

The Fourier Transform and Signal Processing

Cain Gantt

Advisor: Dr. Hong Yue

Abstract

In this project, we explore the Fourier transform and its applications to signal processing. We begin from the definitions of the space of functions under consideration and several of its orthonormal bases, then summarize the Fourier transform and its properties. After that, we discuss the Convolution Theorem and its relationship to the physics behind problems in signal processing. Finally, we investigate the multidimensional Fourier transform; in particular, we consider the 2-dimensional transform and its use in image processing and other problems. We include an example of a typical image processing task and demonstrate how the Convolution Theorem is applied to obtain a solution.

1 Function Properties

We begin by stating the properties of the functions that we will investigate and by providing appropriate definitions [1]. The functions we are considering are piecewise-continuous complex-valued functions of real variables; that is, functions mapping values in \mathbb{R} or subsets thereof into the complex plane. A function is piecewise continuous if it contains no infinite discontinuities, and each finite subinterval of its domain contains a finite number of discontinuities. We require piecewise continuity for its favorable properties with respect to integration, as will become clear.

We will also investigate periodic functions. These are functions that have the property that $f(x) = f(x + Tn)$ for each x in the domain of the function and for each integer n . The positive real value T is such that it is the smallest such value to satisfy this property; we say the function has a period of T , or that the function is T -periodic.

Using this definition, each function defined on some interval I with length T , such as $[0, T]$ or $[-\frac{T}{2}, \frac{T}{2}]$, the domain of the function can be extended to all of \mathbb{R} and made T -periodic.

If the function is continuous on I , its periodic extension is continuous on \mathbb{R} if its values at the left and right endpoints of I are equal. Otherwise, the function will be piecewise-continuous on \mathbb{R} .

Lastly, the set E is defined by [1] to be the set of piecewise-continuous complex-valued 1-periodic functions on the interval $[-\frac{1}{2}, \frac{1}{2}]$. We will consider this set and appropriate subsets as vector spaces of functions with respect to different inner products.

2 Vector Space of Real Periodic Functions

Taking from [1], we first consider the subset of E consisting of real-valued functions $f : [-\frac{1}{2}, \frac{1}{2}] \rightarrow \mathbb{R}$, and define an inner product for this subspace as

$$\langle f, g \rangle = 2 \int_{-\frac{1}{2}}^{\frac{1}{2}} f(x) \overline{g(x)} dx,$$

where $f, g \in E$ are real-valued, and $\overline{g(x)}$ denotes the complex conjugate of $g(x)$. Note that, for any real-valued function f , we have $\overline{f(x)} = f(x)$. This subspace of real-valued functions is spanned by the set of functions

$$\left\{ \frac{1}{\sqrt{2}}, \cos(2\pi x), \sin(2\pi x), \cos(4\pi x), \sin(4\pi x), \dots \right\} = \left\{ \frac{1}{\sqrt{2}}, \cos(2\pi n x), \sin(2\pi n x) \right\}_{n=1}^{\infty}.$$

The first of these functions accounts for a vertical shift, while each cosine and sine describe the even and odd portions of a particular frequency. Together, the pair $\cos(2\pi n x)$ and $\sin(2\pi n x)$ can be thought of as describing a single sinusoid of period $\frac{1}{n}$ that may be horizontally translated (or *phase shifted*, as this is referred to in many physical applications). This can be easily verified through the use of the angle sum formulas for cosine or sine.

Note that each of these functions is orthogonal to each other function, since the inner product between any two distinct functions in the set is equal to 0. Each function is also a unit vector since the inner product of each function with itself is equal to 1. Thus, this set forms an orthonormal basis for the subspace of real-valued functions on E .

