UNIVERSITÄT AUGSBURG

Fast Gabor Transformation for processing high quality audio

C. G. v. d. Boogaart, R. Lienhart

Report 2005-21

Oktober 2005
revised Mai 2006

INSTITUT FÜR INFORMATIK
D-86135 AUGSBURG

FAST GABOR TRANSFORMATION FOR PROCESSING HIGH QUALITY AUDIO

C. G. v. d. Boogaart, R. Lienhart

Multimedia Computing Lab
University of Augsburg, 86159 Augsburg, Germany
{boogaart,lienhart}@informatik.uni-augsburg.de

ABSTRACT

The Gabor transformation with a Gaussian window has several advantages over classical short time transformations such as the windowed FFT and Wavelets. It allows for perfect localization in time and frequency according to the absolute bound expressed by the Heisenberg uncertainty principle. Furthermore the time-frequency resolution can be chosen as desired. This is bought dearly by the necessity of oversampling and very large windows resulting in high computational and storage costs. To overcome this disadvantages the FFT can be used as an underlying technique to speed up the computation for some rare dedicated time-frequency resolutions. In this paper the use of the FFT is extended to allow choosing the time-frequency resolution arbitrarily, by introducing only a small computational overhead. With the same approach we show, how to use the FFT to compute the Gabor transformation on non-separable lattices. The speed-up factors over an optimized DFT approach range from 2.5 to 100.

1. INTRODUCTION

Time-frequency transformations reveal local properties of a signal. The kind of properties, however, which are revealed depend strongly on the window and the window length. Multiwindow techniques are therefore used, to find the best matching window length for a given task (see e.g. [1]). In comparison the human ear has a time frequency resolution, which closely reaches the physical limit expressed in the Heisenberg principle. It is also capable of adapting its current time-frequency resolution to the current content of the signal in accordance to the Heisenberg principle (see [2]). With our approach we lower the computational cost of adapting the window to the most suitable time-frequency resolution.

1.1. Related Work

The Gabor transformation ([3]) is related to the MLT (Modulated lapped transform) [4], which has many applications in audio processing. The MLT is also restricted to very dedicated window lengths and does not use a Gaussian window.

The Gabor transformation has like the DFT a complexity in the order of $O(N^2)$ (see e.g. [5]). There exist two main approaches, to reduce the complexity: One approach is based on recursive filters (see e.g. [6]), which lead to a complexity in the order of $O(N)$. Its main disadvantage is, that a non-causal recursive filter has to be used. This is possible in image processing, where the whole image is available, but not for real-time audio processing. The other approach is based on the FFT (see e.g. [7]), with a complexity in the order of $O(NlogN)$. Its disadvantage is, that the possible window lengths are restricted, as will be discussed later in this paper. Our approach uses the FFT, but extends its usage to arbitrary window lengths.

1.2. Outline

We start with some fundamentals about the Gabor transformation in section 2 and discuss how much oversampling is necessary and how large the window has to be for perfect reconstruction. In section 3 we extend the technique of computing the Gabor transformation by the use of the FFT (Fast Fourier Transform) for some rare fixed window lengths to arbitrary window lengths. Section 4 presents results of performance measurements between a DFT and a FFT implementation for various window lengths and summarizes the main results.

2. GABOR TRANSFORMATION

2.1. Fundamentals of the Gabor transformation

The Gabor transformation splits up a time function $x(t)$ in its time-frequency representation $X(t, f)$. In the case of a separable lattice, the Gabor transformation is defined as follows: From a single prototype or windowing function $g(t)$, which is localized in time and frequency, a Gabor system $g_{na,mb}(t)$ is derived by time shift a and frequency shift b (see [8]):

$$g_{na,mb}(t) = e^{2\pi jmbt} g(t - na),\ n, m \in \mathbb{Z},\ a, b \in \mathbb{R}, \quad (1)$$

a and b are called the lattice constants. The Gabor system covers the whole time-frequency plane. The Gabor transformation is than expressed as follows:

