or any of the applications later explained in this article).

As already mentioned, the most used transform is the STFT, and to be more precise, the set of fast algorithms that implement it and are generically known as FFT. The FFT yields a sampled complex spectrum at its output. The number of complex values corresponds to half the number of samples at its input, spread over half the original sampling rate.

We can already analyze a number of features from this spectrum and implement useful transformations. Examples of transformations that can be to some extent implemented after this step are: time scaling, pitch shifting, cross-synthesis . . .

But, using the output of the *STFT*, The *Sinusoidal* model represents a step towards a more flexible representations while compromising both sound fidelity and computing time. It is based on modeling the time-varying spectral characteristics of a sound as sums of time-varying sinusoids. The input sound $s(t)$ is modeled by,

$$s(t) = \sum_{r=1}^{R} A_r(t)\cos[\theta_r(t)] \tag{1}$$

where $A_r(t)$ and $\theta_r(t)$ are the instantaneous amplitude and phase of the r^{th} sinusoid, respectively.

To obtain a sinusoidal representation from a sound, an analysis is performed in order to estimate the instantaneous amplitudes and phases of the sinusoids. This estimation is generally done by first computing the STFT of the sound, then detecting the spectral peaks (and measuring the magnitude, frequency and phase of each one), and finally organizing them as time-varying sinusoidal tracks.

Sinusoidal Modeling is a quite general technique that can be used in a wide range of sounds and offers a gain in flexibility compared with the direct *STFT* implementation (McAulay & Quatieri, 1986).

Furthermore, the *Sinusoidal plus Residual* model can cover a wide "compromise space" and can in fact be seen as the generalization of both the *STFT* and the *Sinusoidal* models. Using this approach, we can decide what part of the spectral information is modeled as *sinusoids* and what is left as *STFT*. With a good analysis, the *Sinusoidal plus Residual* representation is very flexible while maintaining a good sound fidelity and the representation is quite efficient. In this approach, the *Sinusoidal* representation is used to model only the stable partials of a sound. The sum of these sinusoid is the called *Sinudoidal Component*. The *Residual Component*, or its approximation, models what is left, which should ideally be a stochastic component. It is less general than either the *STFT* or the S*inusoidal* representations but it results in an enormous gain in flexibility (Serra, 1989; Serra & Smith, 1990).

The input sound $s(t)$ is modeled by,

$$s(t) = \sum_{r=1}^{R} A_r(t)\cos[\theta_r(t)] + e(t) \tag{2}$$

where $A_r(t)$ and $\theta_r(t)$ are the instantaneous amplitude and phase of the r^{th} sinusoid, respectively, and $e(t)$ is the noise component at time t (in seconds).

The sinusoidal plus residual model assumes that the sinusoids are stable partials of the sound with a slowly changing amplitude and frequency. With this restriction, we are able to add major constraints to the detection of sinusoids in the spectrum and omit the detection of the phase of each peak. The instantaneous phase that appears in the equation is taken to be the integral of the instantaneous frequency $\omega_r(t)$, and therefore satisfies

$$\theta_r(t) = \int_0^t \omega_r(\tau)d\tau \tag{3}$$

where $\omega_r(\pi)$ is the frequency in radians, and r is the sinusoid number. When the sinusoids are used to model only the stable partials of the sound, we refer to this part of the sound as the deterministic component.

Within this model we can either leave the residual signal, $e(t)$, to be the difference between the original sound and the sinusoidal component, resulting into an identity system, or we can assume that $e(t)$ is a stochastic signal. In this case, the residual can be described as filtered white noise,

$$e(t) = \int_0^t h(t,\tau)u(\tau)d\tau \tag{4}$$

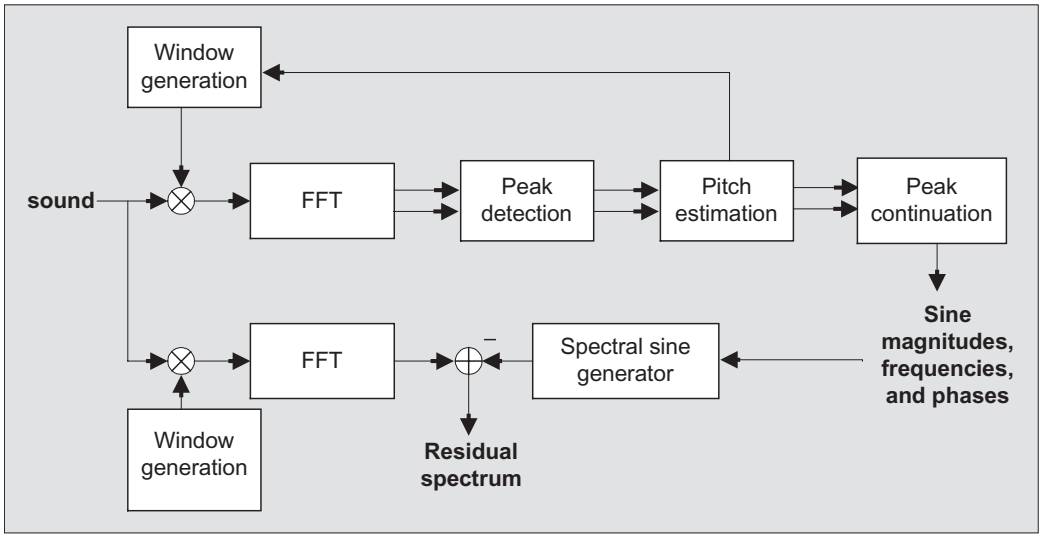

Fig. 8. Block diagram of the Sinusoidal plus residual analysis.

where $u(t)$ is white noise and $h(t,\tau)$ is the response of a time varying filter to an impulse at time t. That is, the residual is modeled by the time-domain convolution of white noise with a time-varying frequency-shaping filter.

The implementation of the analysis for the Sinusoidal plus Residual Model is more complex than the one for the Sinusoidal Model. Figure 8 shows a simplified block-diagram of this analysis.

The first few steps are the same than in a sinusoidal-only analysis. The major differences start on the peak continuation process since in order to have a good partial-residual decomposition we have to refine the peak-continuation process in such a way as to be able to identify the stable partials of the sound. Several strategies can be used to accomplish this. The simplest case is when the sound is monophonic and pseudo-harmonic. By using the fundamental frequency information in the peak continuation algorithm, we can easily identify the harmonic partials.

The residual component can be obtained by (a), directly substracting the generated sinusoidal spectrum from the FFT transform of the original signal as depicted in Figure 8 or by (b) first generating the sinusoidal component with additive synthesis, and then subtracting it from the original waveform. This is possible because the phases of the original sound are matched and therefore the shape of the time domain waveform preserved. A spectral analysis of this time domain residual is done by first windowing it, window which is independent of the one used to find sinusoids, and thus we are free to choose a different time-frequency compromise. An amplitude correction step can improve the time smearing produced in the sinusoidal subtraction. Then the FFT is computed and a fitting curve is applied to the magnitude spectrum to approximate the resulting spectrum. The spectral phases might be discarded when the residual is a stochastic signal.

3. Feature extraction

The accomplishment of a meaningful parameterization for content-based sound transformation applications is a difficult task. We want a parameterization offering an intuitive control over the sound transformation process. Our particular model should grant access to most of the perceptual and contextual attributes of a sound.

The extraction of such attributes should be performed in such a way that, based on our model, we should be able to implement transformations that modify only one of those features without affecting the rest. The key point is thus to extract features that are as much as possible decorrelated from the others so as to allow their transformation without affecting other perceptually meaningful features of the sound. Most of the features that we can obtain from a sound are information attributes that describe its characteristics and have only found applications in the analysis and classification of sounds. They are still of little relevance for designing transformations although some of them have already found use as perceptual-based controls for other transformation blocks. In section 5.7, for example, we introduce a Gender Change. It might be of interest for some applications to have a *gender* control in a vocal multi-effects unit to set the degree of masculinity or femininity of a given voice.

To that aim, one can distinguish at least three levels of abstraction (extraction) from the signal: at any point of the signal, in small arbitrary regions (i.e., frames) and in longer pre-segmented regions.

3.1 Instantaneous descriptors

The set of features that can be extracted at any point in the signal are called *instantaneous* descriptors. In the case of a time domain representation, most of the useful instantaneous

values that can be computed are related to the amplitude or energy of the signal.

If we are dealing with a frequency-domain representation many spectrally-related instantaneous features, such as the spectral centroid or the spectral tilt, can be computed on a given point. Note that, as explained latter, a given spectrum does not have an associated time duration, just a time associated with the middle of the analysis frame. In any case, and to be more precise, one should consider these descriptors as "nearly instantaneous" as they are not associated to a point in time of the signal but rather to a small region or frame.

3.2 Segmentation

An important step towards a musically useful parameterization is the segmentation of a sound into regions that are homogeneous in terms of a set of sound attributes. The goal is to identify regions that, using the signal properties, can then be classified in terms of their content. This way we can identify and extract region attributes that will give higher-level control over the sound.

A useful segmentation process applied to a monophonic source divides a melody into notes and silences and then each note into an attack, a steady state and a release regions. Attack and release regions are identified by the way the instantaneous attributes change in time and the steady state regions are detected by the stability of these same attributes. Global attributes that can characterize attacks and releases refer to the average variation of each of the instantaneous attributes, such as average fundamental frequency variation, average amplitude variation, or average spectral shape change. In the steady state regions, it is meaningful to extract the average of each of the instantaneous attributes and measure other global attributes such as time-varying rate and depth of vibrato.

