Using Nonequispaced Fast Fourier Transformation to Process Optical Coherence Tomography Signals

Dierck Hillmann
Thorlabs HL AG

Gereon Hüttmann
Institut für
Biomedizinische Optik

Peter Koch
Thorlabs HL AG

ABSTRACT

In OCT imaging the spectra that are used for Fourier transformation are in general not acquired linearly in k-space. Therefore one needs to apply an algorithm to re-sample the data and finally do the Fourier Transformation to gain depth information. We compare three algorithms (Non-Equispaced DFT, interpolated FFT and Non-Equispaced FFT) for this purpose in terms of speed and accuracy. The optimal algorithm depends on the OCT device (speed, SNR) and the object.

1. INTRODUCTION

In Fourier Domain OCT imaging one problem is that the spectral data are usually not acquired linearly in the wavenumber k but instead in some other variable λ. Therefore applying a simple Fast Fourier Transformation (FFT) to the data would yield broadened signals and the resulting images will appear blurred. The standard solution to this problem is to linearly interpolate the acquired data with the help of the function $\lambda(k)$ in order to make them linear in the wavenumber k. $\lambda(k)$ can in general be determined using functional optimization or piecewise Fourier transformation (STFT). However, in most cases a simple resampling is not sufficient as this in general produces major side lobes for strong signals especially at high imaging depth. Therefore additional over-sampling is used which reduces this problem and increases image quality significantly, however this oversampling adds computation time as the length of the FFT increases as well.

There are alternative algorithms for performing FFTs on non equispaced nodes.[1] One of the best algorithms for this case is the NFFT, which – depending its settings – often shows better performance and less artifacts than the interpolated FFT (iFFT).

Nowadays high-speed OCT devices have reached acquisition speeds of spectra with more than 100 kHz at 2048 pixels per spectrum. To process this amount of data online on a standard PC the FFT with oversampling and linear interpolation is too slow. However, the processing results do not need to be as precise as for the slower systems as in general the Signal-to-Noise ratio is decreased as well and therefore side-effects of inferior algorithms cannot be seen. We therefore compare various algorithms and approaches to the problem in terms of image quality and speed.

2. ALGORITHMS

2.1 Non-Equispaced Discrete Fourier Transformation

We first have a look at the mathematical precise result, which can be used as a reference for other processing techniques. The Discrete Fourier Transformation $\tilde{f}_x$ of a signal with N data points, acquired linearly in k namely f_k is given by

$$\tilde{f}_x = \sum_{k=0}^{N-1} f_k \exp\left(-\mathrm{i}\frac{2\pi}{N} k \cdot x\right). \tag{1}$$

Optical Coherence Tomography and Coherence Techniques IV, edited by Peter E. Andersen,
Brett E. Bouma, Proc. of SPIE-OSA Biomedical Optics, SPIE Vol. 7372, 73720R
 · doi: 10.1117/12.831789

However, in our case the signal f_k is not given and instead we have a signal f_λ and a function $k(\lambda)$ describing the relation between λ and k. To get $\tilde{f}_x$ we therefore need to compute

$$\tilde{f}_x = \sum_{\lambda=0}^{N-1} f_\lambda \exp\left(-\mathrm{i}\frac{2\pi}{N}k(\lambda)\cdot x\right). \tag{2}$$

The result will be referred to as Non-Equispaced Discrete Fourier Transformation (NDFT). As the creation of an FFT algorithm of a standard DFT (1) uses the equispacing of the input data in k-space we cannot simply create a fast version of (2).