$$c_{nm} = X(na, mb) = \int_{-\infty}^{+\infty} x(t) g^*_{na,mb}(t)\, dt. \quad (2)$$

c_{nm} are called the Gabor coefficients of $x(t)$. The inverse transformation or Gabor expansion is calculated with the window function $\gamma(t)$, which is called dual window of $g(t)$. With the Gabor system $\gamma_{na,mb}(t)$ defined analogously to equation (1) the Gabor expansion is defined as follows (see [9]):

$$x(t) = \frac{1}{L} \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} c_{nm} \gamma_{na,mb}(t) \tag{3}$$

with $L = \sum_{k=-\infty}^{\infty} |\gamma(ka)|^2$. The theory of the Gabor transformation leads to the result, that the window functions $g(t)$ and $\gamma(t)$ and the lattice constants a and b must fulfill certain requirements in order to assure the invertibility. The appropriate choices are discussed in the following two sections. For use in digital signal processing formulas (2) and (3) have to be discretized, using sums instead of integrals and sums of finite length, which is discussed in the last section of this chapter.

2.2. The Gaussian window

The Gaussian window function is given as:

$$g(t) = \frac{1}{\sqrt{2\pi\sigma_t^2}} e^{-\frac{1}{2}\frac{t^2}{\sigma_t^2}}. \tag{4}$$

and has the following advantageous properties[1] (see e.g. [6]):

- Minimal extent in the time frequency plane according to the Heisenberg uncertainty principle.
- Localized shape, i.e. only one local and global maximum with strict decay in time and frequency direction.

The uncertainty principle of Heisenberg says that the product of temporal and frequency extent of a window function has a total lower limit. If the extents are defined in terms of standard deviations of the window function and of its Fourier transformation respectively, it can be expressed with the following inequality (see [10]):

$$\sigma_t \sigma_f \geq \frac{1}{4\pi}. \tag{5}$$

The "$=$" is only reached for the Gaussian window function (see e.g. [11]), which therefore has the minimal possible extent in the time-frequency plane. Its Fourier transformation has the same Gaussian shape as the time function itself:

$$G(f) = \frac{1}{\sqrt{2\pi\sigma_f^2}} e^{-\frac{1}{2}\frac{f^2}{\sigma_f^2}}, \text{with } \sigma_f = \frac{1}{4\pi\sigma_t}. \tag{6}$$

The dual window $\gamma(t)$ should also be localized in time and frequency to preserve the local influence of the Gabor coefficients on the result of the inverse transformation. This is best achieved by the Gaussian as dual window-function again. To ensure perfect reconstruction some restrictions are imposed on the choice of lattice constants a and b as discussed in the following section.

[1] These properties imply, that the Gaussian window extends to infinity and is never truncated. See section 2.4 for a discussion on how to truncate, in order to preserve these properties.

2.3. Choice of lattice constants and oversampling factor

In his original paper ([3]) Dennis Gabor suggested to choose $ab = 1$ which is called critical sampling. This choice has implicit influence on the shape of the dual window. In fact with the Balian-Low theorem it can be shown, that in this case, the dual window extends to infinity and is not localized at all (see [12]). The solution is to choose $ab < 1$, which is referred to as the oversampled case. It leads to better localized dual windows and numerically stable analysis and synthesis.

We therefore have to determine an appropriate oversampling factor for $ab < 1$. In the literature normally the cases of rational oversampling ($ab = \frac{p}{q}$, $p, q \in \mathbb{N}$ and $p < q$) and integer oversampling ($ab = \frac{1}{q}$, $q \in \mathbb{N}$) are discussed. Bastiaans [9] proposes to take $ab = \frac{1}{3}$ for which the ideal dual window of the Gaussian is getting very close to a Gaussian window. He mentions that for increasing values of q the resemblance of the Gaussian window and its dual window further increases. In simple empirical hearing tests we found that an oversampling factor starting at $ab = \frac{1}{5}$ avoids hearable differences between an original and a reconstructed sound in case of using the same Gaussian window for analysis and synthesis[2]. This holds for speech and for full bandwidth music. An oversampling factor of $q = 5$ is therefore the necessary and sufficient accuracy for high quality audio processing.