Sound segmentation has proven important in automatic speech recognition and music transcription systems. For our purposes it is also very valuable as a way to apply region dependent transformations. For example, the time stretching algorithm introduced in section 5.3 detects regions with transients in order to preserve their perceptual singularity.

The techniques originally developed for speech (Vidal & Marzal, 1990), based on Pattern-Recognition, Knowledge-Based or Neural Network methodologies, start to be used in music segmentation applications (Rossignol et al., 1999). Most of the approaches apply classification methods that start from sound features, such as the ones described in this paper, and are able to group sequences of frames into predefined categories. No reliable and general-purpose technique has been found. Our experience is that we need to narrow down the problem to a specific type of musical signal or to include a user intervention stage to guide the segmentation process.

3.3 Region attributes

Once a given sound has been segmented into regions we can study and extract the attributes that describe each one. Most of the interesting attributes are simply the mean and variance of each of the frame attributes for the whole region. For example, we can compute the mean and variance for the amplitude of sinusoidal and residual components, the fundamental frequency, the spectral shape of sinusoidal and residual components, or the spectral tilt.

Region attributes can be extracted from the frame attributes in the same way that the frame attributes are extracted from the frame data. The result of the extraction of the frame and region attributes is a hierarchical multi-level data structure where each level represents a different sound abstraction. From several sound representations it is possible to extract the type of attributes mentioned above. The critical issue is how to extract them in order to minimize interferences, thus obtaining, as much as possible, meaningful high-level attributes free of correlations.

The sound transformation approaches exemplified in this article are based on a spectral-based analysis/synthesis framework. For the feature extraction part, we first identify instantaneous attributes and their derivatives in the frequency domain, then we segment the sound, and finally we can extract region attributes.

3.4 Extracting attributes from the Sinusoidal plus Residual model

As already mentioned, it is common practice to extract many of the features of an audio signal from a frequency domain analysis. We will now concentrate on the extraction of attributes from the Sinusoidal plus Residual model described in section 2.

Some of the basic instantaneous attributes of the Sinusoidal plus Residual model at the frame level are: amplitude power of sinusoidal and residual components, total amplitude power, fundamental frequency, spectral shape of sinusoidal and residual components, harmonicity and spectral centroid. These attributes are obtained at each frame using the information that results from the basic Sinusoidal plus Residual analysis and not taking into account the data from previous or future frames.

The power of the sinusoidal component is the sum of the power of all harmonics of the current frame expressed in dB,

$$PS_{total} = 10\log_{10}\left(\sum_{i=1}^{I} a_i^2\right) \tag{5}$$

where a_i is the linear amplitude of the ith harmonic and I is the total number of harmonics found in the current frame.

The power of the residual component is the sum of the absolute values of the residual of the current frame expressed in dB. This amplitude can also be computed by adding the frequency samples of the corresponding power spectrum,

$$\begin{aligned} PR_{total} &= 10\log_{10}\left(\sum_{n=0}^{M-1} |x^2{}_R(n)|\right) \\ &= 10\log_{10}\left(\sum_{k=0}^{N-1} |X^2{}_R(k)|\right) \end{aligned} \tag{6}$$

where $x_R(n)$ is the residual sound, M is the size of the frame, $X_R(k)$ is the spectrum of the residual sound, N is the size of the magnitude spectrum and k is the spectral bin index.

The total power of the sound at the current frame is the sum of its absolute values expressed in dB. It can also be computed by summing the power of the sinusoidal and residual components,

$$\begin{aligned} P_{total} &= 10\log_{10}\left(\sum_{n=0}^{M-1}\left|x^2(n)\right|\right) \\ &= 10\log_{10}\left(\sum_{k=0}^{N-1}\left|X^2(k)\right|\right) \\ &= 10\log_{10}\left(\sum_{i=1}^{I} a^2{}_i + \sum_{k=0}^{N-1}\left|X^2{}_R(k)\right|\right) \end{aligned} \tag{7}$$

where $x(n)$ is the original sound and $X(k)$ is its spectrum.

The fundamental frequency is the frequency that best explains the harmonics of the current frame. Many different algorithms can be used to compute the fundamental (see Gómez et al., 2002 in this same issue) but a reasonable approximation can be the weighted average of all the normalized harmonic frequencies,

$$F_0 = \sum_{i=1}^{I} \frac{f_i}{i} \times \frac{a_i}{\sum_{i=1}^{I} a_i} \tag{8}$$

where f_i is the frequency of the ith harmonic peak output from the proposed spectral analysis (see Fig. 8).

The spectral shape of the sinusoidal component is the envelope described by the amplitudes and frequencies of the harmonics, or its approximation,

$$Sshape = \{(f_1, a_1)(f_2, a_2)\ldots(f_I, a_I)\} \tag{9}$$

The spectral shape of the residual component is an approximation of the magnitude spectrum of the residual sound at the current frame. A simple function is computed as the line segment approximation of the spectrum,

$$\begin{aligned} Rshape = \{e_1, e_2, \ldots, e_q, \ldots, e_{N/M}\} = \\ \max_k[|X_R(qM + k)|] \end{aligned} \tag{10}$$

where $k = -M/2, -M/2+1, \ldots, \ldots, M/2 - 1$, and M is the number of frequency samples used for each calculation of a local maximum. Other spectral approximation techniques can be considered depending on the type of residual and the application.

The computation of many attributes depends on whether the given sound is harmonic or not. Thus, an analysis of the harmonicity of the sound must be performed. A possible way of computing this coefficient is given by:

$$Harm = \sum_{i=1}^{I} |f_i - (F_0 \times i)| \frac{a_i}{\sum_{i=1}^{I} a_i} \tag{11}$$

where $A(fr, h)$ denotes the amplitude of the h^{th} harmonic in the spectrum computed at frame fr. Note that in the previous formula, a value close to 0 will mean that the sound is harmonic (value of 0 will only be possible in synthetic sounds).

Once the analyzed sound has been categorized either as harmonic or inharmonic, different formulas can be used to compute a given descriptor. In the case of the spectral centroid (Petters et al., 1999), for instance, for harmonic sounds the computation is given by:

$$Hcentroid = \frac{\sum_{i=1}^{I} f_i \times a_i}{\sum_{i=1}^{I} a_i} \tag{12}$$

whereas for inharmonic sounds the centroid is computed using the whole spectrum

$$Icentroid = \frac{\sum_{k=0}^{N-1} f_k \times a_k}{\sum_{k=0}^{N-1} a_k} \tag{13}$$

where $f_k = \frac{k}{2N} \times SR$, k is the spectral bin index, N is the size of the spectrun and SR is the sampling rate. (The centroid, which is here computed using the amplitude of the spectral bins or harmonics, can also be computed using energy instead.)

Other features that can be extracted from the Sinusoidal plus Residual model and that have found applications in some fields are: the number of sinusoids, energy of any of the components, spectral tilt, noisiness, odd/even partial ratio, attack harmonic coherence, derivative or relative derivative of any instantaneous attribute, spectral flux, attack, and release times.

The frame-to-frame variation of each attribute is a useful measure of its time evolution, thus an indication of changes in the sound. It is computed in the same way for each attribute,

$$\Delta = \frac{Val(l) - Val(l-1)}{H/SR} \tag{14}$$

where $Val(l)$ is the attribute value for the current frame, $Val(l - 1)$ is the attribute value for the previous one and H is the sample distance from one frame to the next (hop-size) and SR the sampling rate.

Some implementations of frame to frame variation attributes though, use special computations that slightly differ from the general formula. In Petters et al. (1999), for example, the frame-to-frame variation of the spectral shape for harmonic sounds is computed using the following formula:

$$Hsv(fr) = \frac{\sum_{h=1}^{I} A(fr-1,h) * A(fr,h)}{\sqrt{\sum_h A^2(fr-1,h)}\sqrt{\sum_h A^2(fr,h)}} \tag{15}$$

4. Perceptually adapted features

Up until this point, we have extracted and computed features that are directly related to the signal-domain characteristics of the sound and, although they may have a perceptual meaning, they are not taking into account the importance of the listener's perceptual filter.

In that sense, for example, it is well known that the amplitude of the sound is not directly related to the sensation of loudness produced on the listener, not even in a logarithmic scale (see Moore et al., 1997). Fletcher and Munson (1933) established a set of equal-loudness curves called isophones. The main characteristic of these curves is that the relation between a logarithmic physical measure and its psychoacoustical counterpart is a frequency dependant function. Although this curves have proven only valid for the stable part of pure sinus (more than 500 ms), they have been used as a quite robust approximation for measuring loudness of complex mixtures (Pfeiffer, 1999).

The physical measure related to the psychoacoustical sensation called loudness is the sound pressure level (SPL), measured as:

$$L = 20\log_{10}\frac{p}{p_0}[d \tag{16}$$

where p_0 is the sound pressure level at the threshold of audition.

The problem of automatically relating this feature to subjective loudness has been addressed in several ways. See Pfeiffer (1999) for examples.