Each function in this subspace can be represented by a linear combination of these basis vectors as follows:

$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(2\pi n x) + b_n \sin(2\pi n x)],$$

where the coefficients

$$a_n = \langle f(x), \cos 2\pi nx \rangle = 2 \int_{-\frac{1}{2}}^{\frac{1}{2}} f(x) \cos(2\pi nx) dx \quad \text{for } n = 0, 1, 2, \dots$$

$$b_n = \langle f(x), \sin 2\pi nx \rangle = 2 \int_{-\frac{1}{2}}^{\frac{1}{2}} f(x) \sin(2\pi nx) dx \quad \text{for } n = 1, 2, 3, \dots$$

This representation of the function is the *real Fourier series* of f . Note that equality does not necessarily hold without considering the convergence of this infinite summation. Imposing the restriction that f has finite-valued one-sided derivatives at all points $x \in (-\frac{1}{2}, \frac{1}{2})$, including the left-derivative at the right endpoint and vice-versa, is sufficient to provide pointwise convergence for the series on $[-\frac{1}{2}, \frac{1}{2}]$. An equivalent condition is the requirement that f' be piecewise continuous, thus providing $f' \in E$. By Dirichlet's Theorem, for each function $f \in E$ with $f' \in E$, its real Fourier series will converge to the average of the one-sided limits at each point [1]. For points in the domain where f is continuous, the one-sided limits are necessarily equal, and the series converges to the value of the function at that point. At the endpoints, the series will converge to the average of the function's one-sided limit at each endpoint.

As brief aside, consider the case of the vector space of “arrows” in \mathbb{R}^n ; the inner product of a vector with a particular basis vector informally represents how much that vector “points” in the same direction as the basis vector. In our vector space of functions, each real coefficient a_n and b_n analogously represents how much the function f “corresponds with” the cosine or sine with period $\frac{1}{n}$ (or, with frequency n). Another interpretation is that the coefficient describes how much each particular frequency is present in the function.

3 Vector Space of E

Turning our attention to the entire function space E , we will need a set of complex-valued basis vectors. Similar to how real-valued sinusoidal functions can be expressed using a (real) linear combination of a cosine and sine with equal period, we can use Euler's formula $e^{i\theta} = \cos \theta + i \sin \theta$ to express complex-valued periodic functions.

We define an inner product for the vector space E as

$$\langle f, g \rangle = \int_{-\frac{1}{2}}^{\frac{1}{2}} f(x) \overline{g(x)} dx.$$

One orthonormal basis for E is the set of complex exponentials

$$\{e^{i2\pi nx}\}_{n=-\infty}^{\infty} = \{\dots, e^{-i4\pi x}, e^{-i2\pi x}, 1, e^{i2\pi x}, e^{i4\pi x}, \dots\}.$$

For a particular complex exponential, varying x corresponds to a rotation around the unit circle in the complex plane. With this interpretation, the functions containing positive integer values for n produce counterclockwise rotation, and those with negative n produce clockwise rotation.

The *complex Fourier series* of a function $f \in E$ with $f' \in E$ is given by

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{i2\pi nx}$$

Again, each coefficient c_n is the inner product of f with the appropriate complex exponential:

$$c_n = \langle f(x), e^{i2\pi nx} \rangle = \int_{-\frac{1}{2}}^{\frac{1}{2}} f(x) \overline{e^{i2\pi nx}} dx = \int_{-\frac{1}{2}}^{\frac{1}{2}} f(x) e^{-i2\pi nx} dx \quad (1)$$

As with the real Fourier series, these complex coefficients c_n can be thought of as describing the frequency content of the function f . Taking cues from the field of complex analysis, the modulus of c_n conveys information about the amplitude of that sinusoid, and its argument conveys information about the phase. This effectively encodes all of the necessary information about the periodic function.

4 The Fourier Transform

To motivate the formulation of the Fourier transform, we pose the question: *Given a real-valued function f , can we create a new function to describe its frequency content?* Fortunately, the coefficients of the complex Fourier series lead us towards a solution. Changing our perspective, given some $f \in E$, the complex inner product (1) for E can be thought of as a mapping from an integer n , representing a particular frequency, to a complex value c_n that contains amplitude and phase information of the complex sinusoid that corresponds to that frequency present in f .

