



Università
della
Svizzera
italiana

Faculty
of
Informatics

Bachelor Thesis

May 28, 2023

Spectrally Accurate Resampling of High Quality Rotated Images

Dylan Reid Ramelli

Abstract

Advisor
Prof Rolf Krause
Co-Advisors
Dr Diego Rossinelli, Dr Patrick Zulian

Advisor's approval (Prof Rolf Krause):

Date:

Figure 1. Unit circle with $N = 3$,

1 Introduction, Motivation

For many years the amount of digital imaging data has been increasing exponentially (citation) and as such the need for acquiring this large data and process it with care has become a central point of focus. The goal of this project is to process high quality signals, in this case images, by rotating them using techniques such as the Fourier Transform, Shift Theorem, FIR filtering and CUDA parallel processing. The accurate rotation of images is very useful in many applications such as Data Augmentation for example. For Data augmentation we want to increase the coverage of a certain dataset. In Convolutional Neural Networks we can augment the dataset by rotating the input images by some random rotation and then feeding them to the model to improve it without having to collect more data externally.

Another application of this technique can be seen in Data Visualization for multi-modal imaging. In this case we can either acquire images at different times and then combine them together so that we can manipulate the images by rotation for example to align them correctly. Go into more detail.. ask Diego.

In our project we will use a method of rotation which was described in the article titled "Convolution-based interpolation for fast, high-quality rotation of images"[1] where the rotation matrix

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

is refactorized as three matrices each of which represents a shearing in a cardinal direction.

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} 1 & -\tan\theta/2 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ \sin\theta & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & -\tan\theta/2 \\ 0 & 1 \end{pmatrix}$$

The first matrix shears the image in the x direction by $\Delta_x = -y \cdot \tan(\theta/2)$, the second matrix shears the image in the y direction by $\Delta_y = x \cdot \sin(\theta)$ and the last matrix shears again in the x direction by Δ_x .

Given this definition of a 2D rotation we can perform the operation as a sequence of three 1D translations on the 1D input array that represents our image.

"Explain 2D to 3D."

FT: Direct solver instead of iterative, takes in a lot of data.

2 State of the art

For certain applications such as... it is required to be able to rotate images while maintaining the highest quality possible. To achieve satisfying results the general approach is to use bilinear and nearest neighbour interpolation.

Include images from Diego of blood vessels in the heart if I think

3 Project Requirements

To complete this project we made use of different tools such as, Python, Jupyter Notebook, C/C++ and CUDA.

3.1 Frequencies of a signal

$$freq_n = \lfloor \frac{n}{2} \rfloor + 1 \quad (1)$$

Visual representation with $n = 3$ of the discrete Fourier transform of a function $g(x)$

$$G_m = \sum_{k=0}^{N-1} g(k) e^{-i \frac{2\pi}{N} km}$$

To shift a signal by a fractional amount it is imperative to use the correct frequency when doing so.

Given that for n samples we have $\lfloor \frac{n}{2} \rfloor + 1$ distinct frequencies

Euler's Formula

$$e^{ix} = \cos(x) + i\sin(x) \quad (2)$$

Useful property:

$$\cos(w) = \frac{1}{2}(e^{iw} + e^{-iw}) \quad (3)$$

3.2 Fourier Transform with Shift Theorem

Here we derive the shift theorem for a discrete signal starting from the normal Fourier Transform with g_k being our 1D input array, N the number of samples and m the frequency:

$$G_m = \sum_{k=0}^{N-1} g_k[k] e^{-i \frac{2\pi}{N} km}$$

Now instead of $g_k[k]$, we want $g_k[k - \delta]$ where δ is the amount we want to shift. So the above equation can be rewritten as:

$$Z_m = \sum_{k=0}^{N-1} g_k[k - \delta] e^{-i \frac{2\pi}{N} km}$$

$$Z_m = \sum_{r=0-\delta}^{N-1-\delta} g_k[r] e^{-i \frac{2\pi}{N} km}, r = k - \delta$$

Since $r = k - \delta$ then $k = r + \delta$, and as such:

$$Z_m = \sum_{r=-\delta}^{N-1-\delta} g_k[r] e^{-i \frac{2\pi}{N} (r+\delta)m}$$

We can then separate the exponential:

$$Z_m = \sum_{r=-\delta}^{N-1-\delta} g_k[r] e^{-i \frac{2\pi}{N} rm} e^{-i \frac{2\pi}{N} \delta m}$$

And factor it out of the sum:

$$Z_m = e^{-i \frac{2\pi}{N} \delta m} \sum_{r=-\delta}^{N-1-\delta} g_k[r] e^{-i \frac{2\pi}{N} rm}$$

