REAL-TIME AUDIO VISUALIZATION WITH REASSIGNED NON-UNIFORM FILTER BANKS

*Zdenek Prusa, Nicki Holighaus**

Acoustics Research Institute,
Austrian Academy of Sciences
Vienna, Austria
{zdenek.prusa,nicki.holighaus}@oeaw.ac.at

ABSTRACT

Filter banks, both uniform and non-uniform, are widely used for signal analysis and processing. However, the application of a time-frequency localized filter inevitably causes some amount of spectral and temporal leakage that, simultaneously, cannot be arbitrarily reduced. Reassignment is a classical procedure to eliminate this leakage in short-time Fourier spectrograms, thereby providing a sharper, more exact time-frequency domain signal representation. The reassignment technique was recently generalized to general filter banks, opening new possibilities for its application in signal analysis and processing. We present here the very first implementation of filter bank reassignment in a real-time analysis setting, more specifically as visualization in a basic audio player application. The visualization provides a low delay moving spectrogram with respect to virtually any time-frequency filter bank by interfacing the C backend of the LTFAT open-source toolbox for time-frequency processing. Low delay is achieved by blockwise processing, implemented with the JUCE C++ Library.

1. INTRODUCTION

Time-frequency representations, be it in the form of the short-time Fourier transform (STFT) [1, 2], windowed MDCT [3] or (non-uniform) filter banks [4, 5], are crucial tools for many signal analysis and processing applications. In particular, their squared magnitude, the *spectrogram* is frequently used to determine the local frequency content of an analyzed signal. The filtering process, by convolution with a time- and frequency-localized filter, introduces spectral and temporal *leakage* in the representation. In other words, a perfectly frequency-(time-)localized signal will be represented with a certain spread over several frequency bands (time positions), depending on the filter shape. This smoothing effect is subject to Heisenberg's uncertainty inequality and thus cannot be arbitrarily reduced.

In an attempt to overcome this smoothing effect, various alternative time-frequency representations have been proposed. The quadratic time-frequency representations in Cohen's class [6, 7], for example, are given by the Wigner-Ville distribution (WVD) [8, 9] convolved with a smoothing kernel. While the spectrogram suffers from a large amount of smoothing, the WVD produces undesirable interference terms. Cohen's class representations allow the choice of a kernel that places them somewhere between these two extrema. Consequently, representations of this kind are often designed with a certain trade-off between smoothing and interference attenuation in mind, e.g. the smoothed pseudo WVD [10] and Born-Jordan distributions.

Non-uniform filter banks can to some degree compensate for the leakage problem by varying the filter shape along frequency, and thereby attempting to locally choose a time-frequency leakage trade-off that least obscures important signal features. This can be achieved either by following a fixed rule or in a signal-dependent fashion. In particular, wavelet filter banks [11, 12] can be constructed through a dilation rule.

Similary, variation along time leads to nonstationary Gabor transforms [13, 14]. Joint time-frequency adaptation provides additional flexibility and has been studied e.g. in [15, 16]. All these methods have in common with the filter bank setting, that they do not reduce the smoothing effect overall, but instead try to select locally a good trade-off between time and frequency smoothing, adapted to signal characteristics.

In contrast, the reassignment method, introduced by Kodera et al. [17] and extended by Auger and Flandrin [18, 19], attempts the deconvolution of the short-time Fourier spectrogram. This is achieved by obtaining *instantaneous frequency* and *group delay* estimates from the partial derivatives of the phase of the STFT, which are subsequently applied to *reassign* time-frequency energy to its supposed point of origin, resulting in a considerably sharpened time-frequency representation without unnecessary interference. In [18], the authors also proposed an efficient means of obtaining the partial phase derivatives, which was recently used to extend the reassignment method to general non-uniform filter banks [20]. The combination of non-uniform filter banks and the reassignment method facilitates the design of highly efficient time-frequency representations with excellent concentration. Therefore, they can be of great use in audio signal analysis, visualization and processing alike.

In this contribution, we recall the results from [20] and present the first *real-time capable implementation* of reassigned filter banks in the form of low delay audio visualization integrated into a basic audio player. The audio player and visualization software rely on the open-source Large Time-Frequency Analysis Toolbox (LTFAT)[1], specifically on its C backend, and the JUCE C++ Library[2].