Also, and in a different sense, the spectrum of a signal, as the one computed by the STFT, could be perceptually adapted by taking into account the logarithmic frequency response of the human ear. That would lead us to the use of some sort of constant-Q transform scheme such as the Wavelet transform. Wavelets use different window sizes depending on the frequency range being analyzed. Thus, the spectral resolution is also frequency dependent, resembling the response of the human ear (Brown, 1991).

Although this perceptual adaption is theoretically interesting, it has found little application so far. The computation cost and complexity of perceptual features does seldom pay for the increase of naturalness that can be gained in the transformation. A good mapping scheme is usually enough. Nevertheless, not many conclusions have been put forward on this area of perceptual adapted sound transformations and it is, surely, a field to invest efforts in the near future.

5. Main axes in a sound transformation

The main perceptual axes in a sound are (arguably): timbre, pitch, loudness, duration, position and quality. Ideally, we are looking for transformations that can change the sound in one of its dimensions without affecting any other.

In the following sections we will briefly give some examples of transformation engines that have been developed with that idea in mind. Regarding pitch, we will present ways of transposing a sound without affecting its timbre. We will then outline the basic characteristics of the perceived loudness and some automatic systems that simulate its computation. Then, we will talk about time-scaling, which is, changing the duration of a sound without affecting its pitch or its timbre. We will also mention the main characteristics of Spatialization, which is the recreation of the sensation of space and location by artificial means. Another important axis is "sound quality." Different parameters are involved in this axis but maybe the clearest one is the signal to noise ratio, for that reason when talking about modifying the subjective quality of a given source, we will be talking about denoising algorithms. Finally, in the case of timbre, we will introduce the concept of timbre space and give some examples of sound morphing engines.

We will finish this section by giving some examples of how primitive transformations in any of these axes can be combined in order to create even more musically meaningful effects.

5.1 Pitch

Pitch shifting is an effect that aims at transposing the original pitch of a sound without affecting other perceptual features, namely its timbre and its duration. But, the first difficulty that we encounter is to define precisely what we mean by pitch.

Pitch can be defined as the frequency of the sinusoid that can be consistently matched by listeners to the sound being analyzed (Hartman, 1996). But although the term *pitch* should be reserved when talking about perceptual issues, in many practical applications the signal-domain concept of *fundamental frequency* has a very similar meaning. For this reason, and only in the context of this article, we do not make much distinction between the two terms. In fact, most pitch shifting systems are actually shifting the fundamental frequency and the harmonics of a sound.

The basic approach to implement a pitch-shifting algorithm is to transpose the spectrum of a sound without affecting its spectral shape (feature directly related to the timbre of a sound). A quite intuitive way to accomplishing this is by using the sinusoidal plus residual model. Once we have separated both components of the sound, we extract the spectral shape of the sinusoidal one, we multiply the frequency of every partial by a given factor, and we apply again the original spectral shape. A similar procedure may (or may not) be applied to the residual. (Of course, if the residual component does not need to be treated separately, other analysis-synthesis techniques, such as PSOLA or Phase Vocoder, may be used).

5.2 Loudness

As already mentioned in section 4, extracting this perceptual descriptor using an automatic extraction process is already a difficult task.

A further problem we can encounter is the gain introduced by the recording chain. Thus, when we analyze a sound file stored in our hard disk, we have no way of effectively measuring the loudness of the original sound, unless we have access to the settings of all the devices used in the original recording (microphone, preamplifier, mixer, computer soundcard, . . .).

We can overcome this problem by facing the fact that any possible transformation applied in the digital domain will be dealing with relative and non-absolute loudness values. In any case, the implementation of a loudness scaling effect should be based on a logarithmic scale.

But loudness is also used in a musical sense in order to represent the sound level of an acoustical instrument. The mechanisms that relate the actions of a player with the sound level produced by a given instruments are usually so complex that seldom this feature can be decorrelated from others such as timbre. Thus, the difference in sound between playing a soft and a loud note in an instrument is not only its sound level.

In the case of a piano, for example, in Solà (1997) a transformation was implemented in order to obtain all possible musical loudness (dynamics of a note) out of a single previously analyzed note. It was concluded, that the most feasible implementation was based on taking the highest possible dynamic as a starting point. Then, it is "just" a matter of subtracting the spectral information that is not needed to obtain the notes that have lower dynamic values.

For the case of the piano it was concluded that three different aspects of the spectral representation of the original sound needed to be controlled in order to apply that transformation in an effective and natural way. These are: the overall amplitude, the amplitude of the residual component (percussiveness of the attack) and the spectral shape of the sinusoidal sound (brightness).

5.3 Duration

Time-scaling an audio signal means changing the length of the sound without affecting other perceptual features, such as pitch or timbre. Many different techniques, both in time and frequency domain, have been proposed to implement this effect. Some frequency domain techniques yield high-quality results and can work with large scaling factors. However, they are bound to present some artifacts, like phasiness, loss of attack sharpness and loss of stereo image. In this section we will present a frequency domain technique for near lossless time-scale modification of a general musical stereo mix (Bonada, 2000).

The general block diagram of the system is represented in Figure 9. It is important to remark that the frame rate used in both the analysis and synthesis modules is the same, as opposed to the most broadly used time-scale techniques in which a change of frame rate in synthesis is used in order to achieve the effect. Therefore, in some cases an analysis frame is used twice (or more) while on other cases some frames

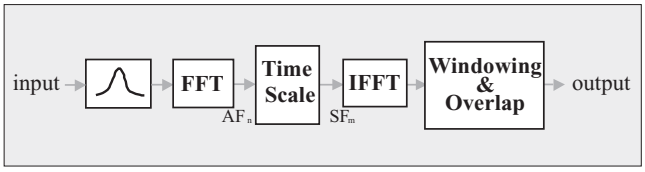

Fig. 9. General diagram of the time stretching system.

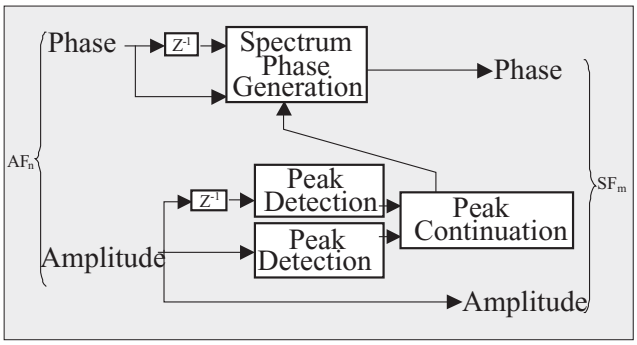

Fig. 10. The time-scale module.

are never used. This technique will not add any artifacts, provided the frame size we use is small enough and the sound does not present abrupt changes in that particular region.

In Figure 10, a more detailed block diagram of the time-scale module is shown. The analysis frames (AF_n), containing the spectrum amplitude and phase envelopes, are fed to the time-scaling module. This module performs a peak detection and a peak continuation algorithm on the current and previous (Z^{-1}) amplitude envelopes. These peaks are used as inputs to the spectrum phase generation module. Note that the time-scale module only changes the phase, leaving the spectral amplitude envelope as it is.

The usage of the same frame rate in analysis and synthesis allows us to suppose that the phase variation of each peak between two consecutives synthesis frames is the same as in the analysis frames.

The phase vocoder approach for time scaling results into very well-known artifacts. In this section we will describe each of these problems and the solution that the implementation we are proposing can provide.

In the phase vocoder implementation, the phase of each bin advances at different speed. This introduces a loss of peak's phase coherence that is known as *phasiness*. To avoid this problem we can apply the original relative behavior of the phase around the peak. Following Laroche and Dolson (1997) each peak location subdivides the spectrum into a different region, with a phase related to that of the peak. The phase around each peak is obtained by applying the delta phase function of the original spectrum phase envelope (see Fig. 11).

Another typical artifact of the phase vocoder approach is the smoothing of the attack transients. A possible solution is to modify the sinusoidal plus residual model in order to have a specific model for the transients (Verma & Meng, 1998).

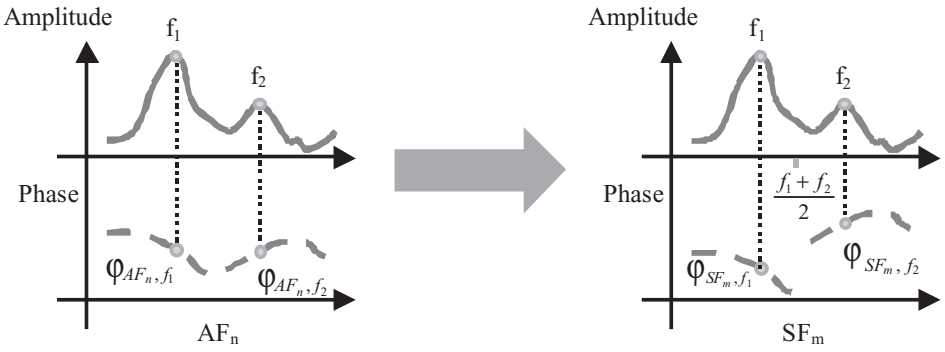

Fig. 11. Original delta phase function around each peak.