However, since this integral is only taken over $[-\frac{1}{2}, \frac{1}{2}]$, it is only able to capture information about f on that interval. If we wish to consider functions f that are defined on all of \mathbb{R} , we could change the inner product to an improper integral from $-\infty$ to ∞ . This allows

the integral to capture information over the entire real line; however, in doing so, we must revisit the convergence of the integral (1). Obtaining finite values for each c_n is desirable, so we desire the integrand $f(x)e^{-i2\pi nx}$ to be absolutely integrable. Since $|e^{-i2\pi nx}| = 1$, that f is absolutely integrable on \mathbb{R} is a sufficient condition to provide

$$\int_{-\infty}^{\infty} |f(x)e^{-i2\pi nx}| dx < \infty.$$

Lastly, to describe all frequencies present in the function, rather than only those with integer value, we can change from the discrete variable $n \in \mathbb{Z}$ to a continuous variable $\xi \in \mathbb{R}$.

With all of the pieces in place, the Fourier transform of a function $f(x)$ is defined by [1] as

$$F(\xi) = \mathcal{F}[f(x)] = \int_{-\infty}^{\infty} f(x)e^{-i2\pi x\xi} dx. \quad (2)$$

This is sometimes referred to as the *forward* Fourier transform, and we will refer to it as such. Similarly, the *inverse* Fourier transform of $F(\xi)$ is defined to be

$$f(x) = \mathcal{F}^{-1}[F(\xi)] = \int_{-\infty}^{\infty} f(\xi)e^{i2\pi x\xi} d\xi. \quad (3)$$

The forward transform takes in a function $f : \mathbb{R} \rightarrow \mathbb{C}$ and produces a function $F : \mathbb{R} \rightarrow \mathbb{C}$. Naturally, its inverse produces f given F . To make better sense of the relationship between these variables x and ξ , we turn back to the physical interpretation of these two functions.

We can model a spatial wave-function (such as the peaks and troughs along the cross section of a body of water) with a function f that relates the position x to the wave's height $f(x)$. This gives x units of length or distance. For the exponential function to have a dimensionless argument, we give ξ units of reciprocal distance, which describes cycles per unit distance. This describes the *spatial frequency* of the wave. The transformed function $F(\xi)$ takes a particular spatial frequency as input and returns a complex number that describes the amplitude and phase of the spatial wave with that frequency that best corresponds with f .

Another interpretation of domains for the functions f and F are time and temporal frequency, respectively. For example, an audio signal can be modeled as a function of time, $f(t)$, and its transform $F(\nu)$ describes the frequency content of the audio. If t is given in seconds, then ν has units of Hertz. There are many other pairs of variables that can be related in this manner. In the field of quantum physics, the Fourier transform relates the

position of a particle to its momentum. Pairs of variables that are related in this way are sometimes referred to as *conjugate variables*.

5 Properties of the Fourier Transform

In our studies of the transform, it was particularly interesting to see the relationship between two actions on a function through the Fourier transform. Perhaps the most straightforward is the linearity of the transform, such that

$$\mathcal{F}[af + bg](\xi) = aF(\xi) + bG(\xi)$$

This follows from the linearity of the integral used in the definition. The transform of purely real- or imaginary-valued functions f also displays interesting symmetries. For instance, if f is real-valued, then $F(-\xi) = \overline{F(\xi)}$. Similarly, the parity of the function f reveals additional symmetries. Exploiting these symmetries can reduce the amount of computation required to obtain F .

More interestingly, given $c \in \mathbb{R}$, the two relationships

$$\mathcal{F}[e^{i2\pi cx} \cdot f(x)](\xi) = F(\xi - c) \quad \text{and} \quad \mathcal{F}[f(x - c)](\xi) = e^{-i2\pi c\xi} \cdot F(\xi)$$

indicate that a horizontal translation in one domain corresponds to a complex rotation in the other domain. Again, since the complex values of F describe the amplitude and phase, this interpretation provides that a horizontal translation (phase shift) corresponds to a change in the phase angle of the sinusoids describing the function.