Organization of the paper: The next section recalls the theoretical background behind the reassignment method for general filter banks. In particular, we show how to efficiently obtain the reassigned filter bank representation from the original filter bank coefficients by means of two additional filter bank analyses, specifically designed for the task. Section 2.2 recalls the derivation of the

* This work was supported by the Austrian Science Fund (FWF) START-project FLAME ("Frames and Linear Operators for Acoustical Modeling and Parameter Estimation"; Y 551-N13).

[1] http://ltfat.github.io
[2] http://www.juce.com

general reassignment operators from the short-time Fourier case. Section 3 describes the user experience and functionality of the player during runtime, while Section 4 provides some technical information about the implementation. Finally, the manuscript concludes with a short discussion of possible future work.

2. PRELIMINARIES

Although the previous publications on reassignment were developed for continuous time signals, see references in Section 1, we will present the theory for finite energy sequences $f \in \ell^2(\mathbb{Z})$ to reflect the actual implementation. We denote the DTFT of f as $\widehat{f}(\omega) = \sum_{l\in\mathbb{Z}} f(l)e^{-2\pi i\omega l}$ and the set of nonnegative integers strictly smaller than $K > 0$ by $\mathbb{Z}_K$.

For a signal $f \in \ell^2(\mathbb{Z})$, its STFT with respect to the window $g \in \ell^2(\mathbb{Z})$ is given by

$$\begin{aligned} V_g f(x,\omega) &= \langle f, g_{x,\omega}\rangle \\ &= \sqrt{\mathcal{S}_g f(x,\omega)}e^{2\pi i\phi(x,\omega)}, \end{aligned} \tag{1}$$

where $g_{x,\omega}[l] = e^{2\pi i\omega(l-x)}g[l-x]$, $\mathcal{S}_g f = |\mathcal{V}_g f|^2$ is the spectrogram and $\phi : \mathbb{Z}\times\mathbb{T} \to \mathbb{R}$ is the phase of the STFT.

Let $\boldsymbol{g} = \{g_k \in \ell^2(\mathbb{Z})\}_{k\in\mathbb{Z}_K}$, $\boldsymbol{a} = \{a_k \in \mathbb{N}\}_{k\in\mathbb{Z}_K}$ a sequence of functions and *decimation factors*, respectively. We call the system $\left\{\overline{g_k[na_k - \cdot]}\right\}_{n,k\in\mathbb{Z}}$ a *(analysis) filter bank (FB)*. The associated K-channel *filter bank analysis* is given by

$$c_{n,k} := c_f[n,k] := \langle f, \overline{g_k[na_k - \cdot]}\rangle. \tag{2}$$

A filter bank forms a *frame*, if there are constants $0 < A \le B < \infty$, such that $A\|f\|_2^2 \le \|c\|^2 \le B\|f\|_2^2$, for all $f \in \ell^2(\mathbb{Z})$. The frame property guarantees the stable invertibility of the coefficient mapping by means of a *dual frame* $\{\widetilde{g_{n,k}}\}_{n\in\mathbb{Z},k\in\mathbb{Z}_K}$, i.e.

$$f = \sum_{n,k} c_{n,k}\widetilde{g_{n,k}}, \text{ for all } f \in \ell^2(\mathbb{Z}). \tag{3}$$

2.1. Reassignment for filter banks

In [20] we show that reassignment can be applied to time-frequency FBs. For that purpose denote by ω_k the center frequency[3] of g_k. Define

$$g_k^T[l] := lg_k[l],\ g_k^F := e^{2\pi i\omega_k l}(e^{-2\pi i\omega_k l}g_k)', \tag{4}$$

for all $l \in \mathbb{Z}$, $k \in \mathbb{Z}_K$. Here f' denotes any suitable discrete derivative of f. We can write

$$c_f^T[n,k] = \langle f, \overline{g_k^T[na_k - \cdot]}\rangle,\ c_f^F[n,k] = \langle f, \overline{g_k^F[na_k - \cdot]}\rangle. \tag{5}$$

Moreover, we obtain the estimated true time position in samples

$$\boldsymbol{x}_0(na_k,\omega_k) = na_k + \mathbf{Re}\left(c_f^T[n,k]/c_f[n,k]\right), \tag{6}$$

whenever $c_f[n,k] \neq 0$ and similarly the estimated true normalized frequency position

$$\boldsymbol{\omega}_0(na_k,\omega_k) = \omega_k - \mathbf{Im}\left(c_f^F[n,k]/c_f[n,k]\right). \tag{7}$$

[3]For the reassignment procedure to make sense, we assume that g_k is well-localized around time 0 and $\widehat{g_k}$ is well-localized around frequency ω_k with $\omega_j < \omega_k$ if $j < k$ and that the points ω_k, $k \in \mathbb{Z}_K$ are well-distributed on the torus.