The NDFT is in principle a matrix multiplication and one can precompute the matrix elements to get optimal performance. But due to the high number of multiplications and additions needed the NDFT is still slow compared to the alternative algorithms. The computational complexity of this appraoch is approximately $\sim \mathcal{O}(N^2)$. Similar to standard DFT being a development into orthogonal functions, namely unchirped sine and cosine functions, this approach can be seen as the development into orthogonal chirped sine and cosine fucntions. A similar approach was already suggested by.[2]

2.2 Interpolated FFT

The interpolated FFT (iFFT) starts by resampling the data given linearly in λ to be linearly in k. The interpolation step is done by means of linear interpolation. Assuming that λdescribes the index of the data points that have been acquired and thus running from 0 to $N-1$, the values f_k are approximated by

$$\begin{aligned} f_{\alpha k} &\approx \left(f_{\lceil\lambda(k)\rceil} - f_{\lfloor\lambda(k)\rfloor}\right)\left(\lambda(k) - \lfloor\lambda(k)\rfloor\right) + f_{\lfloor\lambda(k)\rfloor} \\ &= f_{\lceil\lambda(k)\rceil}\underbrace{\left(\lambda(k) - \lfloor\lambda(k)\rfloor\right)}_{\text{weight right point}} + f_{\lfloor\lambda(k)\rfloor}\underbrace{\left(1 - \left(\lambda(k) - \lfloor\lambda(k)\rfloor\right)\right)}_{\text{weight left point}}, \text{ for all } \alpha k \in \mathbb{N}, \end{aligned}$$

where $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ donate the rouding up and rounding off operations, repcetively. For this interpolation an oversampling factor α is used: The N input values f_λ are hence used to create αN values, the length of the subsequent FFT is therefore increased by a factor α, as is the computation time. Fig. 1a shows that the linear interpolation can be seen as a weighted average of the neighboring points of the interpolated values. The weights for the two points are given by a triangle function given by

$$\triangle(k) = \begin{cases} 1 - |k| & \text{for } k \in [-1;1] \\ 0 & \text{otherwise} \end{cases}$$

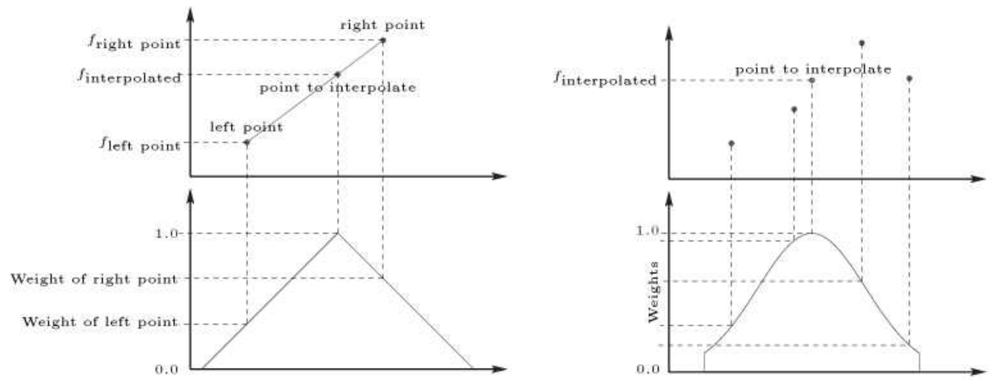
The linear interpolation is therefore also a convolution with $\triangle(k)$. After the FFT one therefore needs to deconvolve the function by dividing the results with the Fourier transform of this triangle function. Otherwise one will get results which appear to have an increased roll-off in signal intensity, although the SNR will remain unchanged. The Fourier Transformation of $\triangle(k)$ is given by

$$\tilde{\triangle}(x) = \int_{-\infty}^{+\infty} \mathrm{d}k\,\triangle(k)\cdot \mathrm{e}^{-\mathrm{i}kx} = \frac{\sin^2\left(\frac{x}{2}\right)}{\left(\frac{x}{2}\right)^2} = \operatorname{sinc}^2\left(\frac{x}{2\pi}\right).$$

To the result $\tilde{f}_x$ of the FFT one therefore needs to multiply a correction c_x which is given by

$$c_x^{-1} = \frac{\sin^2\frac{\pi x}{N}}{\left(\frac{\pi x}{N}\right)^2} = \operatorname{sinc}^2\left(\frac{x}{N}\right)$$

The linear interpolation and the following FFT have approximately a complexity of $\sim \mathcal{O}(\alpha N)$ and $\sim \mathcal{O}(\alpha N \cdot \log \alpha N)$, respectively. For optimal performance one can precompute the factors used for the linear interpolation and the values used for the deconvolution after the FFT. In the following we will refer to the this algorithm as iFFTα with α replaced by a number reflecting the oversampling.