Perhaps the most powerful property of the Fourier transform for signal processing applications is given in the Convolution Theorem [1]. Simply put, it states that the transform (forward or inverse) of the convolution of two functions is equivalent to the product of their transforms. This is equivalent to the following two statements:

$$\mathcal{F}[f * g] = \mathcal{F}[f] \cdot \mathcal{F}[g] = F \cdot G \tag{4}$$

$$\mathcal{F}^{-1}[F * G] = \mathcal{F}^{-1}[F] \cdot \mathcal{F}^{-1}[G] = f \cdot g \tag{5}$$

where

$$(f * g)(x) = \int_{-\infty}^{\infty} f(y) \cdot g(x - y)dy \tag{6}$$

denotes the convolution of any two complex-valued functions f and g of real variables. Though the convolution of two functions is demanding and often has few symmetries aiding in computation, the Convolution Theorem allows the convolution of two functions to be computed through transforming two functions, performing pointwise multiplication, and taking the inverse transform of the resulting product.

6 Multidimensional Fourier Transform

To consider functions of multiple variables, we first consider our variables \mathbf{x} and $\boldsymbol{\xi}$ as elements of \mathbb{R}^n , such that

$$\begin{aligned}\mathbf{x} &= (x_1, x_2, \dots, x_n), \text{ and} \\ \boldsymbol{\xi} &= (\xi_1, \xi_2, \dots, \xi_n).\end{aligned}$$

Utilizing the dot product on \mathbb{R}^n , we find

$$\mathbf{x} \cdot \boldsymbol{\xi} = x_1 \xi_1 + x_2 \xi_2 + \dots + x_n \xi_n \in \mathbb{R} \quad (7)$$

The multidimensional Fourier transform and its inverse transform relate the functions $f(\mathbf{x})$ and $F(\boldsymbol{\xi})$ as follows:

$$F(\boldsymbol{\xi}) = \mathcal{F}[f(\mathbf{x})] = \int_{\mathbb{R}^n} f(\mathbf{x}) e^{-i2\pi \mathbf{x} \cdot \boldsymbol{\xi}} d\mathbf{x} \quad (8)$$

$$f(\mathbf{x}) = \mathcal{F}^{-1}[F(\boldsymbol{\xi})] = \int_{\mathbb{R}^n} F(\boldsymbol{\xi}) e^{i2\pi \mathbf{x} \cdot \boldsymbol{\xi}} d\boldsymbol{\xi} \quad (9)$$

To better understand how this is computed, we expand the dot product in the exponent using (7) to obtain

$$\begin{aligned}F(\boldsymbol{\xi}) &= \int_{\mathbb{R}^n} f(\mathbf{x}) e^{-i2\pi \mathbf{x} \cdot \boldsymbol{\xi}} d\mathbf{x} \\ &= \int_{\mathbb{R}^n} f(\mathbf{x}) e^{-i2\pi(x_1 \xi_1 + x_2 \xi_2 + \dots + x_n \xi_n)} d\mathbf{x} \\ &= \int_{\mathbb{R}^n} f(x_1, x_2, \dots, x_n) e^{-i2\pi x_1 \xi_1} e^{-i2\pi x_2 \xi_2} \dots e^{-i2\pi x_n \xi_n} dx_1 dx_2 \dots dx_n \\ &= \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(x_1, x_2, \dots, x_n) e^{-i2\pi x_1 \xi_1} dx_1 \right] e^{-i2\pi x_2 \xi_2} dx_2 \dots e^{-i2\pi x_n \xi_n} dx_n\end{aligned}$$

The innermost integral, written here with respect to x_1 , is simply the one-dimensional Fourier transform with respect to x_1 ; the resulting function is a function of ξ_1, x_2, \dots, x_n . Continuing in this manner, we find that the multidimensional transform can be computed with n independent Fourier transforms, one along each dimension.

7 Image Processing

As we have touched on previously, the Fourier transform has extensive applications to signal processing. A number of signal processing techniques, such as filtering, are modeled using the convolution of two functions. Since the convolution is very computationally intensive, especially so in higher dimensions, it is common practice to take advantage of the Convolution Theorem [2]. If we first compute the transform of each function, multiply the transformed functions, and finally compute the inverse transform of their product, the result is exactly the convolution of the two initial functions.