We might prefer the reassigned representation to be defined on the sampling points (na_k, ω_k). This leads to the *reassigned filter bank (RFB)* defined as

$$c_f^{\mathcal{R}}[n,k] := \sum_{(m,l)\in L_{n,k}} |c_f[m,l]|^2, \tag{8}$$

where

$$L_{n,k} := \{(m,l) \in \mathbb{Z}^2 \ :\ (\boldsymbol{n}_0[m,l], \boldsymbol{k}_0[m,l]) = (n,k)\}, \tag{9}$$

and

$$\boldsymbol{k}_0[m,l] := \arg\min_{k\in\mathbb{Z}_K} |\omega_k - \boldsymbol{\omega}_0(la_k,\omega_k)| \text{ and} \tag{10}$$

$$\boldsymbol{n}_0[m,l] := \left\lfloor \frac{\boldsymbol{x}_0(la_k,x_k)}{a_{k_0(m,l)}} \right\rceil. \tag{11}$$

Note that in this setup, frequency reassignment has priority over time reassignment.

2.2. Derivation of the reassignment operators

The reassignment operators for the filter bank case are in fact easily derived from known expressions for the reassignment operators for the short-time Fourier transform. To see that, consider a function $g \in \ell^2(\mathbb{Z})$, well-localized around $x \in \mathbb{Z}$ with its Fourier transform $\hat{g}$ well-localized around $\omega \in \mathbb{T}$. Then

$$\tilde{g}[l] := e^{-2\pi i\omega l}g[l+x] \tag{12}$$

is well-localized around 0 in time and frequency and we have

$$V_{\tilde{g}} f(x,\omega) = \langle f, \tilde{g}_{x,\omega}\rangle = \langle f, g\rangle. \tag{13}$$

For the short-time Fourier transform $V_{\tilde{g}}f$, the reassignment operators are derived from the instantaneous frequency and group delay and defined as $\boldsymbol{x}_0(x,\omega) = x - \frac{\partial}{\partial\omega}\phi(x,\omega)$ and $\boldsymbol{\omega_0}(x,\omega) = \frac{\partial}{\partial x}\phi(x,\omega)$, where ϕ is the phase of $V_{\tilde{g}}f$ as in Eq. (1). In contrast to the continuous case $f, g \in \mathbf{L}^2(\mathbb{R})$ usually present in the literature, the derivative $\frac{\partial}{\partial x}$ over $x \in \mathbb{Z}$ depends on a chosen convention for discrete derivatives. In practice however, different discrete derivatives provide similar reassignment results.

Auger and Flandrin [18] have shown that the reassignment operators can be expressed as the pointwise product of the STFTs with respect to 3 window functions, depending on the window $\tilde{g}$ and this is true for the discrete signals as well. We obtain:

$$\begin{aligned} \boldsymbol{x}_0(x,\omega) &= x + \mathbf{Re}\left(\mathcal{V}_{\tilde{g}_T} f(x,\omega)/\mathcal{V}_{\tilde{g}} f(x,\omega)\right) \\ &= x + \mathbf{Re}\left(\frac{\mathcal{V}_{\tilde{g}_T} f(x,\omega)\overline{\mathcal{V}_{\tilde{g}} f(x,\omega)}}{\mathcal{S}_{\tilde{g}} f(x,\omega)}\right) \end{aligned} \tag{14}$$

and

$$\begin{aligned} \boldsymbol{\omega_0}(x,\omega) &= \omega - \mathbf{Im}\left(\mathcal{V}_{\tilde{g}'} f(x,\omega)/\mathcal{V}_{\tilde{g}} f(x,\omega)\right) \\ &= \omega - \mathbf{Im}\left(\frac{\mathcal{V}_{\tilde{g}'} f(x,\omega)\overline{\mathcal{V}_{\tilde{g}} f(x,\omega)}}{\mathcal{S}_{\tilde{g}} f(x,\omega)}\right), \end{aligned} \tag{15}$$

whenever $\mathcal{S}_{\tilde{g}} f(x,\omega) \neq 0$ and 0 else. Here, $\tilde{g}'[l] = \frac{\partial}{\partial l}\tilde{g}[l]$ is the discrete derivative and $\tilde{g}_T[l] := l\tilde{g}[l]$ is a time weighted version of $\tilde{g}$.