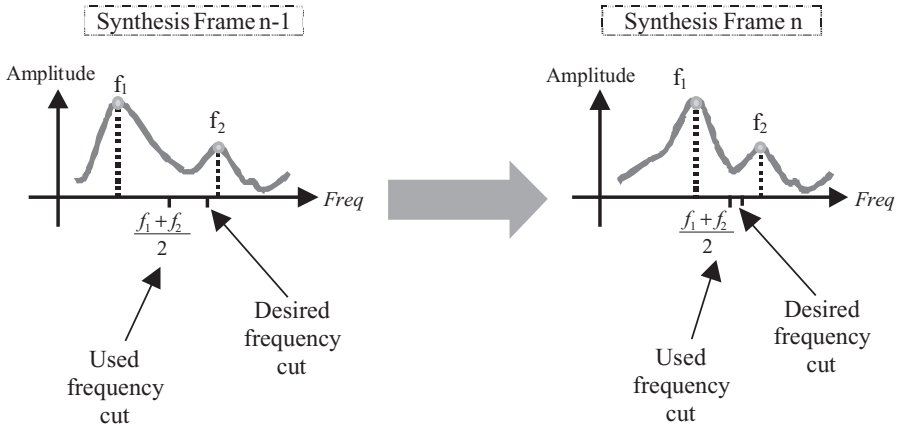

Fig. 12. Adaptive frequency cut.

Another possible approach is to not time-scale the input signal on this kind of regions so that the original duration is respected. Consequently, and in order to preserve the overall scaling factor, a greater amount of scaling should be applied to surrounding regions.

In order to apply the previous technique, it is necessary to detect attack transients of the sound in an unsupervised manner. The computation of relative changes of energy along several frequency bands can be used for that purpose. A low frequency band could, for example, detect sharp bass notes, while a high frequency band could be set to detect hits of a crash cymbal.

A basic property of the STFT implies that it is desirable to have long windows in order to achieve a high frequency resolution, but conversely it is also desirable to have short windows so to achieve a better temporal resolution. The solution proposed is to use parallel windowing, that is, several analysis channels. Obviously, the window should be longer for low frequencies than for high frequencies. The peak detection process is applied to each of the channels while the peak continuation takes care of the desired channel frequency cuts, so it can connect peaks of different channels. Then the time-scale module fills the spectrum of all the channels and applies a set of parallel filters $H_n(f)$ that must add up to a constant (*all pass filter*).

If the cutoff frequency of a channel was close to a spectral peak, this peak would be split into two different channels and we would be introducing artifacts. For this reason, we need to provide our system with time-varying frequency cuts. Each frequency cut is set to the middle point between the two closest to the original frequency cut (see Fig. 12).

In the case of stereo signals, if we process each of the two channels independently, most of the stereo image is bound to be lost. This artifact is mainly due to the fact that the time-scale process changes the phase relation between the two channels. Therefore, if we want to keep the stereo image, it is necessary to preserve the phase and amplitude relation between left and right channels. In the system presented, the amplitude relation is already preserved because the spectrum amplitude is kept unchanged. On the other hand, the phase relation is forced to be the desired one, bin by bin.

The system here presented can deal with time varying scaling factors with no loss of quality tradeoff. The only significant change is that the time increments of the synthesis frames expressed in the input signal are not constant. The application of time-varying tempo variations opens up many new and interesting perspectives. The system could be easily adapted and used for alignment and synchronization of two sound sources. It could be also used to control the tempo of

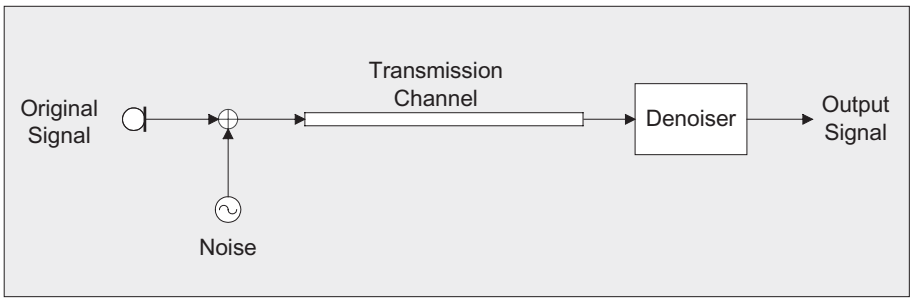

Fig. 13. Basic signal denoising scheme.

a piece of music, like a conductor does, and by using a score following system and audio accompaniment could be synchronized in real-time with a live performance.

5.4 Space

Spatial location is often referred to as another basic dimension in audio processing. It is far beyond the scope of this paper to go into the details of the many techniques that have been developed for sound spatialization. See Rocchesso (2002) for an introduction to spatial effects with practical applications.

It is interesting to note, though, that some of the currently used techniques for spatialization rely on the content of the audiovisual material in order to process the audio signal and decide to what channel the signal has to be sent to. In this sense, it is interesting to mention MPEG-4's AudioBIFS (Binary Format for Scenes) that takes this idea a step beyond, defining "virtual scenes" made up of sound objects that organized through scene graphs where the nodes represent the actual content (Scheirer et al., 1998).

5.5 Quality

Since the very beginning of information technology, the most obvious transformation that was pursued when processing a sound was to reduce the noise that somehow had been added to the original source. The problem, though, is that when the sound is processed no information is available about the source and only some hypothesis can be made about the noise signal. Thus, the feasible goal in this type of situation is to increase the signal-to-noise ratio (SNR) since the complete elimination of the noise is a rather impossible objective in most cases.

For that reason, most of the traditional systems are based on filtering schemes or statistical processing algorithms that aim at modeling the noise and the signal source.

Although a filter might be implemented in the time domain by convolving the impulse response of the filter with the signal most of the denoising systems are implemented in a frequency domain where the filter is implemented by multiplying the frequency response of the filter with the transform of the signal. More recently, wavelets and the Karhunen-Loeve (Mittal et al., 2000) transform have been used for optimizing denoising algorithms.

As far as we are aware, the only "content-dependant" denoising algorithms that have been implemented are those used in speech transmission systems. These algorithms can take advantage of the content in the sense that they use models to identify the spoken signal so to get rid of the noisy component.

Even so, more sophisticated techniques that include machine learning techniques have already been used. In Di Giura et al. (1997), for example, a denoising technique based on fuzzy rules was implemented.

5.6 Timbre

Maybe the most obvious dimension we think when dealing with content transformation is the timbre of a given sound. Timbre is defined as all those characteristics that distinguish two sounds of the same pitch, duration and loudness. As a matter of fact, the psychoacoustical sensation timbre depends on many characteristics of the signal such as its instantaneous spectral shape and its evolution, the relation of its harmonics or some other features related to the attack, release and temporal structure.

On the other hand, timbre is hardly decorrelated from other features in music. For example, when an instrument plays two notes at different loudness or pitches, the timbre is very much likely to vary accordingly.

Ideally, we would like to have some sort of representation of timbre that allowed transforming continuously from one timbre to a different one. For example, we would like to gradually transform the sound of a trumpet into that of a violin. And even more, we would like to be able to define spaces where a given point has intermediate features corresponding to each of the axis.

This leads us to the idea of *Timbre Space*. This control structure has been used in many different ways taking a variety of forms and even of labels but it is commonly acknowledged that it was first named in Wessel (1979).

In Amatriain et al. (1998), we implemented a timbre space control structure. The axes of the space were user-defined, so each of the dimensions corresponded to a feature the instrument designer decided could be perceptually important. The

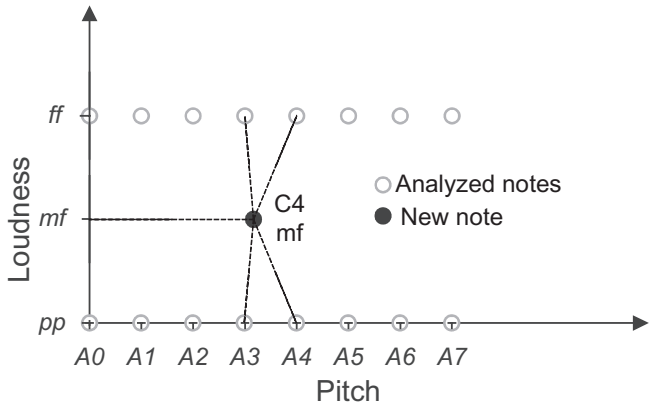

Fig. 14. Two-dimensional timbre space.

outputs of an SMS (Spectral Modeling Synthesis) (Serra, 1998) analysis were then positioned in the desired spatial coordinates. Intermediate values were obtained by interpolating values of spectral features. As shown in the next figure, a timbre space is a control structure that can be also used to control features of the sound that are not essentially timbral but that influence it, such as pitch or loudness. In the particular example illustrated, a note with intermediate loudness and pitch (C4, mf) is obtained by interpolating spectral data from previously analyzed notes. (From Amatriain et al., 1998).

An example of a perceptual derived timbre space can be found in the Timbre Descriptor included in the MPEG-7 standard (Petters et al., 1999). In that case, the issue was not to find the most interesting dimensions for transforming the timbre of a given instrument but rather to use those dimensions in a search and retrieval scenario. Two different spaces were derived from psychoacoustical experiments (Krumhals, 1989; McAdams et al., 1995; Lakatos, 1999): one for percussive and another for harmonic instruments. In all experiments, listeners were asked to perform similarity measures on pairs of sounds. Multidimensional Scaling (MDS) was then used to translate results into a low-dimensional space. Then the descriptor that best explained each axis was chosen and a mathematical relation between axis position and descriptor value was derived by a linear-regression method. Finally, position and distance methods were used for justifying independence of descriptors and the use of an euclidian distance and for estimating the signal descriptors coefficients in the distance measure. (Peeters, 2000).