The greatest advantage comes when considering the symmetries of the Fourier transform, allowing us to perform fewer computations for the forward and inverse transforms. Some of these symmetries, along with memory-management techniques, allow a far more computationally efficient algorithm for computing the Fourier transform on discrete data sets [3, 4].

In the case of image processing, we will consider two-dimensional Fourier transforms. We can model a grayscale image as a function $f(\mathbf{x})$, which assigns an intensity value to each coordinate $\mathbf{x} = (x_1, x_2)$ in the plane. (A full-color image can be treated as a triplet of such functions, one for each of the red, green, and blue channels in the image.) As is often the case when relating our equations to physical phenomena, this will be a real-valued function with a range $[0, 1]$; a value of 0 represents black pixels and 1 represents a white pixel. The transformed function $F(\boldsymbol{\xi})$ assigns a value to each spatial frequency pair $\boldsymbol{\xi} = (\xi_1, \xi_2)$. Here, interpreting these elements as vectors introduces the notion of a direction for the planar wave described by the corresponding sinusoid, as seen in Figure 1.

An illustrative example is beneficial to understanding how the Convolution Theorem applies to image processing. Let $f(\mathbf{x})$ again be a grayscale image. We consider some function $h(\mathbf{x})$ that acts as a filter function which produces a processed image $g(\mathbf{x}) = (f * h)(\mathbf{x})$. With



Figure 1: *Left to right:* Low, intermediate, and high frequency planar waves [5]. The red arrows drawn in image denotes a vector representation of $\boldsymbol{\xi} = (\xi_1, \xi_2)$.

the Convolution Theorem, we find

$$\begin{aligned} g(\mathbf{x}) &= (f * h)(\mathbf{x}) \\ &= \mathcal{F}^{-1} [F \cdot H] \end{aligned} \tag{10}$$

In practice, we care less about $h(\mathbf{x})$ than we do its transform $H(\boldsymbol{\xi})$; instead we can choose a function H to modify the frequency content of f however we desire [6]. For instance, if we wish to blur the image, we can boost its low frequency content relative to its high frequencies. Similarly, sharpening the image corresponds to increasing its high frequency content relative to its low frequency content. It is worth noting that the low frequency vectors closest to $(\xi_1, \xi_2) = (0, 0)$ correspond to the overall brightness of the image. This can be seen in Figure 2.

We now demonstrate the application of (10). We can define a filter function

$$H(\xi_1, \xi_2) = |\xi_1 \xi_2| + 0.3, \tag{11}$$

constructed as a sharpening filter that simultaneously brightens the image. This function H and the resulting function g are shown in Figure 3. Figures 4-7 showcase various other filtering functions and their corresponding output.



Figure 2: *Left:* An example input image, representing $f(\mathbf{x})$. *Right:* The magnitude of the transformed function $F(\boldsymbol{\xi})$; the values have been log-scaled to improve visual contrast of the values close to zero.



Figure 3: Processed image (left) produced by filter $H(\boldsymbol{\xi}) = |\xi_1\xi_2| + 0.3$ (right).

