IMPROVED CONTROL FOR SELECTIVE MINIMIZATION OF MASKING USING INTER-CHANNEL DEPENDANCY EFFECTS

Enrique Perez Gonzalez and Joshua D. Reiss

Centre for Digital Music,
Queen Mary University of London,
Electronic Engineering,
Mile End Road, E1 4NS
London, United Kingdom
enrique.perez@elec.qmul.ac.uk
josh.reiss@elec.qmul.ac.uk

ABSTRACT

A digital audio effect for real time mixing applications, which dynamically adapts to the multi-channel input, has been implemented. The resulting audio mix is the direct result of the analysis of the content of each individual channel with respect to the other channels. The implementation permits the enhancement of a source with respect to the rest of the mixture by selectivity unmasking its spectral content from spectrally related channels. A masking measurement has also been implemented in order to measure the efficiency of the algorithm.

1. INTRODUCTION

Researchers have classified sound effects taxonomically [1] by their preferred implementation; filters, delays, modulators, time-segment processing, time-frequency processing, etc. Similarly, researchers have also classified effects by their perceptual attributes [2] into those which modify timbre, delay, pitch, positions or quality. Although these classifications tend to be accurate in many contexts, they are not optimal for the understanding the signal processing control architectures of some more complex effects. More recently, an Adaptive Digital Audio Effect (ADAFx) class was proposed [3]. This class uses features extracted from the signals to control the signal processing process. One of the most important contributions of the introduction of ADAFx is the proposed categorizing of the control architecture of sound effects. It is the aim of the authors to propose and present in this section a modified and more general classification of this control categorisation. In the context of this paper spectral masking represents the amplitude overlap of the spectrum of audio sources and does not take into account any psychoacoustic masking models.

The most simple control architecture is direct user control devices. These make no use of features extracted from the input signal channels involved, and are therefore non-adaptive. A multi-channel extension of this approach is the result of unifying the user interface, for example when linking a stereo equaliser. This provides exactly the same equalisation for the left and right channel using a single user panel,. Although the user interface is unified, the output signal processing is independent of the signal content.

The second type of control architecture consists of auto-adaptive effects. Here, the control parameter is based on a feature extracted from the input channel. These include, for example, simple single channel noise gates and compressors.

The third type is the external-adaptive effect, which takes its control processing variable from a different channel to the one on which it has been applied. It is called a feedforward external adaptive effect if it takes its control variable from the input, and a feedback external adaptive effect if it takes its control feature from the output. This is the case of ducking effects [4], side chain effects, auto tuning and harmonizers.

A fourth and final type of control architecture are the cross adaptive effects, in which the resulting signal process is the direct result of the analysis of the content of each individual channel with respect to the other channels. These types of effects are commonly intended for program material or mastering. The simplest of them use a single shared control feature that triggers the same processing for all channels. Therefore their signal processing is accomplished by inter-channel dependency. For example a mastering 5.1 compressor can be configured so that all six channels are compressed simultaneously and by the same amount. As a result of this, the effect requires only one of the six channels involved to cross a threshold in order to trigger exactly the same amount of compression for all channels. This type of effect is useful in order to avoid single channel compression that could cause an artefact on the spatial image due to a shift in the loudness. In other words the signal processing applied to each channel is dependent on the signal content of all channels involved.

1.1. Cross-adaptive effects with mixing applications

Complex effects which use multiple inter-channel dependent control variables exist. For example an auto-mixing panner [5] takes the panning decisions based on the spectral content of all the channels involved. These complex cross-adaptive effects are not commonly seen and to the authors' knowledge there is currently no plug-in software dedicated to hosting this type of effects efficiently. This is partly because plug-ins work on a single channel basis and plug-ins do not have a dedicated channel buss for transferring control variables or sharing multiple audio steams within different channels. For this reason, most cross-

adaptive effects are limited to mastering applications. However, given the current predominance of digital mixing boards and sequencing software with flexible architectures, implementing complex cross-adaptive audio effects for non-mastering applications represents a significant opportunity. Such effects can have applications in channel enhancement, adaptive source mixing, aided mixing and even for implementing autonomous mixing tasks.