In the following, if necessary we name a, b for the critical sampled case a_{crit}, b_{crit} and for the oversampled case a_{over}, b_{over}. The resulting lattice and how it covers the time-frequency plain is illustrated in figure 1. The gray shaded circles indicate the extent of a single Gaussian window in the time-frequency plain expressed in its standard deviations σ_t and σ_f. To ensure the same overlapping of the Gaussians in time and frequency direction, we therefore have to set:

$$\frac{\sigma_t}{\sigma_f} = \frac{a_{crit}}{b_{crit}} = \frac{a_{over}}{b_{over}}. \tag{7}$$

With $a_{over} b_{over} = \frac{1}{q}$ this holds for:

$$a_{over} = \frac{a_{crit}}{\sqrt{q}}, b_{over} = \frac{b_{crit}}{\sqrt{q}}. \tag{8}$$

With formula (6) and formula (7) we can solve:

$$\sigma_t = \frac{1}{\sqrt{4\pi}}\sqrt{\frac{a}{b}}, \sigma_f = \frac{1}{\sqrt{4\pi}}\sqrt{\frac{b}{a}}. \tag{9}$$

[2] Sound examples can be found at: http://www.informatik.uni-augsburg.de/~boogaart/.

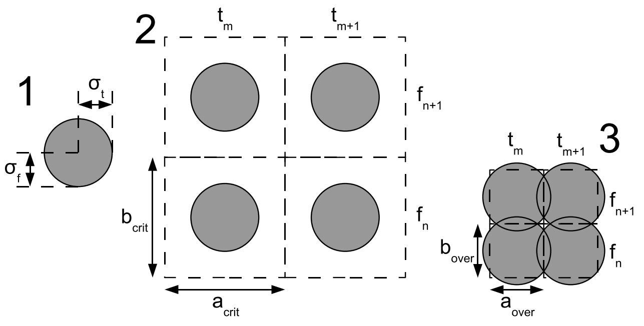

Fig. 1. Coverage of the time-frequency plain for the separable case: The gray shaded circles indicate the extent of a (1) single Gaussian window expressed in its standard deviations σ_t and σ_f. (2) the critical sampled case and (3) the oversampled case.

As we will in the following always choose b and determine the other values, we write further with $ab = \frac{1}{q}$:

$$\sigma_t = \frac{1}{\sqrt{4\pi q}}\frac{1}{b}, \sigma_f = \sqrt{\frac{q}{4\pi}}b. \qquad (10)$$

With an oversampling factor of $q = 5$ and these formulas, it is feasible to take the Gaussian window $g(t)$ as its own dual window $\gamma(t) = g(t)$. One still has the freedom to choose either a_{crit} or b_{crit}, i.e. to choose an appropriate window length for the current task.

2.4. Discretization and Truncation

The formulas (2) and (3) are for a continuous representation of $x(t)$ and calculate sums and integrals over infinity. Therefore they have to be discretized and the sums and integrals have to be truncated in order to be implemented. This corresponds to bandlimiting and sampling $x(t)$ and to truncating $g(t)$. $x(t)$ is sampled with the sampling frequency f_s. With $T = 1/f_s$ and $k \in \mathbb{Z}$ $x(t)$ becomes $x(kT)$. To fulfill the sampling theorem, the bandwidth f_B of $x(kT)$ has to fulfill $f_B \leq f_s/2$ (see e.g. [5])[3]. We define $M \in \mathbb{N}$ as the number of frequency bands with $M = \lceil \frac{1}{2}\frac{f_s}{b_{over}} \rceil$. With t_{cut} half the window length, we define $N \in \mathbb{N}$ with $N = t_{cut}f_s = \frac{t_{cut}}{T}$ as half the window length in samples. The Gabor transformation can then be implemented as:

$$c_{nm} = X(na, mb) = \frac{1}{L}\sum_{k=-N}^{N-1} x(kT)g^*_{na,mb}(kT) \qquad (11)$$

with its inverse:

$$x(kT) = \frac{1}{L}\sum_{n=\lfloor \frac{kT-t_{cut}}{a} \rfloor}^{\lceil \frac{kT+t_{cut}}{a} \rceil}\sum_{m=0}^{M-1} c_{nm}g_{na,mb}(kT) \qquad (12)$$