In the case of the percussive instruments a two dimensional space was used. The first dimension, corresponding to the perception of temporal characteristics included both log attack time and temporal centroid, the second dimension was based only in one descriptor, spectral centroid. On the other hand, a four dimensional space was derived for harmonic instruments. One dimension corresponded to the temporal perception and was only based on the log attack time while the other three included spectral parameters, namely spectral centroid, spectral deviation, spectral spread and spectral variation (all of them computed on the harmonic component of the related spectrum).

Morphing

Out of the interpolation of data resulting from two or more analyses of audio signals, we can create the so-called sound hybrids or morphs. Most of the morphing techniques are based on the interpolation of sound parameterizations resulting from analysis/synthesis techniques, such as the Short-time Fourier Transform (STFT), Linear Predictive Coding (LPC), Cepstrum or Sinusoidal Models. Morphing is not only applied on the timbre related attributes of the sounds but can be used to obtain different characteristics of a given source in a dimension where a "traditional" transformation would be hard to accomplish. It is in the timbre domain where this technique yields results impossible to obtain using other transformation techniques. That is the reason why this technique has been included in this section and an example of an application that uses it extensively is now given.

An automatic impersonating system that allows the user to morph in real-time his/her voice attributes (such as pitch, timbre, vibrato and articulations) with the ones from a pre-recorded singer (which from now on we will refer to as *target*) was developed (Cano et al., 2000) for a karaoke type application. Such particular case of morph is after controlling the resulting synthetic voice by mixing some characteristics of the two singing voice signals.

Figure 15 shows the general block diagram of the voice impersonator system. The system relies on two main techniques that define and constrict the architecture: the SMS framework and a Hidden Markov Model based Speech Recognizer (ASR). The SMS implementation is responsible of providing a suitable parameterization of the singing voice in order to perform the morph in a flexible and musically-meaningful way. On the other hand, the SR is responsible of matching the singing voice of the user with the target.

The system requires the phonetic transcription of the lyrics, the melody as MIDI data, and the actual recording to be used as the target audio data, which is usually the target's performance of the complete song to be morphed. This recording has to be analyzed with SMS, segmented into morphing units (phonemes), and each unit labeled with the appropriate note and phonetic information of the song beforehand. Once this preparation stage is concluded we can start processing the user's voice. The SMS frame based analysis outcomes a set of SMS analyzed frames with its corresponding appropriate parameterization. Each of these frames is then associated with the phoneme of a specific moment of the song and thus with a target frame. The system first recognizes what the user is singing (phonemes and notes) and then looks for the same sounds in the target's performance (i.e., synchronizing the sounds). Once a user frame is matched with a target frame, they are morphed by interpolating their parameters and the new morphed values are then added back to the synthesis frame of the user. Finally

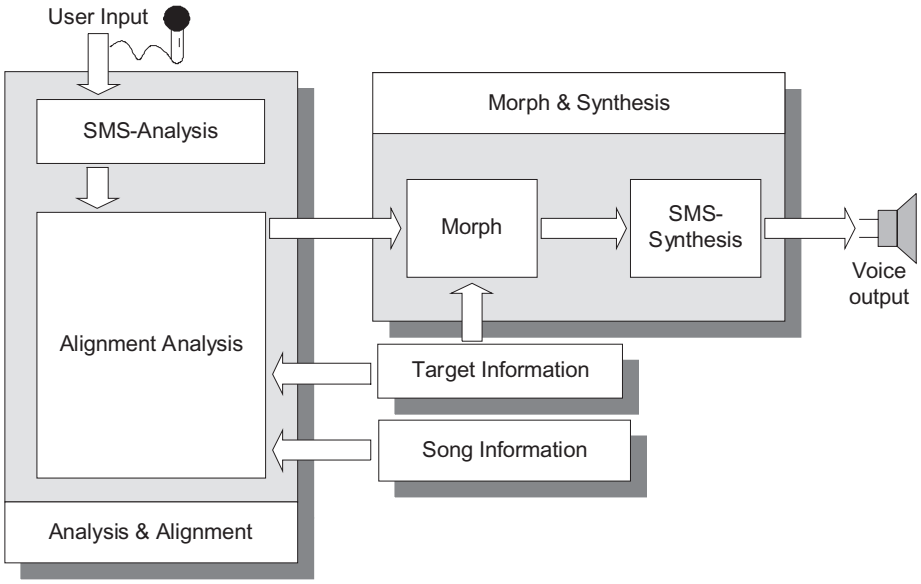

Fig. 15. Impersonating system block diagram.

the synthesis is done with the standard synthesis procedures of SMS. All this is accomplished in real-time.

Only voiced phonemes are morphed and the user has control over which and by how much each parameter is interpolated. The frames belonging to unvoiced phonemes are left untouched, thus always having the user's unvoiced consonants in the output. In general, the amplitude will not be interpolated, thus always using the amplitude from the user. This will give the user the feeling of being in control.

Several modifications are done to the basic SMS procedures to adapt them to the requirements of the impersonator system. The major changes include the real-time implementation of the whole analysis/synthesis process with a processing latency of less than 30 milliseconds and the tuning of all parameters to the particular case of the singing voice. These modifications include the extraction of higher-level parameters meaningful in the case of the singing voice and that will be later used in the morphing process.

To solve the matching problem the system includes an Automatic Speech Recognizer (ASR) based on phoneme-base discrete HMM's. This ASR has been adapted to handle musical information and work with very low delay (Loscos et al., 1999) since we cannot wait for a phoneme to be finished before we recognize, moreover, we have to assign a phoneme to each frame. This would be a rather impossible/impractical situation if it was not for the fact that the lyrics of the song are known beforehand. This reduces a big portion of the search problem: all the possible paths are restricted to just one string of phonemes, with several possible pronunciations. The problem is cut down to the question of locating the phoneme in the lyrics and placing the start and end points.

Besides knowing the lyrics, musical information is also available. The user is singing along with the music, and hopefully according to a tempo and melody already specified in the score. Thus, we also know the time at which a phoneme is supposed to be sung, its approximate duration, its associated pitch, etc. All this information is used to improve the performance of the recognizer and also to allow resynchronization, for example in the case that the singer skips a part of the song.

In most cases, the durations of the user and target phonemes to be morphed will be different. If a given user's phoneme is shorter than the one from the target, the system will simply skip the remaining part of the target phoneme and go directly to the articulation portion. In the case when the user sings a longer phoneme than the one present in the target data, the system applies some looping techniques in order to stretch the synthesis.

5.7 Combining different axes

By combining different "basic" effects we are able to step higher in the level of abstraction and get closer to what a user could ask for in a sound transformation environment, we can change, for example, the gender of a given vocal sound.

The timbre of a vocal sound is basically characterized by its spectral shape, namely the position of the spectral resonances known as formants.

A basic way to change the gender of a voice is thus to apply two transformations: a pitch transposition and a shift in the spectral shape. To convert a male voice into a female one we can transpose an octave higher and apply a spectral shape shift (when a female singer rises up the pitch, the formants move along with the fundamental). To do the inverse transform we also transpose the pitch an octave lower but the spectral shape shift has to be performed in such a way that the formants of the female voice remain stable along different pitches.

6. Changing the musical meaning: techniques for expressive transformations

The systems described so far, although addressing the content-level, they are focused on transforming sound properties that are more or less related to signal properties. But, if we raise our level of abstraction a bit more, we may address the issue of how these sound transformations may affect the musical (symbolic) layer. Or the other way round, how can we transform the sound in order to accomplish a musically meaningful effect.

The transformations presented in this section are mainly focused on exploiting the musical structure of the sound for generating expressive transformations. They can be also described as decision processes that determine the mapping between input parameters and control parameters driving the different basic transformations described in the previous section (pitch, loudness, duration, and timbre). For instance, a time scaling system can be used for transforming the durations of the notes contained in a sound, given a musical intention. Thus, these musical transformations are concerned with musical concepts such as rubato, characteristics of note attacks, or articulation.

In order to perform transformations at the musical level, we have a new requirement: we need to extract musical features of the sound at the analysis step (see Fig. 7). These features can be numerical – like the starting time of a note- or symbolic – like the categorization of sound regions as note attacks, note releases, or note decays. Moreover, the values for these features can have an associated degree of uncertainty.

Another important aspect of musical expressive transformations is that some of them contain a subjective component and/or only have partial formal models-like for example, the expression of emotional content. These two characteristics have motivated the use of artificial intelligence (AI) techniques for implementing musical transformations. The models used in the musical transformation systems are then acquired by means of the analysis of collections of sound data. Thus, these AI approaches introduce an additional off-line phase of training/learning. Then, the idea illustrated in Figure 16 enriches the schema of sound transformation by introducing this training step. This difference with respect to the transformations presented in the previous section is also important because some systems can use different features for the training phase and for the transformation phase. For example, the training phase for learning time variation rules can be performed using MIDI data as input but generating knowledge (in a form of a collection of rules) that can be applied to more general analysis features.