(a) Weights for the data points $f_{\lfloor \lambda(k) \rfloor}$ (left point) and $f_{\lceil \lambda(k) \rceil}$ (right point) are representing a convolution with a triangle function for the linear interpolation.

(b) For NFFT the triangle function is replaced by a Kaiser-Bessel window.

Figure 1: Interpolated FFT and NFFT in comparison.

2.3 Non-Equispaced Fast Fourier Transformation

The Non-Equispaced Fast Fourier Transformation (NFFT) is similar to the iFFT, but instead of convolving with a triangle function another window function is used which is cut off sufficiently far from the data point that needs to be approximated (see fig. 1b).[1] The cut off is determined by a parameter m and therefore the algorithm has in addition to α another parameter that influences accuracy and speed of the algorithm. In general the algorithm will be faster but increased artifacts are observed when α and m are chosen smaller. The convolution has a complexity of about $\sim \mathcal{O}(m\alpha N)$ and therefore the complete complexity is approximately $\sim \mathcal{O}(m\alpha N + \alpha N \cdot \log \alpha N)$.

In addition, it has been shown that in most cases the Kaiser-Bessel window yields best results. As for OCT images the window function values for the data points can be precomputed completely this window function is the best choice for our purpose. We will refer to this algorithm as NFFT$\,\alpha m$.

3. RESULTS

3.1 Benchmarks

To compare the speed of the different algorithms we also take additional processing steps into account to get a real measure on how many A-scans per second can be performed. We measure the full processing from integer raw data as acquired by the camera/analog-digital converter to a final OCT image as floating point values. The steps included are:

1. casting the camera output (8-bit unsigned integer data) to single precision (32-bit) floating point numbers
2. removing offset errors (dark signals of the camera)
3. apodization (deconvolution) of the spectrum by dividing by a signal-less reference spectrum
4. Fourier Transformation (including the λ to k conversion)
5. building the absolute values of the complex output
6. taking the logarithms
7. applying a high pass filter to remove the Fourier transform of the spectrum itself, which is just a constant addition to each A-scan pixel if performed at the end

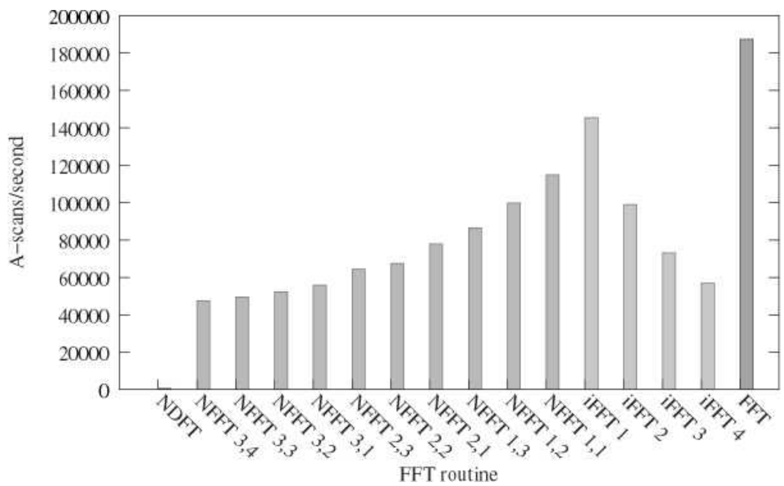


Figure 2: Benchmark results of various processing routines.