[3]For HiFi audio typical values are $f_s = 44.1kHz$ (CD) or higher and $f_B = 20kHz$.

where $L = \sqrt{q\sum_{k=-N}^{N-1}|g(ka)|^2}$ and $m \in [0, M-1]$. We still have to determine t_{cut} and N respectively, which correspond as mentioned to the truncation of the Gaussian window. We have chosen empirical listening tests to find an appropriate truncation. The goal was to get a reproduced sound with no hearable difference from the original sound. We define the decline D of the Gaussian window from the maximum to the cut expressed in dB as:

$$D = 20log\frac{g(0)}{g(t_{cut})}dB. \qquad (13)$$

For a given D we get with (4):

$$t_{cut} = \sqrt{2ln\left(10^{\frac{D}{20}}\right)}\sigma_t. \qquad (14)$$

In our implementation high values for D (up to $200dB$) have been tested in $10dB$ increments, but values of $D \geq 30dB$ have shown to be completely sufficient for high quality audio[4].

Remark: Storing the Gabor coefficients is very efficient independent of the time-frequency resolution. A discretized time signal of duration t_{dur} needs $N^{\mathbb{R}} = t_{dur}f_s$ real sample values. The Gabor coefficients need

$$N^{\mathbb{C}}_{Gabor} = \frac{t_{dur}}{a}\frac{f_B}{b} = t_{dur}f_B q \qquad (15)$$

complex or

$$N^{\mathbb{R}}_{Gabor} = 2 \cdot N^{\mathbb{C}}_{Gabor} = 2t_{dur}f_B q \qquad (16)$$

real and imaginary values combined for storage. That is for storage of the Gabor transformation $N^{\mathbb{R}}_{Gabor} = q \cdot N^{\mathbb{R}}$ values are needed, independent of the current time frequency resolution. This can be further enhanced by setting $f_B = 20kHz$ with $f_B < f_s/2$.

3. COMPUTING THE GABOR TRANSFORMATION BY MEANS OF FFT

3.1. Fundamentals of the FFT

The FFT allows fast implementations of the DFT (Discretized Fourier Transform) for all cases, where the DFT length is a power of 2. We use similar naming conventions as for the Gabor transformation before, in order to stress the similarities. We therefore use $2M$ as the DFT length. The DFT length is the number of input samples and output frequency values of the DFT. The DFT is than defined as (see [5]):

$$X(mb) = \sum_{k=-M}^{M-1} x(kT)e^{-2\pi jmk\frac{1}{2M}} \qquad (17)$$

[4]Sound examples can be found at: http://www.informatik.uni-augsburg.de/~boogaart/.

and the IDFT (inverse DFT) is defined as:

$$x(kT) = \sum_{m=0}^{2M-1} X(mb)e^{2\pi jmk\frac{1}{2M}}. \qquad (18)$$

with $m \in [0, 2M-1]$. If $x(kT)$ is real, which of course is always the case for audio signals, the upper half of $X(mb)$ is the flipped, conjugate complex of the lower half of $X(mb)$ (see e.g [5]). It is therefore sufficient, to store only the values $X(mb)$ for $m \in [0, M-1]$ and reconstruct the other values for inverse transformation. In case of the FFT and the IFFT, the formulas stay the same with the added constraint, that the DFT/FFT length is a power of 2.

3.2. Fast Gabor transformation for selected window lengths

We now want to show, how the FFT can be used to compute the Gabor transformation. We therefore have to consider the window function and have to adapt the output of the FFT, so that it matches the lattice of the Gabor transformation.