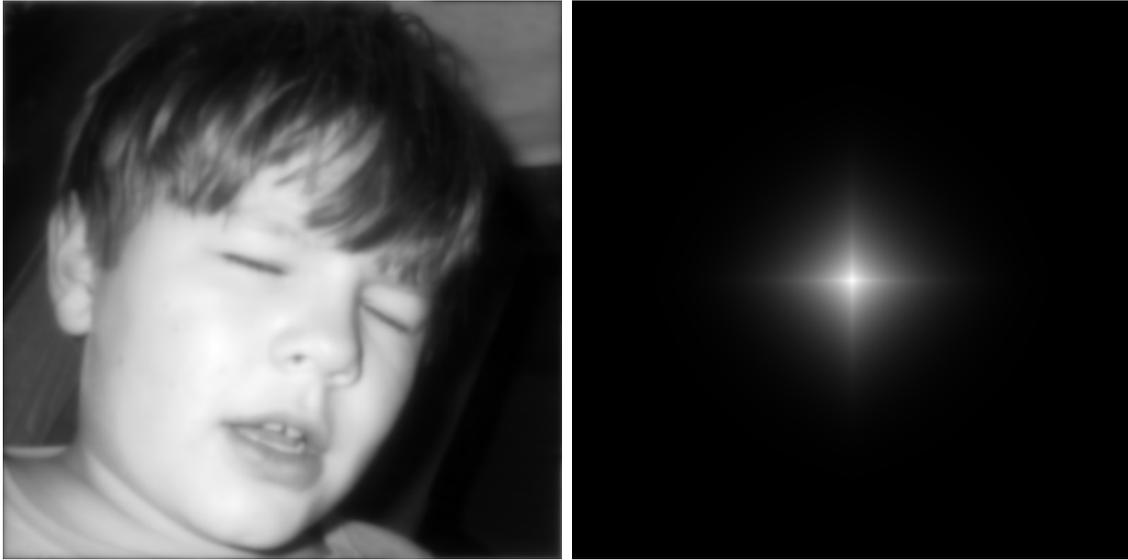


Figure 4: Processed image (left) produced by filter $H(\boldsymbol{\xi}) = [(1 - |\xi_1|)(1 - |\xi_2|)]^5$ (right).

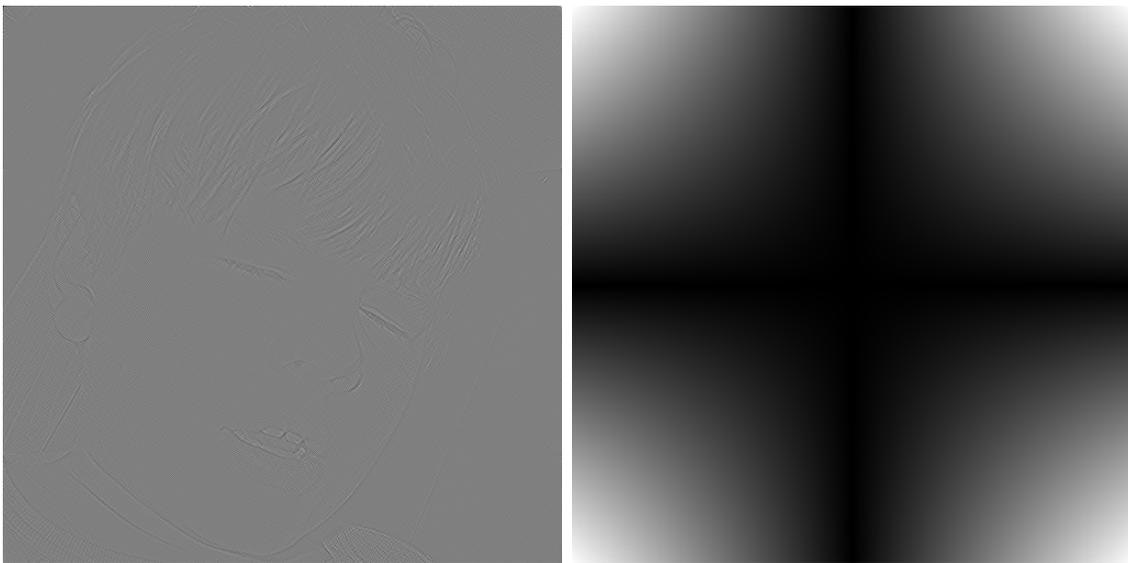


Figure 5: Processed image (left) produced by filter $H(\boldsymbol{\xi}) = |\xi_1 \xi_2|$ (right).



Figure 6: Processed image (left) produced by filter $H(\boldsymbol{\xi}) = (1 - |\xi_2|)^{10}$ (right).



Figure 7: Processed image (left) produced by filter $H(\boldsymbol{\xi}) = (1 - |\xi_1|)^{10}$ (right).

8 Beyond Image Processing

Image processing problems similar to those posed in eqn. (10) are solved by designing a filter function H such that the resulting image $g(\mathbf{x})$ has certain desirable properties. A far more formidable set of problems are inverse problems [7]. Better stated, given a function $g(\mathbf{x})$, we wish to undo the effects of convolution by some unknown filtering operation to find the original image $f(\mathbf{x})$.