The sum now has exactly the same range as before:

$$Z_m = e^{-i \frac{2\pi}{N} \delta m} \sum_{k=0}^{N-1} g_k[k] e^{-i \frac{2\pi}{N} km}$$

$$Z_m = e^{-i \frac{2\pi}{N} \delta m} G_m = H_m \cdot G_m \quad (4)$$

Here is a small example made in python:

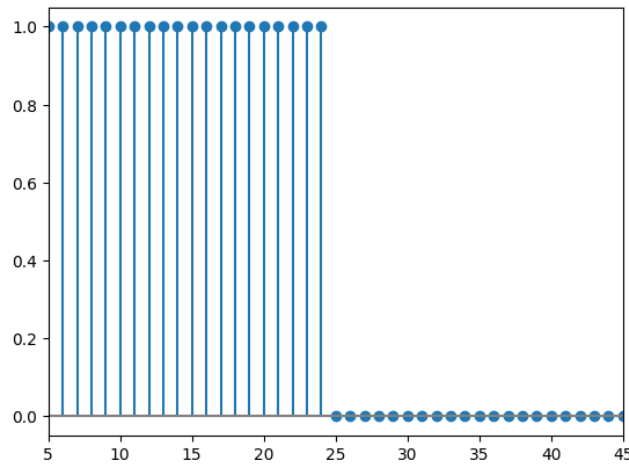


Figure 2. Original 1D function.

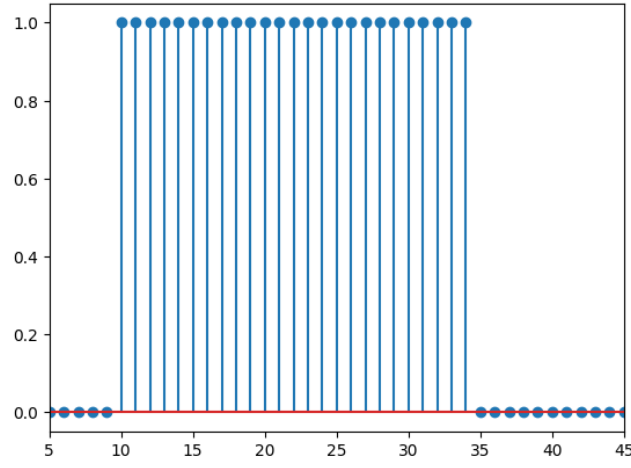


Figure 3. Function shifted by $\delta = 10$

The above equation works well for non fractional shifts but has a certain problem when we want to shift by a fractional amount. "Explain why".

To solve this we need to take into consideration the negative frequencies and in particular if the number of samples is odd or even. Normally we multiply each sample by its corresponding phase, which is based on the frequency number m but since we need to take into consideration the negative frequencies we can define a function called wavenum that returns the correct frequency index to use in the phase shift:

$$wave_n(m) = (m + N/2) \bmod N - (N/2)$$

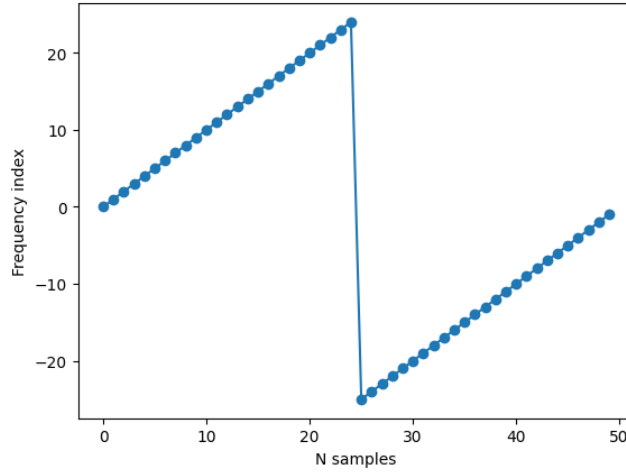


Figure 4. Wavenum function with $N = 50$

$$H_m = e^{-i \frac{2\pi}{N} \delta wave_n(m)}$$

$$Z_m = H_{wave_n(m)} \cdot G_m \quad (5)$$

3.3 Lanczos Filter and FIR

To alleviate the problems that arise when shifting a signal by a fractional amount we can use a filter L that is convoluted with the shifted signal.

$$L_m = \frac{\sin(a)}{a} \Big|_{a=\frac{2\pi wave_n(m)}{N}}$$

Apply lanczos filter to fractional shift to smooth the result.