For a fixed time, e.g. $n = 0$ the Gabor transformation (11) can be written as:

$$c_{0m} = \sum_{k=-N}^{N-1} x(kT)\frac{1}{L}g(kT)e^{-2\pi jmb_{over}kT}. \qquad (19)$$

The window can be understood as part of the signal. We therefore define a new signal $\hat{x}(kT) = x(kT)\frac{1}{L}g(kT)$ and get with $b_{over} = \frac{f_s}{2M}$ and $T = \frac{1}{f_s}$:

$$c_{0m} = \sum_{k=-N}^{N-1} \hat{x}(kT)e^{-2\pi jmk\frac{1}{2M}}. \qquad (20)$$

Now we have to have a look on the window length and the number of frequency bands which have to be identically for the FFT. So either M or N have to be changed. We therefore determine $N = t_{cut}f_s$:

$$N = \sqrt{2ln\left(10^{\frac{D}{20}}\right)}\,\sigma_t f_s = \sqrt{\frac{2ln\left(10^{\frac{D}{20}}\right)}{4\pi q}}\,\frac{f_s}{b_{over}}. \qquad (21)$$

With $D = 30dB$ and $q = 5$ we get $N = 0.46891\frac{f_s}{b_{over}}$. In case of the FFT M is no more rounded and becomes exactly $M = \frac{1}{2}\frac{f_s}{b_{over}} = 0.5\frac{f_s}{b_{over}}$. That is $N < M$. It is therefore possible, to extend the window length to $2M > 2N$ and to compute the Gabor transformation as follows:

$$c_{0m} = \sum_{k=-M}^{M-1} \hat{x}(kT)e^{-2\pi jmk\frac{1}{2M}}. \qquad (22)$$

This formula can be realized through a windowed FFT of $x(kT)$ every a_{over}. The inverse Gabor transformation can be realized by overlapped adding of subsequent IFFT's. The upper half of $X(mb)$ is reconstructed by a flipped, conjugate complex of c_{nm}.

Calculating the Gabor transformation by means of the FFT reduces the computational load greatly. As the FFT only exists for dedicated lengths, these forces the use of only some $b_{over} = \frac{f_s}{2M}$. They are moreover dependent on the sampling frequency f_s of the original signal. If for some reason a different b_{over} is needed the only possible solution up to now was to calculate a much more time consuming DFT. Moreover this only works for the Gabor transformation on a separable lattice, but not on a non-separable lattice as illustrated in figure 2. The non-separable case is sometimes preferred, because it is more effective in covering the time-frequency plane. Every second vector of frequency values is shifted by $\frac{b_{over}}{2}$ and the lattice gets a hexagonal structure. The shifted frequency vectors can not be calculated with the FFT in the classical way.

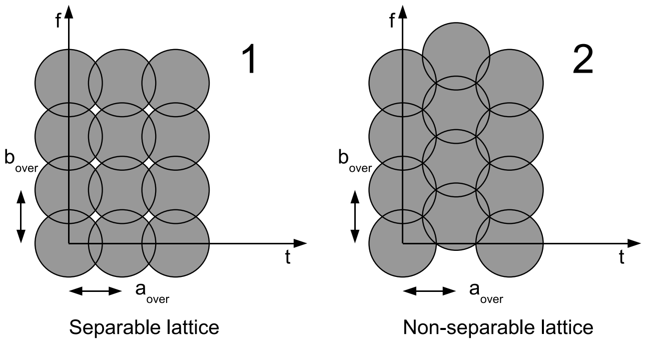

Fig. 2. Coverage of the time-frequency plane for (1) the separable lattice and (2) the non-separable lattice. In (2) the shifted frequency vectors can not be calculated with the FFT.

3.3. Fast Gabor transformation for arbitrary window lengths

These hard restrictions can be soften by allowing some computational overhead. An overhead is generally acceptable if the overall execution duration is still lower than that of a DFT.