Note that differentiation is a translation-invariant operator and time weighting is invariant under modulation, i.e. multiplication

with $e^{2\pi i\xi l}$ for $\xi \in \mathbb{T}$. Therefore, with $g^T[l] := [l-x]g[l]$ and $g^F(t) := e^{2\pi i\omega l}(e^{-2\pi i\omega l}g)'[l]$, we obtain

$$\mathcal{V}_{\tilde{g}_T} f(x,\omega) = \langle f, g^T \rangle, \text{ and } \mathcal{V}_{\tilde{g}'} f(x,\omega) = \langle f, g^F \rangle. \quad (16)$$

Given a filter bank or continuous filter bank, simply set $g = \overline{g_k[-\cdot]}$ to obtain Eqs. (6) and (7).

2.3. Filter bank choice

The reassignment method works optimally, only if the chosen filter bank is already able to provide some (phase space/time-frequency) separation of the individual signal components. Although RM is able to improve the separation, and localization, of time-freuqency components, if this is not the case, there are clear limits to this. If, for example, a coefficient $c_f[n,k]$ contains equal amounts of energy from two signal components centered at frequencies $\omega_1 \neq \omega_2$, then it cannot be expected that the estimated true frequency position $\boldsymbol{\omega}_0(na_k, \omega_k)$ provides a meaningful value. In practice, it will usually point at some frequency in $[\omega_1, \omega_2]$, at least not degrading localization.

Consequently, RM can only provide optimal performance when combined with a filter bank that is adapted to the signal class at hand. In the setting of complex audio signals, this usually requires good frequency localization in a band of low frequencies and, for increasing frequency, a gradual increase localization in time resolution. Therefore, filter banks adapted to perceptually-motivated frequency scales such as Bark, ERB or Mel [21, 22, 23, 24], or constant-Q filter banks [25, 26, 27, 28], can be expected to perform well.

3. DESCRIPTION OF THE AUDIO PLAYER AND VISUALIZATION

The main purpose of the proposed audio player application is to provide a body and interface for the visualization of audio content by means of a low delay spectro-temporal filter bank representation, with emphasis on reassigned filter banks. Nonetheless, it provides most functionality expected of a basic audio player such as playback control (*play, pause, stop*), loading several files into a playlist, switching between tracks in the list and automatic looping of the full playlist or single tracks. The basic interface, consisting of the toolbar below the moving spectrogram in the main player window, as well as the playlist window, should be familiar to anyone that has used one of the countless media players available. For an illustration and detailed explanation, see Figures 1 and 2.

In addition to the files in the playlist, the player provides a temporary slot for loading a single file without adding it to the list. This is done by selecting *Open File* → *Load Audio file* from the top menu bar.

Audio devices can be selected through *Options* → *Audio Settings*. If an input source is selected, the *source selector* allows on-the-fly switching between file playback and the input source.

The *main features*, however, are the ability to switch between the original and reassigned filter bank representations by the press of the *reassignment toggle* and loading of alternative analysis filter banks during runtime, through the *load filter bank* button, see Figure 1. The filterbanks can be generated using LTFAT in MATLAB/GNU Octave and exported in a binary format using script `write_filterbank_bin.m` included in the plugin source code archive, see Section 4. The binary file contains the generated filterbank g_k, a_k together with g_k^T and g_k^F for $k = 0, \ldots, K-1$