In the next two subsections we will give examples of musical transformation systems grouped by different axes: tempo, dynamics, and articulation transformations. Because the systems dealing with tempo transformations also deal with dynamics transformations, we will describe both transformations together. The final subsection will describe the

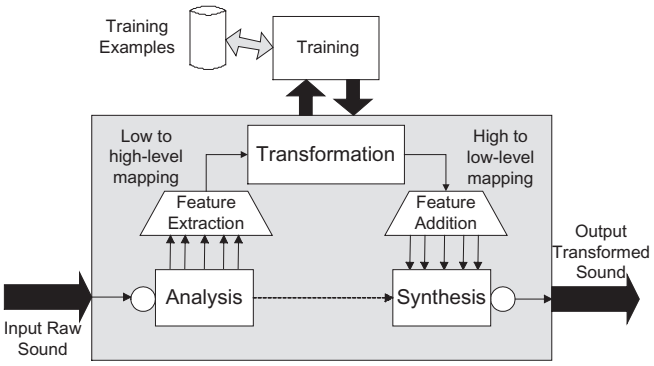

Fig. 16. Training cycle in the high to low-level mapping.

SaxEx system, an attempt to combine all three expressive axes using a spectral model as input/output. The SaxEx system achieves this goal by means of restricting the focus of application: monophonic jazz melodies.

6.1 Tempo and dynamics transformations

Transformations of tempo are related with transformations of time (time scaling). The goal in tempo transformations for a given sound is to decide an envelope for time scaling taking into account the musical content of that sound. That is, expanding or shorting notes and anticipating or delaying notes.

Transformations of dynamics are related with a change in loudness. The goal in dynamics transformations for a given sound is to decide an envelope for loudness transformations in a similar way than tempo transformations act in time scaling. For example, a common dynamics transformation is the generation of crescendo or diminuendo curves.

Rules for music performance

The rules for music performance designed by the KTH group (Bemdtsson, 1996; Friberg, 1995) were one of the first attempts to provide high-level musical transformations. They are not only restricted to tempo and dynamics transformations but the majority of them are applied to these two transformations. These rules are inferred either from theoretical musical knowledge or by experimental results (training), specially using the analysis-by-synthesis approach. The rules are divided in three main classes: *Differentiation rules*, which enhance the differences between scale tones; *Grouping rules*, which show what tones belong together; and *Ensemble rules*, that synchronize the various voices in an ensemble.

Artificial Neural Networks approach

Another approach taken for performing tempo and dynamics transformation is the use of artificial neural network techniques (ANN). In Bresin (1998) a system that combines symbolic decision rules with ANNs is implemented for sim-

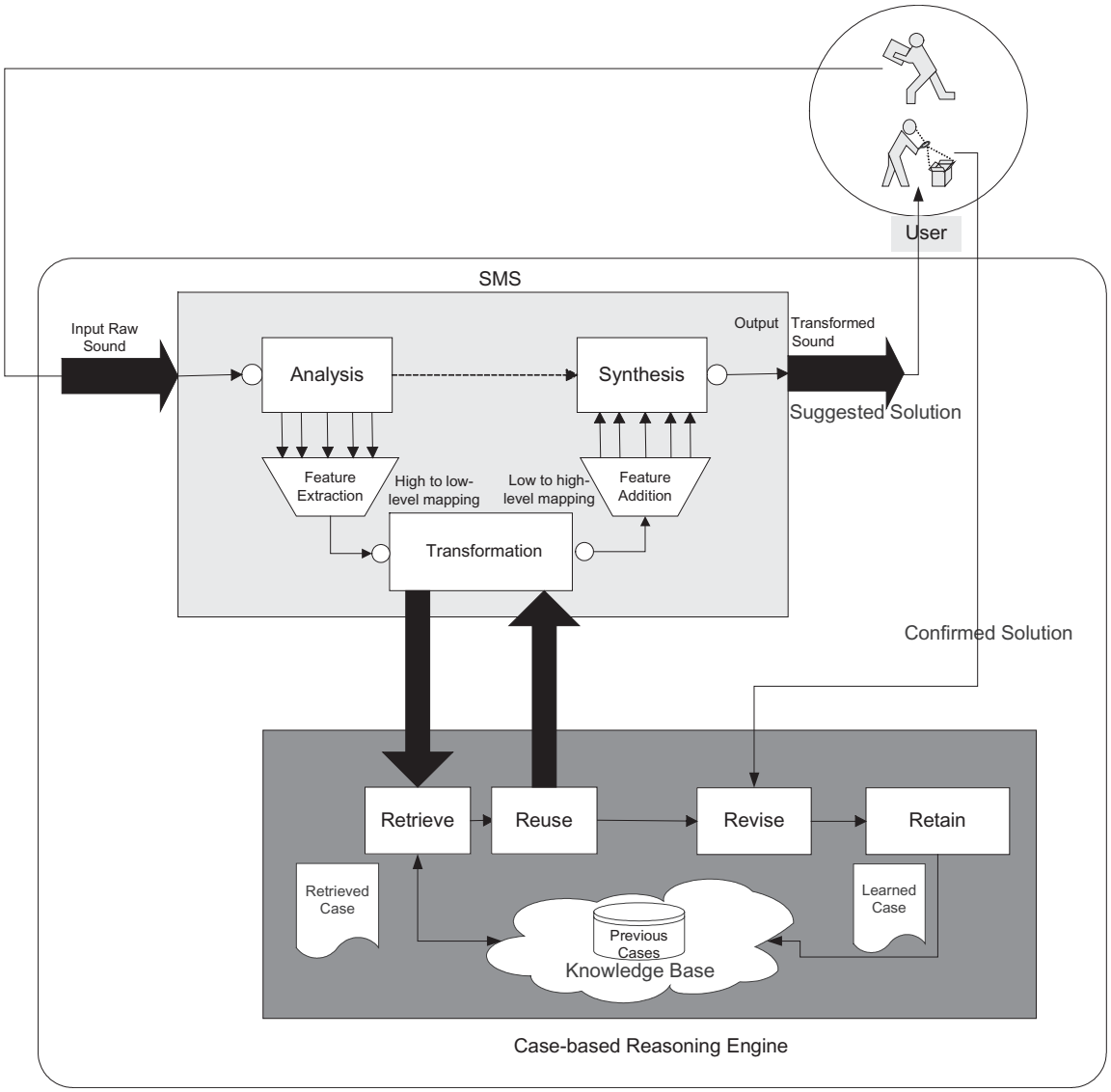

Fig. 17. Saxex, the basic system.

ulating the style of real piano performers. The outputs of the ANNs were expressing time and loudness deviations. For developing the ANNs they extend the standard feed-forward ANN trained with the back propagation algorithm with feedback connections from the output neurons to the input neurons.

Inductive learning and knowledge discovery approach

The work of G. Widmer (Widmer, 2001) is another example of the use of rules for performing transformations of tempo and dynamics. The approach taken in this project was first to record and preprocess a large amount of high-quality musical performances (containing more than 400.000 notes). Then, and using this large amount of examples, they are applying machine learning techniques for inducing performance rules. The project is now producing the first very promising results in the form of collections of induced rules. These rules are not really incorporated in a transformation system yet but they are providing knowledge for designing a performance theory.

6.2 Articulation transformations

There are not many research results on articulation transformations. The work from the University of Padova (De Poli et al., 1998) addresses the articulation goal. From the analysis of the differences between recordings obtained by asking a musician to play the same theme given different adjectives (light, heavy, soft, hard, bright, and dark), they implemented a synthesis module based on a violin physical model. In this work, an analysis-by-synthesis methodology is adopted for determining the acoustic parameters to be controlled.

Another research focused on the analysis of the articulation transformations is presented in Bresin (2000). The paper is mainly centered on the analysis of articulation strategies. It provides experimental results of the analysis of piano recordings played given eight different adjectives (light, heavy, soft, hard, bright, dark, passionate, and flat). In Bresin (2001) these articulation rules are incorporated into the Director Musices system.

The work by R. Danenberg (Dannenberg & Dereny, 1998) is also a good example of articulation transformations for trumpet synthesis. They developed a trumpet synthesizer that combines a physical model with a performance model. The goal of the performance model is to generate control information for the physical model. They performed a collection of controlled recordings for analyzing the trumpet behavior and then, they generalized the results obtained. An important remark of their work was that the phrase synthesis approach, as opposed to of the note-by-note approach, is crucial for obtaining expressive performances. That is, the study of articulation transformations is required.

6.3 SaxEx

The goal of the SaxEx system (Arcos et al., 1998) is to generate expressive melodies by means of performing transformations in five different expressive parameters: dynamics, tempo, vibrato, articulation, and note attacks. For achieving this goal SaxEx works with the analysis features provided by the SMS (Serra, 1998) spectral model. The counterpart of the use of this rich model is that SaxEx is only focused on generating expressive melodies for monophonic instruments in the context of jazz ballads.

The transformation input

The current approach taken in SaxEx for modeling the analysis features provided by the SMS techniques is the use of fuzzy techniques. The advantage of using fuzzy techniques is that we can abstract numerical features provided by a low level analysis by a collection of fuzzy labels with associated membership degrees. This approach facilitates the reasoning process without losing information. For instance, we have divided each expressive parameter with a collection of five fuzzy-sets with labels very-low, low, medium, high, and very high (see Fig. 18). When deciding the value to be applied in a given expressive transformation (SaxEx uses fuzzy sets for combining different alternatives (Arcos & Lopez de Mántaras, 2002).