If we define a monaural mix as the result of combining a group of input channels Ch_n in order to combing them into a single channel mixture, *mix*, where N is the total number of channels in the mixture and n goes from 1 to N, then we can say that the mix is given by the addition of sources given by:

$$\text{mix}=\sum_{n=1}^{N}\text{Ch}_\text{n}(t) \tag{1}$$

Given this model we can generalize a cross-adaptive effect with applications to mixing as the addition of the effect functions, one per channel, in which each effect function is dependent on the available feature vectors extracted from all the channels involved in the audio mix. Assuming there are up to x features per channel, the feature vector fv_x for each channel is obtained by a predetermined real-time feature retrieval algorithm (*FRA*);

$$fv_\text{x}=FRA(Ch_n) \tag{2}$$

$$\text{mix}_\text{fx}=\sum_{n=0}^{N}\text{fv}_\text{xn}\left(Ch_n,fv_1,fv_2,\cdots,fv_{\text{X-1}},fv_\text{X}\right) \tag{3}$$

Where fv_{xn} is the indexed version of fv_x for the N^{th} channel contained within a multiple channel source mixture, MIX_{fx}, to which an adaptive effect has been applied.

Therefore we can say that the resulting signal process applied to each channel is the direct result of the analysis of the content of each individual channel with respect to the other channels.

1.2. Cross-adaptive effects for mixing applications using spectral masking

Spectral masking is a sound artifact which results from the total or partial loss of spectral content perception of one or more channels when they are mixed together. When sources are combined, the content of one source at a given frequency may be low with respect to the other sources in the mix. Thus the listener may not be able to associate that portion of content with its source. Although this obstruction or masking of spectral content has been used as a means of increasing compression ratios of sound files [6], when creating a sound mixture, it is in most cases an undesired artifact because it hides some of the source content, and may render some musical instruments unheard.

Spectral masking for a given source can be measured by obtaining the amount of overlap between the source and the rest of the mix. For a given channel of interest, Ch_m, we can define the spectral masking *SM* of the channel with respect to the rest of the mixture, as follows:

$$\text{SM}=\left(\text{FFT}\left\{\text{Ch}_\text{m}\right\}\right)^2-\left(FFT\left\{\text{mix-Ch}_\text{m}\right\}\right)^2 \tag{4}$$

Where $SM>0$ means the channel is unmasked and $SM\leq 0$ means the channel is masked by the rest of the mix. Equation 4 is dependent on the FFT resolution used. Also, since spectral masking is an amplitude difference measurement it is recommended to compensate for any windowing amplitude artefacts as it might affect the measurements.

The accumulated spectral masking of a source, *ASM*, with respect to the rest of the mix can be obtained by accumulating the result of Eq. 4 over different frames, and is given by:

$$\text{ASM}=\sum_{t=0}^{\infty}\text{SM}_\text{t} \tag{5}$$

While performing audio mixing, one of the reasons for setting different relative levels and different equalization curves is to enhance or de-enhance some of the sources of the mix by reducing the spectral masking. This is a complex task and it requires an understanding of the relationship between the spectral content of the sources and the relative levels among channels.
With this in mind the authors have developed a real-time cross-adaptive channel enhancer that realizes a selective minimisation of spectral masking for control of inter-channel dependency effects. The goal of this effect is to enhance a user selected channel by ensuring it is spectrally unmasked from the rest of the mixture. The method uses full range magnitude adjustments to unmask the source instead of equalization techniques. This facilitates the mixing process, both providing support to professional mixing engineers, and providing a method by which musicians and performers without mixing expertise may still create mixes with minimal masking.

2. IMPLEMENTATION

The cross-adaptive channel enhancement that has been implemented allows the user to enhance a user-selected channel by unmasking it from the rest of the channels. The simple approach to this would be to simply lower the amplitude levels of all other channels with respect to the channel that you want to enhance. This approach is inefficient, due to the fact that it affects all channels, even when the channels are not spectrally related to the channel the user wishes to enhance. A preferred approach, and the one that has been implemented, is to lower the levels of the other channels in proportion to their spectral relationship to the user-selected channel.

2.1. Inter-channel spectral decomposition classification

The first step in the proposed method is the classification of the incoming sources into spectral classes. This process is performed outside of the audible signal-processing path. The implementation is depicted in Figure 1, and is based on an accumulative spectral decomposition classification method presented by the authors in [5].