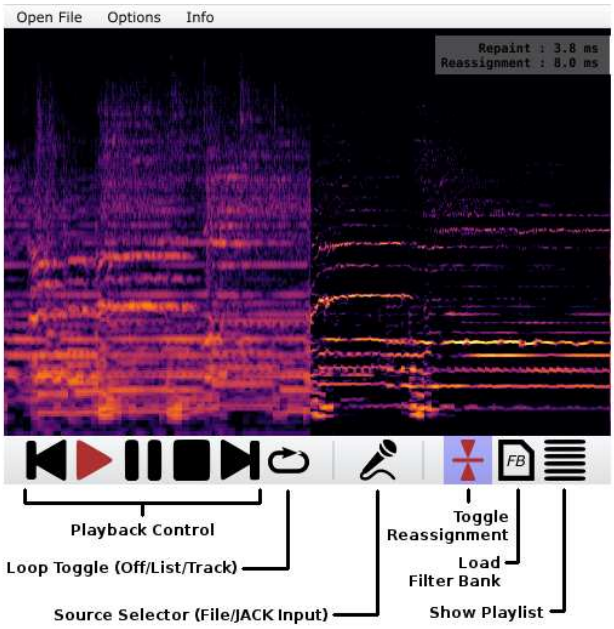

Figure 1: Screenshot of the audio player interface, showing (left) the non-reassigned spectrogram and the reassigned spectrogram (right). The main player window performs the usual playback control tasks through the 5 playback control buttons. The *loop toggle* can be used to choose between (not) looping the playlist or looping the current track. The *source selector* switches between file input (default) and input through the JACK input stream (only available if JACK is selected in the audio settings). On-the-fly toggling between the standard and reassigned spectrogram is enabled by the *reassignment toggle*, while the last two buttons allow loading an alternative filter bank exported from MATLAB through the provided `write_filterbank_bin.m` and showing the playlist.

for a fixed buffer length. Currently, LTFAT contains the following band-limited perfect-reconstruction filterbank generating routines:

- `erbfilters` – Equivalent rectangular bandwidth filterbank [23, 24]
- `cqtfilters` – Constant-Q filterbank [27]
- `warpedfilters` – General frequency warped filterbanks [29]
- `audfilters` – Band-limited filters adapted to auditory scales [24]

Custom filterbanks can be used provided they conform to LTFAT filterbank format.

Finally, the dynamic range of the spectrogram visualization can be varied by a slider, available on right-clicking the spectrogram element, see Figure 3.

4. TECHNICAL BACKGROUND OF THE IMPLEMENTATION

Although the reassignment procedure was defined for general filterbanks, in our contribution, we use *band-limited* filterbanks with possibly rational subsampling rates (that is non-integer a_k) which

Figure 2: Screenshot of the playlist element. The playlist body shows all the loaded files, with the currently played file highlighted in blue. Individual items can be selected by a single click on the entry. Multiple entries are selected by additionally pressing the Ctrl or Shift buttons, similar to standard window manager behavior. Double-clicking starts playback of the clicked file, while the $+$ and $-$ buttons on the bottom of the window allow for adding files to the playlist or removing them. If the playlist is closed, pressing the *playlist button* in the main window makes it reappear.

enables an efficient implementation. More precisely, for the filtering, we follow the computational framework from [28] (using the phase convention introduced later in [30]) which was derived for constant-Q transform, but it applies to general band-limited filterbanks as well. By default, we use ERBlets [24] with 510 filters with rational subsampling rates and buffer length $L = 4096$ samples. The decimation factors are chosen to provide the least redundant, perfect reconstruction system with no aliasing in the subbands, given the fixed ERBlet filters. The frequency responses are shown in Fig. 4.

4.1. Dependencies

The player was implemented using the C++ library JUCE [31] (version 4.1.0), the implementation of the filterbank and the reassignment was linked from the C backend library[4] of LTFAT [32] which in turn uses FFTW library (version 3.3.4) [33] for the FFT calculations.

All involved libraries are licensed under GPL, therefore our program is licensed under terms of GPLv3 and its source codes are freely available[5].

The program was primarily developed on Linux, but it also compiles and runs on Windows. The interested reader can find the *Introjucer* project template included with the source code.

[4] `http://github.com/ltfat/libltfat`

[5] `http://ltfat.github.io/related/reassignment`

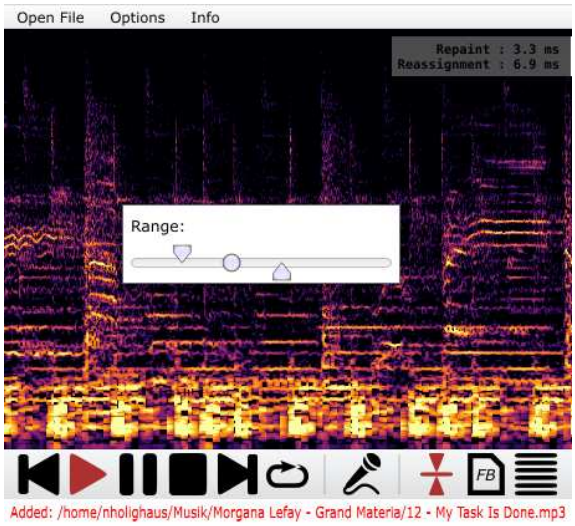

Figure 3: The dynamic range of the moving spectrogram can be conveniently modified through the use of the *dynamic range slider* appearing on right-click on the spectrogram.