It is common practice to zero pad a given signal to a power of two boundary, if a DFT shall be calculated fast. By this action, the frequency spacing of the resulting values is also modified. We follow a very similar idea. By extending the window from a FFT boundary at $2M$ to a broader window at the FFT boundary $2\dot{M} = 2M \cdot 2^\eta$ ($\eta \in \mathbb{N}$), the density of frequency values is increased. Instead of $b_{over} = \frac{f_s}{2M}$ we get $\dot{b}_{over} = \frac{f_s}{2\dot{M}} = \frac{b_{over}}{2^\eta}$. Because of the frequency-shape of the window, which is not altered by altering the FFT length, the frequency values can now be downsampled by a factor $\kappa \in \mathbb{N}$, $\kappa < 2^\eta$, to get a $\ddot{b}_{over} = \kappa\dot{b}_{0ver}$. Figure 3 explains the scenario.

For inverse transformation, the skipped values have to be reconstructed. Again the upper half of $X(mb)$ is the flipped,

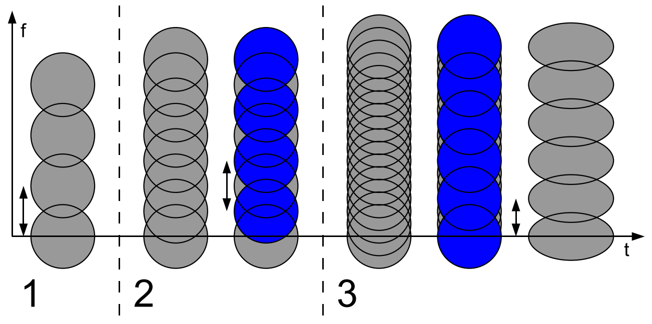

Fig. 3. This illustrates the downsampling cases. The vertical arrows mark in each case b_{over}. (1) shows a FFT with length $2M$ and the corresponding b_{over}. (2) shows a FFT with length $2\dot{M} = 2M \cdot 2$. The blue circles mark a possible frequency vector for the non-separable lattice with the same lattice constants. (3) shows a FFT with length $2\dot{M} = 2M \cdot 2^2$. The blue circles mark a frequency vector generated by downsampling with factor 3. This alters the lattice constants. The resulting frequency vector with adapted σ_t and σ_f is shown.

conjugate complex of c_{nm}. The complex frequency signal has to be sampled up by the factor κ and convoluted with the reconstruction filter, which is the frequency representation of the Gaussian window. A convolution in the frequency domain corresponds to a (much more efficient) multiplication in the time domain. The convolution is therefore replaced by a multiplication with the time representation of the Gaussian window in the time domain.

4. RESULTS

Figure 4 shows the data of the performance measurements.

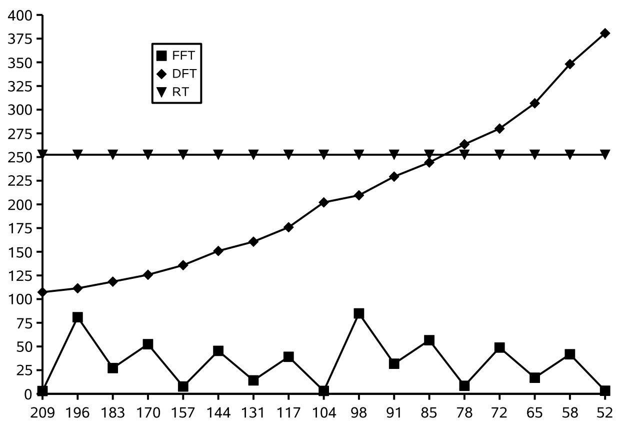

Fig. 4. Comparison of computation times in seconds on the y-axis over b_{crit} on the x-axis. RT denotes the length of the file in seconds (realtime-border). As can be seen the FFT approach outperforms the DFT approach.

Two implementations of the Gabor transformation are compared, one based on the DFT and the other based on the FFT. Both implementations are highly optimized by making use of the SIMD instructions of current CPUs. The test signal was a mono audio file of 252.4 sec at 48 kHz with 32 bit floating point precision. The compuations where performed on a AMD Opteron processor with 2.0 GHz. The computation times in seconds on the y-axis are plotted against b_{crit} on the x-axis. The horizontal line denotes the length of the file in seconds and therefore represents the realtime border. As can be seen in figure 4 the FFT approach outperforms the DFT approach. Table 1 shows how b_{crit} has to be translated into the relevant parameters of the transformations.