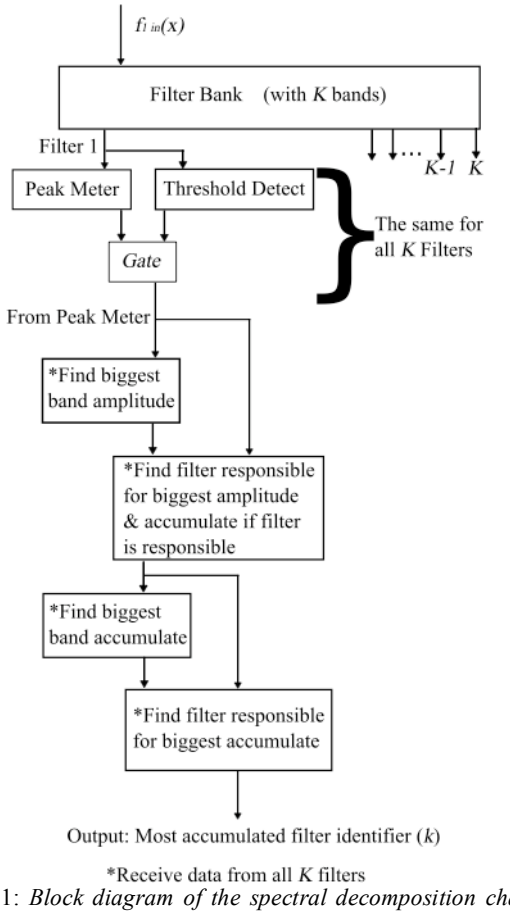

Figure 1: *Block diagram of the spectral decomposition channel categorization algorithm.*

The filter bank has K filters, in which K is equal to the total number of channels (N) being processed by the algorithm. This means that although the filter bank is working on an individual channel basis it expands or contracts dynamically, in proportion to the total number of source channels involved in the cross-adaptive analysis. Therefore each filter contained within the filter bank can have a corresponding k_n value which goes from 1 to N. The filter bank is applied to each input channel and a score related to the maximum peak excitation filter is accumulated and updated every 1ms. The resulting k_n filter gets updated as the input signal changes while the accumulation converges into a stable k_n value. The Accumulative Spectral Decomposition (*ASD*) algorithm categorises every single channel into a k_n class, where the higher the value of n the higher the frequency of the k_n class. Therefore the *ASD* classifier is a function dependent on the signal content of a channel and outputs a spectral feature corresponding to a filter contained within the filter bank.

$$k_{\mathrm{n}} = \mathrm{ASD}(Ch_n) \tag{6}$$

Measurements of the individual filters compromising the filter bank implemented are presented on Figure 2. The combined response of the filter bank is presented on Figure 3, it can be seen that the filter bank boosts the low frequencies while slowly decrementing the gain of the higher frequencies. This approach was taken in other to reduce noise associated with high frequencies.

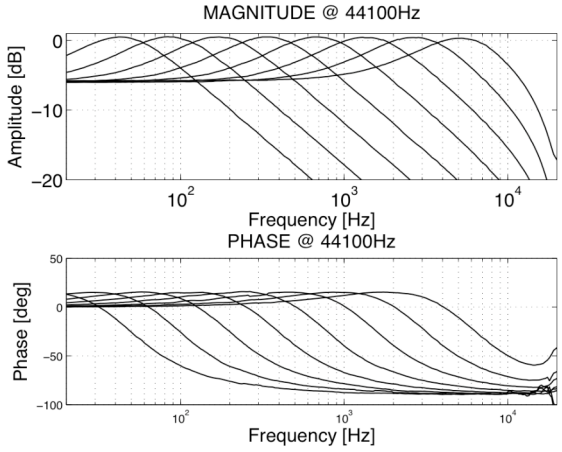

Figure 2: *Magnitude vs. frequency and phase vs. frequency of eight individual filters composing the decomposition filter for a source channel.*