4.2. Implementation

The input stream of audio data blocks is rebuffered according to the filterbank buffer length L with half buffer length overlap. For that purpose we employ a lock-free circular buffer. On the audio thread, the data is only written to it as they arrive. The circular buffer is repeatedly polled for new data in regular intervals ($1/60$ s in our implementation) by a separate thread. When available, L samples are read to a working buffer, but the circular buffer read index is advanced only by $L/2$ samples.

Next, the buffer is weighted with a Hann window such that sum of the shifted windows equals constant 1.

Three sets of filterbank coefficient subbands are computed using the chosen filterbank g_k, a_k $(k = 0, \dots, K-1)$ and its two derived versions[6] g_k^T and g_k^F. Each filter is defined by its frequency response being nonzero only in the range $[b_{\min,k}, b_{\max,k}]$ that is for $B_k = b_{\max,k} - b_{\min,k} + 1$ frequency bins. The bandwidth B_k is common for all three versions of k-th filter. Each filter produces a subband consisting of $N_k = L/a_k$ coefficients. In the following, we always assume that there is no aliasing in the resulting subbands; therefore $N_k \geq B$ and a_k is allowed to be a rational number.

Formally written, the three sets of coefficient subbands $c_{f_w,k}$, $c^T_{f_w,k}$ and $c^F_{f_w,k}$ for a single buffer are obtained as follows (in pseudo Matlab code):

1. Load buffer f of length L.
2. $f_w = w. * f$, where w is length L Hann window.
3. $F_w = \mathrm{FFT}_L(f_w)$

[6] Our implementation of g_k^F relies on the spectral derivative, i.e. the filter is obtained by multiplying the frequency response of g_k with $(\omega - \omega_k)$, where ω_k is the center frequency of g_k. As long as g_k and its Fourier transform $\widehat{g_k}$ are concentrated, we have not observed any meaningful difference in the reassignment result when substituting the spectral derivative with any other suitable discrete derivative. The reason for us choosing this particular convention is simply that g_k^F is obtained by the same procedure as g_k^T, applied in the Fourier domain.

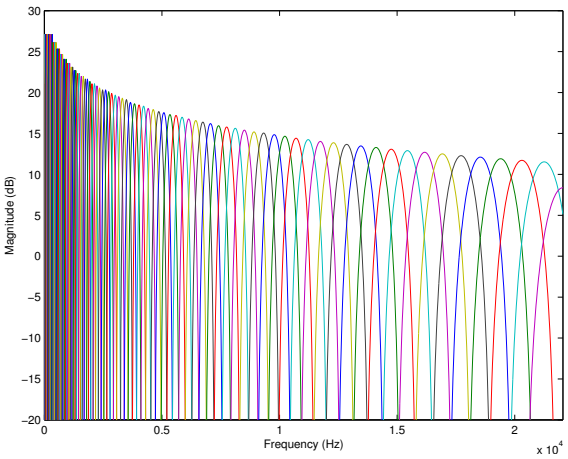

Figure 4: Frequency responses of the default filterbank. Every fifth filter is shown.

4. For $k = 0, \ldots, K-1$ do

   (a) $C_{f_w,k} = F_w . * g_k$ reduced to range $[b_{\min,k}, b_{\max,k}]$

   (b) Extend $C_{f_w,k}$ with zeros to length N_k

   (c) $c_{f_w,k} = \text{IFFT}_{N_k}(\text{circshift}_{N_k}(C_{f_w,k}, -b_{\min,k}))$

5. Repeat loop 4 for g_k^T and for g_k^F to obtain $c_{f_w,k}^T$ and $c_{f_w,k}^F$.

If subband aliasing is present, i.e. $N_k < B$ for some k, then step 4(b) needs to be modified to *fold* C_{f_w} to the appropriate length N_k, but the remainder of this section remains valid.

Before applying the reassignment procedure, we use $c_{f_w,k}$, $c_{f_w,k}^T$ and $c_{f_w,k}^F$ from the current and from the previous buffers to approximate the corresponding true coefficients c_f, c_f^T, c_f^F at the appropriate buffer location. To achieve that, the first half of each subband k from the current buffer is element-wise added to the second half of the corresponding subband from the previous buffer.

The last step is obtaining the reassigned coefficients from (8) using the just computed approximation of c_f, c_f^T, c_f^F.