	FFT			DFT		
b_{crit}	n	down	sec	wlen	sec	speed-up
209,6	9	1	2,99	339	107,36	35,91
196,5	13	15	80,88	363	111,42	1,38
183,4	12	7	27,2	389	118,42	4,35
170,3	13	13	52,4	417	125,73	2,4
157,2	11	3	7,75	453	135,84	17,53
144,1	13	11	45,37	493	150,84	3,32
131,0	12	5	14,2	543	160,63	11,31
117,9	13	9	39,12	603	175,87	4,5
104,8	10	1	3,11	679	202,12	64,99
98,2	14	15	84,9	725	209,62	2,47
91,7	13	7	31,8	777	229,33	7,21
85,1	14	13	56,64	835	244,23	4,31
78,6	12	3	8,5	905	263,43	30,99
72,0	14	11	48,88	987	280,03	5,73
65,5	13	5	17,02	1087	306,75	18,02
58,9	14	9	41,77	1207	348,11	8,33
52,4	11	1	3,27	1359	380,7	116,42

Table 1.

Discussion: For the FFT the cases $n = 9$, $n = 10$ and $n = 11$ are optimal, because of minimal length and no need of downsampling. Speed-up factors around 100 have been measured. They correspond to the standard Gabor transformation with FFT implementation of the typical FFT lengths of 512, 1024 and 2048 samples and the corresponding frequency spacing. The in-between cases are successively filling the intervals in between using our exceeded length and downsampling approach. They have smaller speed-up values compared to the DFT approach of down to 2.5. See table 1 for the exact values. The DFT times are strictly monotonic increasing with the window length. The monotony is preserved for all interim values (see table 1).

The FFT approach outperforms the DFT approach for many window lengths. The finer the window length is choosen, the smaller are the the speed-up factors. Also the longer the window becomes, the better the FFT performs even on very fine window length rasters.

5. REFERENCES

[1] Patrick J. Wolfe, Simon J. Godsill, and Monika Dörfler, "Multi-gabor dictionaries for audio time-frequency analysis," *Proceedings of WASPAA 2001*, 2001.

[2] C. G. v. d. Boogaart, *Master thesis: Eine signal-adaptive Spektral-Transformation für die Zeit-Frequenz-Analyse von Audiosignalen*, Technische Universität München, 2003.

[3] Dennis Gabor, "Theory of communication," *Journal of the Institution of Electrical Engineers*, pp. 429–457, November 1946.

[4] Henrique S. Malvar, *Signal Processing with Lapped Transforms*, Artech House, Inc., Norwood, MA, USA, 1992.

[5] Alan V. Oppenheim and Ronald W. Schafer, *Discrete-time signal processing*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[6] Ian T. Young, Lucas J. van Vliet, and Michael van Ginkel, "Recursive gabor filtering," *IEEE Transactions on Signal Processing*, vol. 50, pp. 2798–2805, 2002.

[7] Sigang Qiu, Feng Zhou, and Phyllis E. Crandall, "Discrete gabor transforms with complexity $o(nlogn)$," *Signal Process.*, vol. 77, no. 2, pp. 159–170, 1999.

[8] Thomas Strohmer, "Approximation of dual gabor frames, window decay, and wireless communications," *Appl.Comp.Harm.Anal.*, vol. 11, no. 2, pp. 243–262, 2001.

[9] Martin J. Bastiaans, "Optimum sampling distances in the gabor scheme," *Proc. CSSP-97, Mierlo, Netherlands*, pp. 35–42, 1997.

[10] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine*, vol. 8, no. 4, pp. 14–38, 1991.

[11] P. A. Tipler, *Physik*, Spektrum Akademischer Verlag, Heiderlberg, Berlin, 1994.

[12] Hans G. Feichtinger and Thomas Strohmer, *Advances in Gabor Analysis*, Birkhäuser, Boston, 2003.