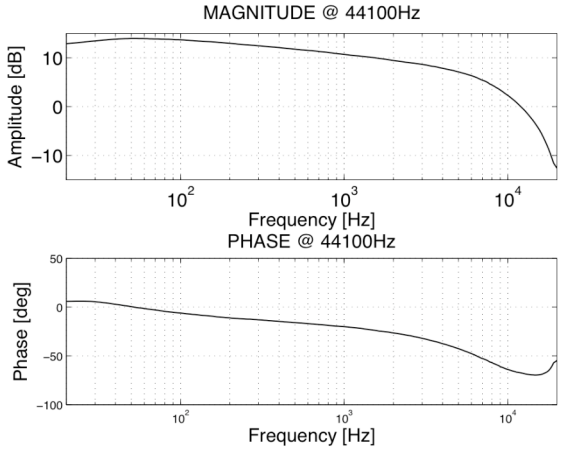

Figure 3: *Magnitude vs. frequency and phase vs. frequency of the combined response of a decomposition filter consisting of eight filters*

2.2. Gaussian dependency.

The second implementation step is to determine a control function which maps the control parameters k_n to the dependency on other channels. For the purpose of nomenclature and implementation we will call the enhanced channel the master channel or Ch_m, with corresponding k_n classification is equal to k_m. Because k_m corresponds to the frequency region of the filter bank that was extracted most consistently, we can assume that Ch_m has most of its spectral content concentrated within that spectral region. Therefore we would like to maximize the attenuation level, A, between the signal level of Ch_m and the rest of the channels which have significant spectral overlap, i.e., share the same k_n classification equal to k_m. On the other hand, we wish to minimize the amount of attenuation, A, for all k_n classifications which have little or no spectral relationship to k_m. This means that for a non-enhanced channel, the further away the k_n classification is from k_m, the less attenuation is required. This calls for a symmet-

ric function of frequency which provides maximal attenuation at the centre frequency of k_m, and smoothly fades to nominal gain as the spectral decomposition classifiers deviate more and more from the value of k_m. This is given by a unitarily normalized Gaussian function, equation 7:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{7}$$

where the Gaussian function *f(x)* is a function of *x*, where *x* represents frequency and μ is the constant which determines the position of the axis of the Gaussian function. σ controls the spread of the Gaussian function and may be given by a user selected variable, *Q*, which directly controls the rate of attenuation for channels with overlapping frequency content. We then proceed to modify *f(x)* to fit the design requirements by performing the following steps.

First we normalize *f(x)* to one, then we obtain the absolute value of its complement, and finally we add a user controllable attenuation variable, *A*. The attenuation variable allows the user to select the amount of attenuation applied at the maximum of the Gaussian function. This is presented in equation 8, where *G(x)* is the inter-channel dependency mapping function.

$$G(x) = \left| \left[A \left(\frac{f(x)}{\frac{1}{\sigma\sqrt{2\pi}}} \right) \right] - 1 \right| \tag{8}$$

Given that we require that the axis of the Gaussian is centered at k_m we must relate μ to k_m. The algorithms has *K* filters comprising the filter bank, where *K* is equal to the number of channels *N*. So k_m must be normalized with respect to *N* in order for μ to be centered exactly at k_m. This normalization is presented in equation 9.

$$\mu = \left[\frac{2}{N-1} (k_m - 1) \right] - 1 \tag{9}$$

Recall that our objective is to enhance the master channel, Ch_m, by reducing the amount of spectral overlap it has with the rest of the mix. So we must keep the master channel gain unchanged while performing a spectrally dependent attenuation to other channels. In other words, the resulting control gain value for the master channel G_m must always be equal to one, while the control gain value for each of the remaining channels, G_n, must be given by evaluating *x* in Eq. 8 with respect to its corresponding k_n spectral classification. Given that the algorithms has a filter bank with *K=N* filters, we must normalize k_n with respect to *N* before evaluating *x*. This normalization is given by equation 10.

$$x = \left[\frac{2}{N-1} (k_n - 1) \right] - 1 \tag{10}$$

A flow diagram of this algorithm is depicted in Figure 4. It can be seen that the five variables needed by the algorithm are:

A) Channel number location: This is the location of the channel querying a control value. It corresponds to the channel to which the control variable result will have a direct effect on its signal processing. This can be automatically obtained from the host and does not require user input.

B) Total number of channels: This corresponds to the overall amount of channels involved in the Cross-Adaptive processing. This variable is user selected, and must be selected at the beginning of the process.

C) Master Channel: This is the channel that the user wishes to enhance. This variable is user selected, and must be selected at the beginning of the process.

D) Attenuation: This is the amount of maximum attenuation applied to sources which are directly related to the spectrum classification of the master channel selected. This variable is user selected.