The overall delay, that is the delay between the instant the circular buffer has L samples available and the instant the data actually appears on the screen, consists of several contributions. The constant part comes from the overlapping of subbands and it is equal to $L/2$ samples. Additional delay is introduced by the computation itself and by the time required to repaint the screen. Sum of these values must be less than $L/2$ times the sampling rate, otherwise the program starts skipping samples. Due to the multithreaded nature of the program, the periodic polling of the circular buffer introduces jitter equal to the request period.

5. CONCLUSION AND OUTLOOK

In this contribution, we discussed the application of the reassignment method to general filter banks. We provided a low delay implementation in a basic audio player that uses the reassigned coefficients for real-time visualization.

In [20], the authors have shown that the reassigned representation can be used as an interface for time-frequency processing, by relying on the *inverse reassignment map*

$$\mathbf{iR}[n,k] = \{(m,l) \in \mathbb{Z} \times \mathbb{Z}_K \ : \ (n_0[m,l], k_0[m,l]) = (n,k)\},$$

which serves as a lookup table during processing. In other words, the user selects a time-frequency region $\Omega \subset \mathbb{Z} \times \mathbb{Z}_K$ in the reassigned spectrogram to be processed and decides on a processing operation. That processing operation is then applied automatically to the filter bank coefficients $c_f[\mathbf{iR}[\Omega]]$. If the filter bank forms a frame, then the modified coefficients can be synthesized to obtain a processed signal.

Simple processing capabilities could be easily added to the current audio player by providing a *brush tool* similar to image processing applications. Modifications could be applied by *painting* over the moving spectrogram, or by pausing playback and painting in a stationary image of the currently displayed audio segment. By implementing a synthesis filter bank, operating in the background, the user could switch between playback the incoming audio stream on the *right end* of the visualization and the, possibly modified outgoing audio stream on the *left end*.

Instead of loading separate filter banks individually, a single file, containing a number of filter banks to switch between on-the-fly would make changing between representations more convenient. In fact, this type of functionality is already partially implemented

In applications where slightly longer delay is acceptable, block processing scheme could be modified to more closely resemble the framework proposed in [28]. This would reduce blocking artifacts, especially when short blocks and filters with very narrow frequency response are considered.

6. REFERENCES

[1] D. Gabor, "Theory of communication," *J. IEE*, vol. 93, no. 26, pp. 429–457, 1946.

[2] Karlheinz Gröchenig, *Foundations of Time-Frequency Analysis*, Appl. Numer. Harmon. Anal. Birkhäuser, Boston, MA, 2001.

[3] Henrique Malvar, *Signal Processing with Lapped Transforms*, Boston, MA: Artech House. xvi, 1992.

[4] Martin Vetterli, "Filter banks allowing perfect reconstruction," *Signal Process.*, vol. 10, pp. 219–244, 1986.

[5] Sony Akkarakaran and P. P. Vaidyanathan, "Nonuniform filter banks: new results and open problems," in *Beyond wavelets*, vol. 10 of *Studies in Computational Mathematics*, pp. 259–301. Elsevier, 2003.

[6] Leon Cohen, "Time-frequency distributions - a review," *Proc. IEEE*, vol. 77, no. 7, pp. 941–981, 1989.

[7] Leon Cohen, *Time-Frequency Analysis: Theory and Applications.*, Prentice Hall Signal Processing Series. Prentice Hall, 1995.

[8] Eugene P. Wigner, "On the quantum correction for thermodynamic equilibrium.," *Phys. Rev., II. Ser.*, vol. 40, pp. 749–759, 1932.

[9] J de Ville et al., "Théorie et applications de la notion de signal analytique," *Cables et transmission*, vol. 2, no. 1, pp. 61–74, 1948.

[10] Wolfgang Martin and Patrick Flandrin, "Wigner-ville spectral analysis of nonstationary processes," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 33, no. 6, pp. 1461–1470, 1985.

[11] Martin Vetterli and Jelena Kovacevic, *Wavelets and subband coding*, vol. 87, Prentice Hall PTR Englewood Cliffs, New Jersey, 1995.

[12] Stéphane Mallat, *A wavelet tour of signal processing: The sparse way*, Academic Press, Third edition, 2009.

[13] Peter Balazs, Monika Dörfler, Florent Jaillet, Nicki Holighaus, and Gino Angelo Velasco, "Theory, implementation and applications of nonstationary Gabor frames," *J. Comput. Appl. Math.*, vol. 236, no. 6, pp. 1481–1496, 2011.