The approach

One of the problems of musical expressive performance is that a large part of this knowledge is implicit and very difficult to model with a complete theory. For an expert instrument player, it is easier to provide this knowledge by means of playing specific performances (providing examples of good expressive performances). This nature of the problem motivated the Saxex team to follow a case-based reasoning approach. Case-based Reasoning (Aamodt & Plaza, 1994) (CBR) is a recent approach to problem solving and learning where a new problem is solved by finding a set of similar previously solved problems, called cases, and reusing them in the new problem situation. The underlying hypothesis of CBR is that similar problems have similar solutions. Thus, CBR is appropriate for problems where (a) many examples of solved problems can be obtained – like in our case where multiple examples can be easily obtained from recordings of human performances; and (b) a large part of the knowledge involved in the solution of problems is tacit, difficult to verbalize and generalize.

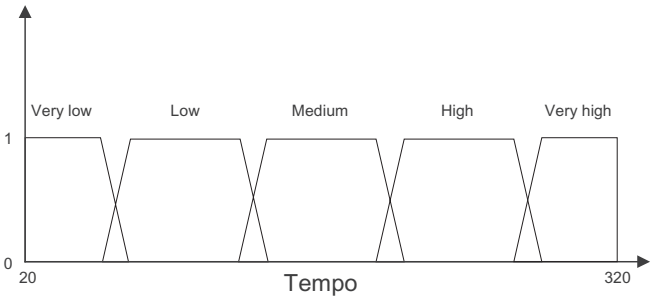

Fig. 18. Linguistic fuzzy values for rubato expressive parameter.

The main subprocesses involved in a CBR system are the retrieval process and the adaptation process. The goal of the retrieval process is to find previously solved problems (cases) "similar" to the current problem. In retrieval, the core decisions are how to define the similarity measures. For instance, given two melodies, how to estimate when they are similar enough. The goal of the adaptation process is to reuse the decisions taken in the cases for solving the new problem. In adaptation, the core decisions are the design of the reusing mechanisms. Finally, incorporating the problems that are solved by a CBR system into its case memory, we are providing learning capabilities to CBR.

The basic system

An input for SaxEx is a musical phrase described by means of SMS analysis features grouped as note regions. For each note region the SMS provides information about durations of attack, release and decay; dynamic envelopes; vibrato levels and non-harmonic signals. Moreover, the user can provide specific qualitative values along three affective dimensions (tender-aggressive, sad-joyful, calm-restless) expressing the user preferences regarding the desired expressive output performance. Affective information can be partially specified, that is, the user does not have to provide values for every dimension. Additionally, information of the harmonic chord sequence in the musical phrase is also included.

The output of the system is a collection of transformation instructions that are sent to the SMS synthesis for generat-

ing an expressive sound transformation. For deciding the transformation instructions Saxex uses a memory of cases of expressive performances previously recorded and coded. For determining what notes are similar to the notes of a given problem, SaxEx is provided by two general theories of musical perception and musical understanding: Narmour's implication/realization (IR) model (Narmour, 1990) and Lerdahl and Jackendoff's generative theory of tonal music (GTTM) (Lerdahl, 1993). Moreover, SaxEx incorporates specific knowledge about Jazz theory. The use of the IR model provides a musical analysis based on the structure of the melodic surface. GTTM, on the other hand, offers a complementary approach to understanding melodies based on a hierarchical structure of musical cognition. The Jazz Theory is introduced in SaxEx for the specific treatment of harmony in jazz. In jazz the notion of tonality is secondary and other aspects such as chord progressions, the tonal functionality of chords, or the use of dominants are more important.

The musical models are the basis for the collection of ten similarity criteria currently implemented in SaxEx. The system provides an initial combination of these criteria but the user can change it. After the retrieval and ranking of the notes more similar to the current problem, the role of the SaxEx adaptation process is to determine the expressive transformations to be applied to each note. That is, for every note in the problem phrase, we have to determine a value for each of the five expressive parameters. To determine these values, the first step is to inspect the solutions given in the precedents (the values chosen in each similar note for each expressive parameter). Because these values are never exactly the same, SaxEx is provided with a collection of reuse criteria that can also be activated and deactivated by the user. An example of a reuse criterion is the "majority rule criterion" that chooses the values of the precedents that belong to the linguistic fuzzy label applied in the majority of precedents.

Having modeled the linguistic values of the expressive parameters by means of fuzzy sets, allows us to apply a fuzzy combination operator to these values of the retrieved notes in the reuse step. The following example describes this combination operation.

Let us assume that the system has retrieved two similar notes whose fuzzy values for the rubato are, respectively, 72 and 190. The system first computes the maximum degree of membership of each one of these two values with respect to the five linguistic values characterizing the *rubato* shown in Figure 18. The maximum membership value of 72 corresponds to the fuzzy value *low* and is 0.90 and that of 190 corresponds to *medium* and is 0.70. Next, it computes a combined fuzzy membership function, based on these two values. This combination consists on the fuzzy disjunction of the fuzzy membership functions *low* and *medium* truncated, respectively, by the 0.90 and 0.70 membership degrees. That is:

$$Max\{\min(0.90, f[low]), \min(0.70, f[medium])\} \quad (17)$$

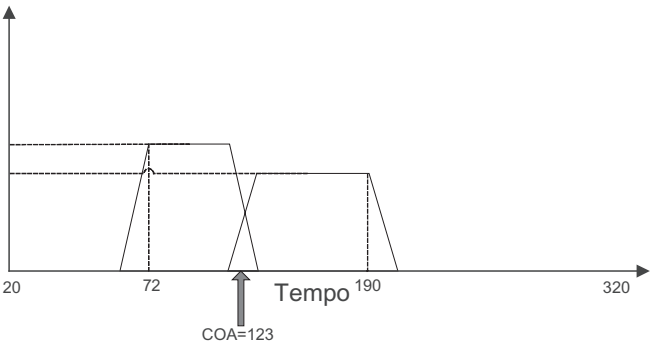

Fig. 19. Fuzzy combination and defuzzification of rubato value.

The result is shown in Figure 19. We finally *defuzzies* this result by computing the COA (Center of Area) of the combined function. The defuzzification step gives the precise value for the tempo to be applied to the initially inexpressive note; in this example the obtained result is 123.

7. Conclusions and future work

We are interested more and more in processing and transforming content and not (only) signal. This approach has already brought up some exciting applications and is likely to be one of the most important future trends. Throughout this article we have tried to give a clear picture of what content-based transformations are, giving examples of sound and music transformations. We have not intended to offer a thorough compilation of already implemented transformation schemes that may address the content level and instead we have concentrated on systems that have been implemented by the authors or are somehow related to our work.

The list of ongoing research and future work on this field could be long enough to make for another article. Suffice it to list the main conceptual lines in which more effort is being put forward:

1. Object Processing. A complex sound mixture may be seen as the sum of different sound objects (mostly, but not only, instruments and sound sources). By analyzing the content of the mix we may be able to process and transform single objects out of the mix (without having to separate them).
2. User-adaptive transformation schemes. Machine learning schemes (such as CBR) can be applied in order to adapt a transformation system to the user preferences. In that way, we may be able to implement more efficient high-level transformations departing from the fact that most high-level labels are not universally accepted concepts and rely much on the user's background.
3. Relating symbolic (musical) and perceptual information to low-level signal data, in both directions (from low level to high level in the analysis step and vice versa in the synthesis stage). This is obviously the most difficult issue that has to be dealt with not only when implementing content-based transformations but also in any kind of content

driven scheme. As in the previous point, Machine Learning techniques are already proving to be a valid approach for descovering these relations.

Acknowledgements

The work reported in this paper has been partially funded by the IST European project CUIDADO and by the TIC national project TABASCO.

References

Aamodt, A., & Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AICom – Artificial Intelligence Communications*, *7*, 39–59.

Amatriain, X., Bonada, J., & Serra, X. (1998). METRIX: A Musical Data Definition Language and Data Structure for a Spectral Modeling Based Synthesizer. In: *Proceedings of the Digital Audio Effects Workshop (DAFX98)*. Barcelona, 1998.

Amatriain, X., Bonada, J., Loscos, A., & Serra, X. (2002). Spectral Processing. In: *DAFX: Digital Audio Effects pp. 373–438*. Udo Zölzer (Ed.). West Sussex, England: John Wiley and Sons, Ltd.

Arcos, J.L., & López de Mántaras, R. (2000). Combining Fuzzy and Case-Based Reasoning to Generate Human-like Music Performances. In: *Information Processing and Management of Uncertainty Congress (IPMU-2000)*, Madrid, 3–7 July 2000. To be published in *Lecture Notes in Artificial Intelligence*. Berlin: Springer Verlag. 2000.

Arcos, J.L., López de Mántaras, R., & Serra, X. (1998). Saxex: A Case-Based Reasoning System for Generating Expressive Musical Performances, *Journal of New Music Research*, *27*, 194–210.

Arfib, D., Keiler, F., & Zölzer, U. (2002). Time Frequency Processing and Source-Filter Processing. In: *DAFX: Digital Audio Effects pp. 237–372*. Udo Zölzer (Ed.), West Sussex, England: John Wiley and Sons, Ltd.

Bemdtsson, G. (1996). The KTH rule system for singing synthesis. *Computer Music Journal*, *20*, 76–91.