The deconvolution for iFFT and NFFT can be included in the final high pass filter step. All algorithms were performed on a Core2 Quad with 2.6 GHz where multi-threading and vectorization were used to maximize speed.

Benchmarks were performed for data acquired by a high speed OCT device with linear-k spectrometer as this is the case where the highest A-scan rates are expected and needed. Anyway, the performance of the NFFT algorithms depends only slightly on the function $k(\lambda)$, namely this function determines how many data points need to be evaluated for the pre-FFT convolution. All other algorithms do not depend on $k(\lambda)$ at all. The benchmark results can be found in fig. 2. For the iFFT the number of original data points that are used for computing an interpolated value is always two and thus fixed, whereas for the NFFT this number might vary slightly. For this reason the NFFT11 shows slightly worse performance than the iFFT1. The other results are as expected, the NFFT performance drops with increased α or m and the iFFT performance drops for increased α. The standard FFT without interpolation is the fastest and the NDFT (a full matrix multiplication) is clearly the slowest.

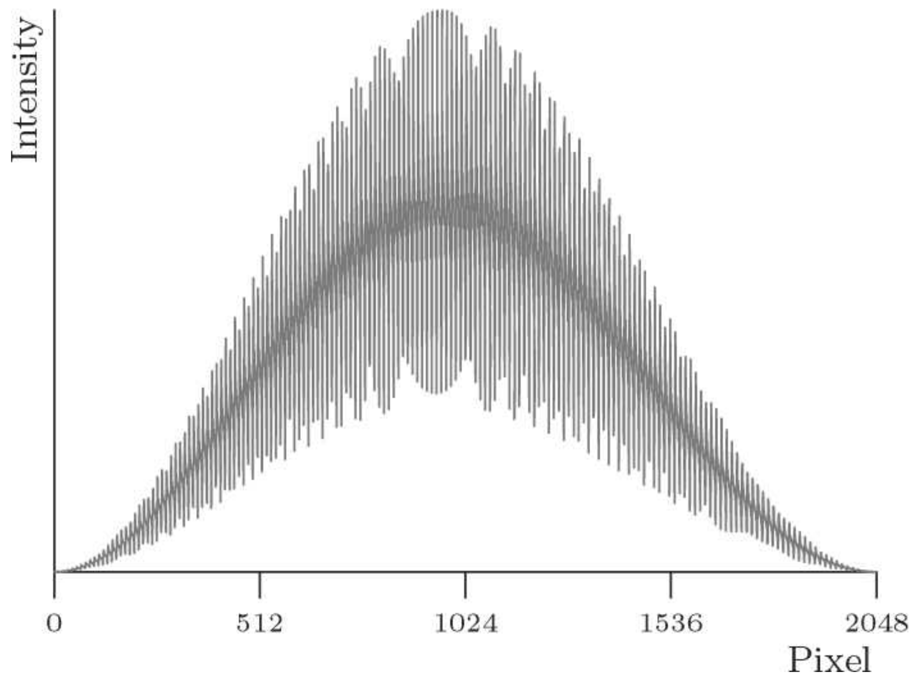
3.2 Simulation

Degradation of the A-scans due to the different algorithms are best seen for artificial signals with no noise. Fot these, the precision of the results are limited by numerical accuracy only. The simulated signals were created with an Hann-windowed spectral shape and five modulation frequencies and an artificial function as chirp which was given by