[14] Marco Liuni, Peter Balazs, and Axel Röbel, "Sound Analysis and Synthesis Adaptive in Time and Two Frequency Bands," in *Proc. of the 14th Int. Conference on Digital Audio Effects (DAFx-11), Paris, France, September 19-23*, September 2011.

[15] Cormac Herley, Jelena Kovacevic, Kannan Ramchandran, and Martin Vetterli, "Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3341–3359, 1993.

[16] Monika Dörfler, "Quilted Gabor frames - A new concept for adaptive time-frequency representation," *Adv. in Appl. Math.*, vol. 47, no. 4, pp. 668–687, Oct. 2011.

[17] Kunihiko Kodera, Claude De Villedary, and Roger Gendrin, "A new method for the numerical analysis of non-stationary signals," *Physics of the Earth and Planetary Interiors*, vol. 12, no. 2, pp. 142–150, 1976.

[18] F. Auger and P. Flandrin, "Improving the readability of time-frequency and time-scale representations by the reassignment method," *Signal Processing, IEEE Transactions on*, vol. 43, no. 5, pp. 1068 –1089, may 1995.

[19] Eric Chassande Mottin, Ingrid Daubechies, Francois Auger, and Patrick Flandrin, "Differential reassignment," *IEEE Signal Processing Letters*, vol. 4, no. 10, pp. 293 –294, oct 1997.

[20] Nicki Holighaus, Zdenek Prusa, and Peter L. Søndergaard, "Reassignment and synchrosqueezing for general time-frequency filter banks, subsampling and processing," *Signal Processing*, pp. 1–8, Aug. 2016.

[21] B. R. Glasberg and B. C. J. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hear. Res.*, vol. 47, pp. 103–138, 1990.

[22] V. Hohmann, "Frequency analysis and synthesis using a gammatone filterbank," *Acta Acust. united Ac.*, vol. 88, no. 3, pp. 433–442, 2002.

[23] Thibaud Necciari, Peter Balazs, Nicki Holighaus, and Peter Søndergaard, "The ERBlet transform: An auditory-based time-frequency representation with perfect reconstruction," in *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, 2013, pp. 498–502.

[24] Thibaud Necciari, Nicki Holighaus, Peter Balazs, and Zdenek Prusa, "A perceptually motivated filter bank with perfect reconstruction for audio signal processing," *submitted, preprint available: http://arxiv.org/abs/1601.06652*, 2015.

[25] Judith Brown, "Calculation of a constant Q spectral transform," *J. Acoust. Soc. Amer.*, vol. 89, no. 1, pp. 425–434, 1991.

[26] Christian Schörkhuber and Anssi Klapuri, "Constant-Q toolbox for music processing," in *Proceedings of the 7th Sound and Music Computing Conference (SMC), 2010*, 2010.

[27] Gino Angelo Velasco, Nicki Holighaus, Monika Dörfler, and Thomas Grill, "Constructing an invertible constant-Q transform with non-stationary Gabor frames," *Proceedings of DAFX11*, 2011.

[28] Nicki Holighaus, Monika Dörfler, Gino Angelo Velasco, and Thomas Grill, "A framework for invertible, real-time constant-Q transforms," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 4, pp. 775 –785, 2013.

[29] Nicki Holighaus, Zdenek Prusa, and Christoph Wiesmeyr, "Designing tight filter bank frames for nonlinear frequency scales," *Sampling Theory and Applications 2015*, 2015.

[30] Christian Schörkhuber, Anssi Klapuri, Nicki Holighaus, and Monika Dörfler, "A Matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution," in *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. AES, January 2014.

[31] Jules Storer, "JUCE: Jules' utility class extensions (version 4.1)," 2014, Available at: `http://www.juce.com`.

[32] Zdenek Prusa, Peter L. Søndergaard, Nicki Holighaus, Christoph Wiesmeyr, and Peter Balazs, "The Large Time-Frequency Analysis Toolbox 2.0," in *Sound, Music, and Motion*, Mitsuko Aramaki, Olivier Derrien, Richard Kronland-Martinet, and Sølvi Ystad, Eds., Lecture Notes in Computer Science, pp. 419–442. Springer International Publishing, 2014.

[33] Matteo Frigo and Steven G. Johnson, "The design and implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005, Special issue on "Program Generation, Optimization, and Platform Adaptation".