E) Q: Corresponds to the smoothness quality factor of the Gaussian curve which controls the attenuation spread over the neighbor classified with a different class than the master channel spectral class. This variable is also user selected

2.3. Algorithm applications to enhancement

The algorithm presented in the previous section devises a Gaussian inter-channel dependency value for every channel. It can be used to determine the amount of gain applied to each channel of an audio mixture. This approach ensures minimal spectral masking while affecting the level of the mixed sources in proportion to their spectral relation to the master channel. Such a system would be governed by a cross-adaptive mixing function such as equation 11.

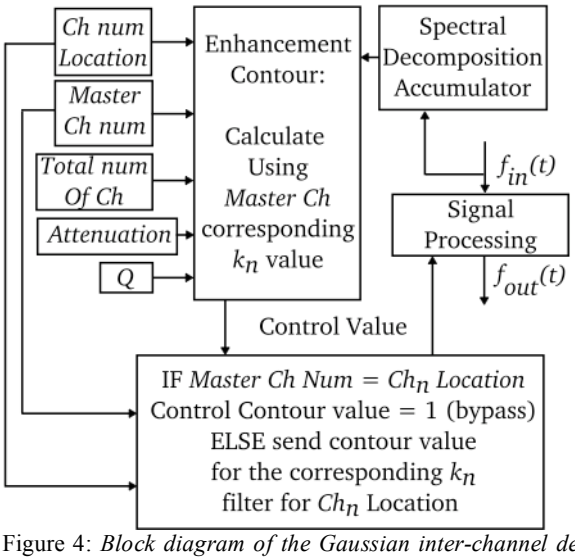

Figure 4: *Block diagram of the Gaussian inter-channel dependency algorithm.*

$$\mathrm{mix_g(t)}=\sum_{n=1}^{N} G_n \mathrm{Ch_n}(t) \tag{11}$$

where $mix_g(t)$ is the overall mix after appling the cross-adaptive effect, G_n is the control value for every channel, Ch_n., and G_n is equal to one for $Ch_n=Ch_m$. n corresponds to every channel involved in the cross-adaptive mixture and takes a value from 1 to N, where N corresponds to the total number of channels involved in the cross-adaptive mixture process. Compared to a system that performs a similar task by using equalization filters, the proposed approach has no channel phase distortion.

Another possible implementation of the algorithm for stereo applications is to reduce directional masking. Directional masking is the equivalent of spectral masking but in the phase domain. Directional masking can be reduced by de-correlating the phase information of the right channel against the left channel. Therefore the more de-correlation the more diffuse the sound, and the more correlated the left and the right channels are, the more present the channel is. By using pseudo-stereo techniques proposed in [7], which split monaural sources and applies all-pass filter networks to the pseudo-left and pseudo-right channels, a stereo effect can be achieved. It is thought that such a effect reduces listening fatigue and enhances the content of the channel to which the pseudo-stereo technique is applied to [8]. The all-pass filter network used for such a method is given by $H_L(y)$ and $H_R(y)$. Using the following equations, we can generate a cross adaptive effect which enhances a target channel by reducing its directional masking:

$$\mathrm{mix_L(t)}=\frac{\sqrt{2}}{2}\sum_{n=1}^{N}\sin(90[1-G_n])H_L(\mathrm{Ch_n}(t))+\cos(90G_n)\mathrm{Ch_n}(t) \tag{12}$$

$$\mathrm{mix_R(t)}=\frac{\sqrt{2}}{2}\sum_{n=1}^{N}\sin(90[1-G_n])H_R(\mathrm{Ch_n}(t))+\cos(90G_n)\mathrm{Ch_n}(t) \tag{13}$$

Given that we desire an implementation of the effect which does not have any effect on the gain, but only on the phase; care has to be taken to ensure that the operations applied ensure unitary gain. First the inter-dependency control variable G_n has been scaled to represent a maximum of 90 degrees and integrated to a sine/cosine law [9] to preserve overall power. Finally a 0.7071 term has been used to preserve the overall power of the constructive interaction of the left and right channel. For this application we must ensure the enhanced target channel does not suffer any diffusion due to the all-pass filter networks. For this reason when $G_m=G_n$, G_n must be equal to one.