Bresin, R. (2001). Articulation Rules for Automatic Music Performance. In: *Proc. of the International Computer Music Conference 2001*, San Francisco: International Computer Music Association.

Bresin, R., & Battel, G.U. (2000). Articulation strategies in expressive piano performance. *Journal of New Music Research*, *29*, 221–224.

Bresin, R. (1998). Artificial neural networks based models for automatic performance of musical scores. *Journal of New Music Research*, *27*, 239–270.

Brown, J.C. (1991). Calculation of a constant Q spectral transform. *Journal of the Acoustical Society of America*, *89*, 425–434.

Bonada, J. (2000). Automatic Technique in Frequency Domain for near-lossless Time-scale Modification of Audio. In: *Proceedings of the 2000 International Computer Music Conference*. San Francisco: Computer Music Association.

Butch Rovan, J., Wanderlay, M., Duvnov, S., & Depalle, P. (1997). Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance. In: Proceedings of *Kansei – The Technology of Emotion Workshop*. Genova – Italia, Oct., 1997.

Camurri, A. (1999). Music content processing and multimedia: Case studies and emerging Applications of intelligent interactive Systems. *Journal of New Music Research*, *28*, 351–363.

Cano, P., Loscos, A., Bonada, J., de Boer, M., & Serra, X. (2000). Voice Morphing System for Impersonating in Karaoke Applications. In: *Proceedings of the 2000 International Computer Music Conference*. San Francisco: Computer Music Association.

Chai, W., & Vercoe, B. (2000). Using User Models in Information Retrieval Systems. In: *Proc. International Symposium on Music Information Retrieval*, Oct. 2000.

Chiariglione, L. (2000). The Value of Content. *Technology Reviews*, p. *79*. Available: http://leonardo.telecomitalia/ab.com/paper/technoreview99/

Dannenberg, R., & Dereny, I. (1998). Combining instrument and performance models for high-quality music synthesis. *Journal of New Music Research*, *27*, 211–238.

De Poli, G., Rodà, A., & Vidolin, A. (1998). Note-by-note analysis of the influence of expressive intentions and musical structure in violin performance. *Journal of New Music Research*, *27*, 293–321.

Di Giura, M., Serina, N., & Rizzotto, G. (1997). Adaptive Fuzzy Filtering for Audio Applications Using a Neuro-Fuzzy Modelization. In: *Proceedings of International Conference on Neural Networks, ICNN'97*.

Dutilleux, P., De Poli, G., & Zöelzer, U. (2002). Time Segment Processing. In: *DAFX: Digital Audio Effects. pp. 201–234.* Udo Zölzer (Ed.). West Sussex, England: John Wiley and Sons, Ltd.

Fletcher, H., & Munson, W.A. (1933). Loudness, its definition, measurement and calculation. *Journal of the Acoustical Society of America*, *5*, 82–108.

Friberg, A. (1995). *A Quantitative Rule System for Musical Performance*. *PhD Dissertation. KTH, Stockholm.*

Gómez, E., Klapuri, A., & Meudic, B. (2002). Melody description and extraction in the context of music content processing. *Journal of New Music Research*, *32*, 23–40.

Hartman, B. (1996). Pitch, periodicity, and auditory organization. *Journal of the Acoustical Society of America*, *100*, 3491–3502.

Karjalainen, M. (1999). Immersion and Content – A Framework for Audio Research. In: *Proceedings of the 1999 IEEE Workshop of Applications of Signal Processing to Audio and Acoustics*.

Krumhansl, C.L. (1989). Why is Timbre so Hard to Understand? *Structure and Perception of electroacoustic sound and music, pp. 43–53. Amsterdam: Elsevier. (Excerpta Medica 846)*.

Lakatos, S. (2000). A common perceptual space for harmonic and percussive timbre. *Perception and Psychophysics*, *62*, 1426–1439.

Laroche, J., & Dolson, M. (1997). About this Phasiness Busi-

ness. In: *Proceedings of International Computer Music Conference*, 1997. San Francisco: International Computer Music Association.

Lerdahl, F., & Jackendoff, R. (1993). An Overview of Hierarchical Structure in Music. In: Stephan M. Schwanaver & David A. Levitt (Eds.), *Machine Models of Music*, *pp. 289–312*. Cambridge, MS: The MIT Press.

Loscos, A., Cano, P., & Bonada, J. (1999). Singing Voice Alignment to Text. In: *Proceedings of the International Computer Music Conference 1999*. San Francisco: International Computer Music Association.

Manjunath, B.S., Salembier, P., & Sikora, T. (Eds.) (2002). *Introduction to MPEG 7: Multimedia Content Description Language.* West Sussex, England: John Wiley and Sons, Ltd.

Martínez, J.M. (2001). Overview of MPEG-7 Standard (version 5.0). *MPEG-7 approved doc. num. ISO/IEC ISO/IEC JTC1/SC29/WG11*. Klangenfurt, July 2002.

McAdams, S., Winsberg, S., Donnadieu, S., De Soete, G., & Krimphoff, J. (1995). Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological Research*, *58*, 177–192.

McAulay, R.J., & Quatieri, T.F. (1986). Speech Analysis/Synthesis based on a Sinusoidal Representation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *34*, 744–754.

Mittal, U., & Phamdo, N. (2000). Signal/Noise KLT Based Approach for Enhancing Speech Degraded by Colored Noise. *IEEE Transactions on Speech Processing and Audio Processing*, *8*.

Moore B., Glasberg B., & Baer, T. (1997). A Model for the prediction of thresholds, loudness, and partial loudness. *Journal of the Audio Engineering Society*, *45*, 224–240.

Narmour, E. (1990). The Analysis and Cognition of Basic Melodic Structures: the Implication-Realization Model. University of Chicago Press.

Pfeiffer, S. (1999). The Importance of Perceptive Adaptation of Sound Features in Audio Content processing. In: *SPIE Storage and Retrieval for Image and Video Databases VII*. January 1999, San Jose, CA, pp. 328–337.

Petters, G., Herrera, P., & Amatriain, X. (1999). Audio CE for Instrument Description (Timbre Similarity). *MPEG 7 Proposal, doc. num. ISO/IEC JTC1/SC29/WG11 M5422*.

Peeters, G., McAdams, S., & Herrera, P. (2000). Instrument Sound Description in the Context of MPEG-7. In: *Proceedings of the 2000 International Computer Music Conference*. San Francisco: Computer Music Association.

Rocchesso, D. (2002). Spatial Effects. In: *DAFX: Digital Audio Effects pp. 137–200*. Udo Zölzer (Ed.). West Sussex, England: John Wiley and Sons, Ltd.

Rodet, X., Depalle, P., & Garcia, G. (1995). New Possibilities in Sound Analysis and Synthesis. In: *Proceedings of Inetrnational Symposium on Musical Acoustics*, IMSA '95.

Rossignol, S., Rodet, X., Soumagne, J., Collette, J., & Depalle, P. (1999). Automatic characterisation of musical signals: Feature extraction and temporal segmentation. *Computer Music Journal*, *28*, 281–295.

Scheirer, E., Väänänen, R., & Huopaniemi, J. (1998). AudioBIFS: The MPEG-4 Standard for Effects Processing. In: *Proceedings of Digital Audio Effects Workshop*. Barcelona.

Schoner, B., Cooper, C., Douglas, C., & Gershenfeld, N. (1998). Data-driven Modeling and Synthesis of Acoustical Instruments. In: *Proceedings of the 1998 International Computer Music Conference*. San Francisco: Computer Music Association.

Serra, X. (1989). *A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition*. Ph.D. Dissertation, Stanford University.

Serra, X., & Smith, J. (1990). Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus Stochastic decomposition. *Computer Music Journal*, *14*, 12–24.

Serra, X., & Bonada, J. (1998). "Sound Transformations Based on the SMS High Level Attributes." *Proceedings of the 98 Digital Audio Effects Workshop*.

Solà, J.M. (1997). *Disseny i Implementació d'un Sintetitzador de Piano*. Graduate Thesis. UPC (Polytechnic University of Catalonia).

Todoroff, T. (2002). Control of Digital Audio Effects. In: *DAFX: Digital Audio Effects*. Udo Zölzer (Ed.). West Sussex, England: John Wiley and Sons, Ltd.

Verma, T.S., & Meng, T.H.Y. (1998). Time Scale Modification Using a Sines + Transients + Noise Signal Model. In: *Proceedings of the Digital Audio Effects Workshop (DAFX98)*, Barcelona, November 1998.

Verfaille, V., & Arfib, D. (2001). A-DAFx: Adaptive Digital Audio Fx. In: *Proceedings of the Digital Audio Effects Workshop (DAFX01)*, Limerick, December 2001.

Vidal, E., & Marzal, A. (1990). A Review and New Approaches for Automatic Segmentation of Speech Signals. In: L. Torres and others (Eds.), *Signal Processing V: Theories and Applications.* Amsterdam: Elsevier Science Publishers.

Wanderley, M., & Battier, M. (Eds.) (2000). M. *Trends in Gestural Control of Music*. Paris: Ircam.

Wessel, D. (1979). Timbre space as a musical control structure. *Computer Music Journal*, *2*, 45–52.

Widmer, G. (2001). The Musical Expression Project: A Challenge for Machine Learning and Knowledge Discovery. In: *Proceedings of the 12th European Conference on Machine Learning (ECML'2001).* Berlin: Springer Verlag.