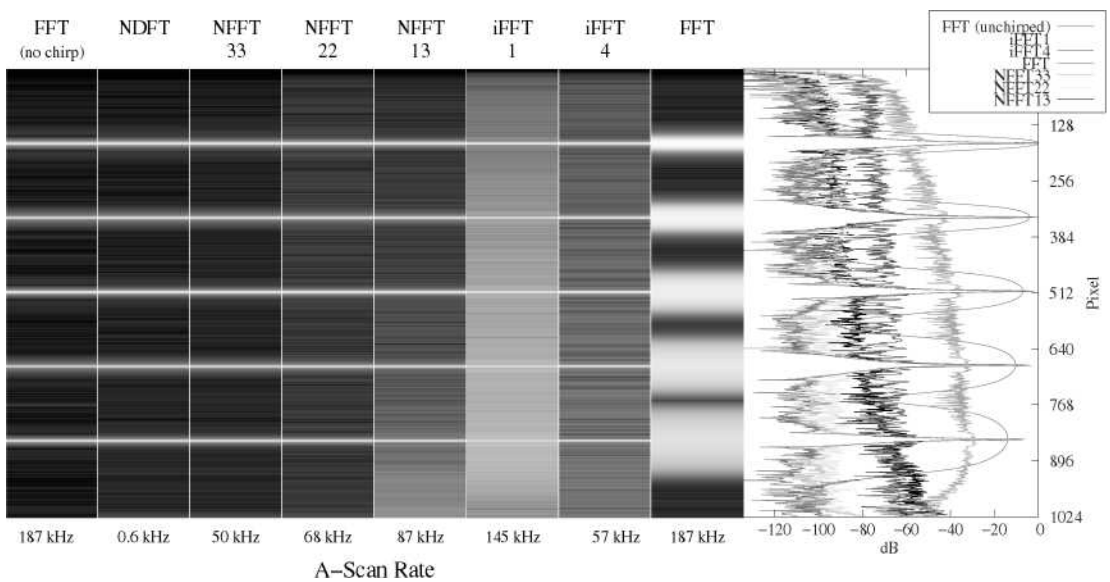
$$k(\lambda) = \lambda + \left[1 - \left(\frac{\lambda - N/2}{N/2}\right)^2\right] \cdot h,$$

where $h = 50$. Additionally, the spectra were convoluted with a Gaussian point-spread function and modulation transfer function with $\sigma = 0.5$ pixels to artificially create a roll-off in signal intensity.

Furthermore we also created a spectrum with no chirp. This is the ideal case where no interpolation and oversampling is needed at all. It can be used to compare the output of the algorithms. The results can be found in Fig. 3.



(a) Simulated spectrum used for evaluation of the algorithms.



(b) Results of the algorithms applied to the artificial spectrum and to a non-chirped version of it.
Figure 3: Results of the simulated spectra and their resulting OCT signals.

In general the results are as expected. Faster algorithms show in most cases more artefacts. However, the NFFT in general shows less artefacts than an iFFT of similar speed. One should keep in mind that for real world situations the data precision is not limited by numerical accuracy but by noise generated by statistical processes and electronics. Thus, depending on the SNR of the OCT device at hand the numerical artefacts are not seen and covered by real noise.

4. CONCLUSION

It has been shown that there are viable alternatives to the standard interpolated FFT when dealing with chirped spectra to gain OCT images. Depending on the device/camera at hand that is used for acquiring the spectra, the object one wants to image and the acquisition speed an algorithm needs to be selected. Whereas for low speed and high SNR devices one can simply use an algorithm yielding optimal image quality in real time the situations is somewhat more difficult for high speed OCT devices. In this case one needs to balance processing speed and image quality. In future work we will show how these algorithms perform for various real-world OCT devices.

REFERENCES

[1] D. Potts, G. S. and Tasche, M., "Fast fourier transforms for nonequispaced data: A tutorial," in [*Modern Sampling Theory: Mathematics and Applications*], Benedetto, J. J. and Ferreira, P., eds., 249–274, Birkhaeuser, Boston (2001).

[2] B. Hofer, B. Povazay, B. H. A. U. G. M. and Drexler, W., "Dispersion encoded full range frequency domain optical coherence tomography," *Opt. Express* **17**, 7–24 (2009).