2.4. Algorithm Interface

In order for the user to have access to the effect, a graphical user interface was implemented and depicted in figure 5. The user interface is arranged in a standard frequency vs. amplitude plot. The vertical lines show the location of the k_n filters (K=8 on figure 5), the user has access to changing the number of k_n filters shown by changing the amount of channels to which the cross-adaptive effect is to be applied. A plot of the intersection of the Gaussian dependence function, $G(x)$, with the k_n filters is also depicted. The user also has control access for the attenuation and Q of the algorithm. The user can choose the channel to be enhanced, and this automatically sets it as the master channel. The master control interface must be hosted separately of any individual channel host interface, as it is the interface for cross-controlling all channels.

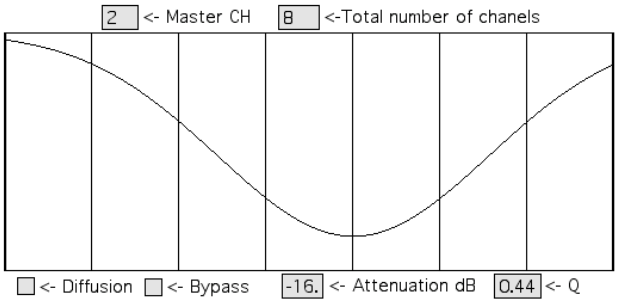

Figure 5: *Master user control interface*

Since there is an actual signal process happening on every channel a small signal processing software device must be contained within each channel. This signal processing device contains a small host interface. This signal processing device is controlled by the inter-dependent variables G_n and in the case of the implementation proposed in this paper, it requires a channel location identifier, which can be automatically assigned by the host. The channel also needs to know if it is a master channel or a slave channel, and this is automatically given by the user selected enhanced channel on the master user interface. Finally, for convenience of the user, a button to call the master user interface from any channel has been included. A depiction of the host interface located on every channel is presented next on figure 6.

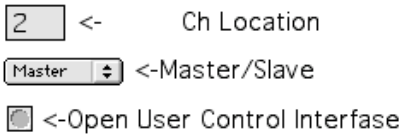

Figure 6: *Host channel interface*

3. RESULTS

For determining the effectiveness of the algorithm, a masking-improvement meter was developed. With the aim of obtaining a perceptual improvement measurement a quantised version of equation 4 was implemented. The quantized implementation was calculated once for the masked index before the effect was applied and once for the masked index after the effect has been applied. All implemented measurements use a 1024 point FFT with no windowing. In order to measure the reduction in spectral masking due to the technique, a simple quantization function was applied to the frequency bins of each frame, *fb*. Quantisation was performed for all bins from 1 to 512 for every given frame. The equations used for implementing such a quantisation are given as follows;

$$\mathrm{UR(fb)}=\begin{cases} 1 if \left(\mathrm{FFT}\{\mathrm{Ch_m}\}\right)^2-\left(FFT\{\mathrm{mix\text{-}Ch_m}\}\right)^2>0 \\ 0 if \left(\mathrm{FFT}\{\mathrm{Ch_m}\}\right)^2-\left(FFT\{\mathrm{mix\text{-}Ch_m}\}\right)^2\le 0 \end{cases} \quad (14)$$

$$\mathrm{UR_{fx}(fb)}=\begin{cases} 1 if \left(\mathrm{FFT}\{\mathrm{Ch_{mfx}}\}\right)^2-\left(FFT\{\mathrm{mix_{fx}\text{-}Ch_{mfx}}\}\right)^2>0 \\ 0 if \left(\mathrm{FFT}\{\mathrm{Ch_{mfx}}\}\right)^2-\left(FFT\{\mathrm{mix_{fx}\text{-}Ch_{mfx}}\}\right)^2\le 0 \end{cases} \quad (15)$$

Where equation 14 corresponds to the quantised calculation of the masking index before the effect was applied and equation 15 corresponds to the quantised calculation of the masking index after the effect was applied.

Finally a perceptual calculation of the quantised unmasked-rate before and after appling the effect was calculated using equations 18 shown next:

$$R_{\%}=\left(\frac{100\sum_{t=0}^{\infty}UR_{\mathrm{fx}}(fb)}{\sum_{t=0}^{\infty}UR(fb)}\right)-100 \quad (16)$$

This implementation gives the perceptual rate difference between the successfully un-masked bins before and after the cross-adaptive effect has been applied. It represents the percentage of masking improvement of using the effect against not using it.