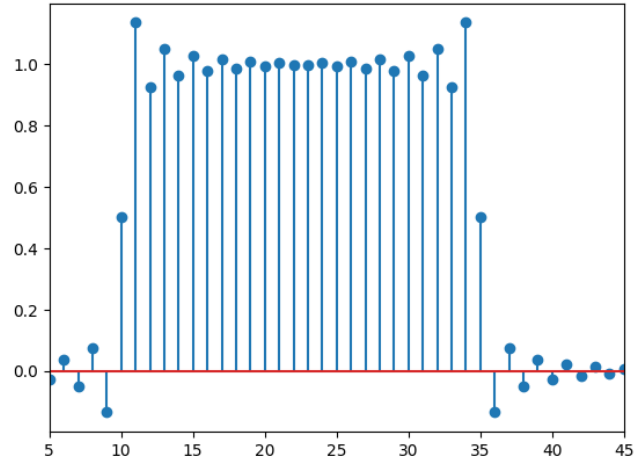


Figure 5. Shifted function by $\delta = 10.5$

$$Z_m = L_m \cdot H_m \cdot G_m$$

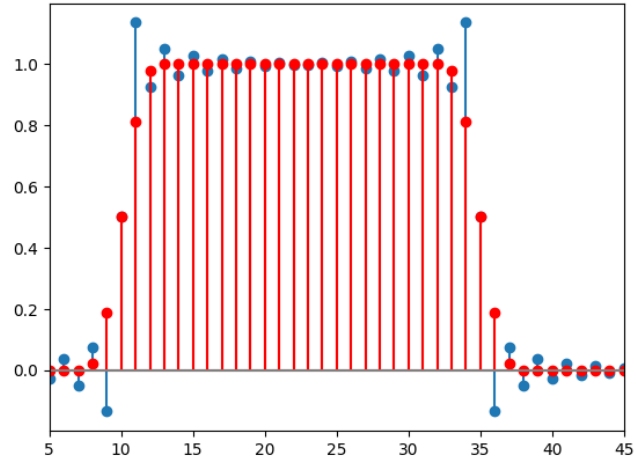


Figure 6. Shifted function by $\delta = 10.5$ with lanczos filter applied (red), no filter(blue).

4 Project Design

Explain 1D,2D,3D transposition. Simplification of 2D rotation using 1D translations.

5 Implementation

- Problems while implementing the code
- Indexing frequencies...
- practical problems...

6 Result

- Rotate image many times like in the paper
- look for artefacts
- compare with C code `gather_noloss.c`
- quality, quantity

Plan



Motivation:

Image

Image

Image

Image

Week	Activity	Duration in Weeks	Status
W1	Finish all 2D image rotations in C and start rotations with CUDA.	1W	COMPLETED
W2	Complete 2D Rotations with CUDA and read/fully understand research article "Convolution-based Interpolation for fast, high quality Rotation of Images."	1W	DONE. CUDA rotations are not complete for gather no loss. Need to understand better when you can call kernels and where.
W3	Meet and ask questions about research article, keep learning paper. Finish gatherloss in C, keep looking at CUDA examples.	1W	Started summarizing the article and also a reference book about digital signal processing, which is helping a lot in understanding the article. DONE
W4	Fully implement CUDA code for gatherloss and fully understand article.	1W	Able to implement CUDA code for gatherloss still need this week for article. DONE
W5	Plot a sinusoid function around multiple circles to create a n image to serve as test for rotations. Understand how to translate an array by a fractional value	1W	DONE
W6	Progress Report , start writing report, start with abstract. Continue with translate signal.	1W	NOT DONE
W7	Start implementing interpolation using 3 pass algorithm in 2d test image	2-3W	IN PROGRESS
W8	Continue implementing interpolation using 3 pass algorithm in 2d test image and start looking into parallelizing the code with either MPI or CUDA	1W	CANCELLED
W9	Continue implementing interpolation using 3 pass algorithm in 2d test image	1W	IN PROGRESS
W10	Continue implementing interpolation using 3 pass algorithm in 2d test image ... and finish report	2-3W	IN PROGRESS
W11	Continue implementing interpolation using 3 pass algorithm in 2d test image ... and finish report	1W	IN PROGRESS
W12	Maybe add CUDA or MPI support and finish report	1W	IN PROGRESS

Table 1. Plan

7 Solution

8 Validation

9 Conclusion

- Summary
- limitations

References

- [1] IEEE Philippe Thévenaz Michel Unser, Senior Member and Leonid Yaroslavsky. Convolution-based interpolation for fast, high-quality rotation of images.