The accumulated masking spectral index for the mixtures before and after applying the effect were depicted as a visual aid based on the implementation of equation 5. The result of this implementation is graphed on figure 7, where all successfully unmasked spectral data has been depicted as falling below the zero crossing threshold. The spectral masking index before applying the effect is depicted in black while the spectral masking index after applying the effect is depicted in grey. A perceptual improvement $R_{\%}(t)$ based on equation 18 is also shown. For the purpose of accuracy all measurements are reset, plotted and re-calculated every time the user changes a parameter in the user interface of the cross adaptive effect.

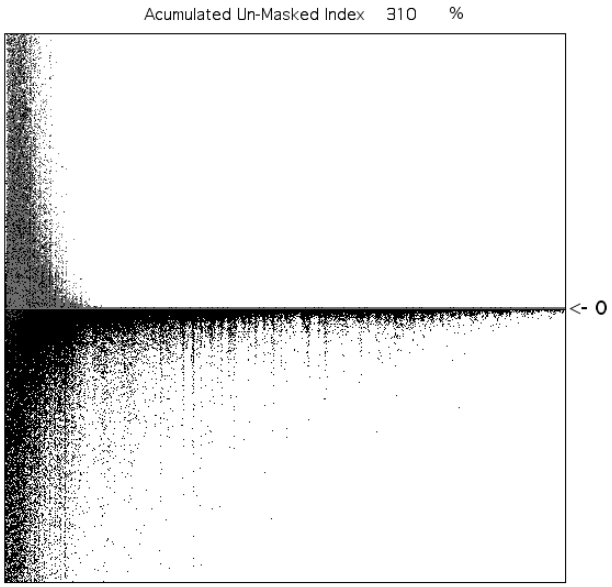

Figure 7: *Accumulated masking index visualization interface*

4. CONCLUSIONS

An effect which uses channel inter-dependency spectral features for enhancement purposes has been implemented. The effect simplifies the complex mixing task of rescaling the levels of multiple sources with respect to their spectral content in order to enhance a source. The user can control the amount of un-masking by changing the Q and attenuation parameters. This controls the inter-channel dependant characteristics of the effect. The effect has a visual and perceptual measurement device that permits quantifying the amount of enhancement applied in terms of the spectral masking improvement. A need for a dedicated cross-adaptive, inter-channel dependency effect host has also been demonstrated.

Future implementations could rely on a similar approach applied to the inter-dependency of the phase relationships between channels while still using spectral features to optimize spectral masking. This would reduce directional masking in a more optimal manner than using spectral features. Integration of a psychoacoustic masking model and the development of an optimal filer bank design for accumulative spectral decomposition feature extraction is still under study.

5. REFERENCES

[1] U. Zölzer, *DAFX Digital Audio Effects*. West Sussex, England: Wiley, 2002.

[2] X. Amatrian, and et al, "Content-based transformations," *Journal of New Music Research,* vol. 32, pp. 95-114, 2003.

[3] V. Verfaille, and et al, "Adaptive Digital Audio Effects (A-DAFx): A New Class of Sound Transformations," *IEEE Transactions On Audio, Speech, and Language Processing,* vol. 1558-7916, 2006.

[4] F. Rumsey, and T. McCormick, "Sound and Recording, an introduction," Focal Press, 2006, p. 302.

[5] E. Perez_Gonzalez, and J. Reiss, "Automatic mixing: live downmixing stereo panner," in *DAFx* Bordeaux-France, 2007.

[6] T. Painter, and A. Spanias, "Perceptual Coding of Digital Audio," *Proceedings of the IEEE,* vol. 88, April 2000.

[7] M. Gerzon, "Signal Processing for Simulating Realistic Stereo Images," in *93rd Convention Audio Engineering Society*, San Francisco,USA, 1992.

[8] P. Mongomery, "Pseudostereo Techniques," Faculty of Architecture, Design and Planning, Uneversity of Sydney, Sydney, Australia 2007.

[9] D. Griesinger, "Stereo and Sorround Panning in Practice," in *112th Audio Engineering Society Convention*, Munich,Germany, 2002.

6. APENDIX

A video demonstration of the implementation can be seen at:
http://web.mac.com/promix_mac/QMUL_EPG/AutoMix_Albums/Pages/Spectral_Enhancer.html