There are many physical processes which can be modeled with a convolution that we may wish undo through solving such an inverse problem. A simple one-dimensional case is the reconstruction of an audio signal that has been corrupted by noise or other artifacts. In imaging, we may wish to correct a photograph that was taken out-of-focus. Many portions of medical imaging, especially 3-dimensional imaging techniques such as computed tomography (CT) or magnetic resonance imaging (MRI), are dependent on undoing a convolution introduced by the physical imaging process. The resulting function obtained is a three-dimensional reconstruction of the imaging field [8].

The solutions to many inverse problems can sometimes be ill-posed since artifacts and noise introduced by audio recording or imaging can include non-linear terms [7]. These inverse problems are then solved as optimization problems, where we wish to find a reconstructed function with the understanding that, in most cases, it will not be identical to the original function. For any hope of a useful solution, it is essential to have a thorough understanding of the physical processes being modeled.

References

- [1] Allan Pinkus and Samy Zafrany. *Fourier Series and Integral Transforms*. Cambridge University Press, 1997.
- [2] William H. Press. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, sep 2007.
- [3] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.
- [4] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on “Program Generation, Optimization, and Platform Adaptation”.
- [5] B. Osgood. *Lecture Notes for EE 261: The Fourier Transform and its Applications*. Stanford Engineering Everywhere, 2007.
- [6] G. L. Zeng. Revisit of the Ramp Filter. *IEEE Transactions on Nuclear Science*, 62(1):131–136, Feb 2015.
- [7] M. T. McCann, K. H. Jin, and M. Unser. A review of convolutional neural networks for inverse problems in imaging. 2017.
- [8] A. Myagotin, A. Voropaev, L. Helfen, D. Hänschke, and T. Baumbach. Efficient volume reconstruction for parallel-beam computed laminography by filtered backprojection on multi-core clusters. *IEEE Transactions on Image Processing*, 22(12):5348–5361, Dec 2013.

A MATLAB Script

```
% predefined image size
s = 512;

% load image file
input = imread('imgs/input.png');

% take 2D FFT
freq = fft2(input,s,s);

% create absolute value arrays
lo = ifftshift([0:(s/2-1) (s/2):-1:1]./(s/2));
hi = ifftshift([(s/2):-1:1 0:(s/2-1)]./(s/2));

% create 2D filter functions
filter_horz = repmat(lo,s,1);
filter_vert = repmat(lo',1,s);

filter_edge = hi'*hi;
filter_shrp = hi'*hi + 0.3;

filter_blur = lo'*lo;
filter_blur10 = filter_blur.^10;

% multiply and inverse transform
post_vert = ifft2(filter_vert.^10 .* freq);
post_horz = ifft2(filter_horz.^10 .* freq);

post_shrp = ifft2(filter_shrp .* freq);
post_edge = ifft2(filter_edge .* freq);

post_blur = ifft2(filter_blur .* freq);
post_blur5= ifft2(filter_blur.^5 .* freq);
```

```

% postprocess to normalize [0,1]
post_freq = abs(freq) - min(min(abs(freq)));
post_freq = post_freq ./ max(max(post_freq));

post_flog = log(abs(freq));
post_flog = post_flog - min(min(post_flog));
post_flog = post_flog ./ max(max(post_flog));

post_vert = post_vert - min(min(post_vert));
post_vert = post_vert ./ max(max(post_vert));

post_horz = post_horz - min(min(post_horz));
post_horz = post_horz ./ max(max(post_horz));

post_shrp = post_shrp - min(min(post_shrp));
post_shrp = post_shrp ./ max(max(post_shrp));

post_edge = post_edge - min(min(post_edge));
post_edge = post_edge ./ max(max(post_edge));

post_blur = post_blur - min(min(post_blur));
post_blur = post_blur ./ max(max(post_blur));

post_blur5 = post_blur5 - min(min(post_blur5));
post_blur5 = post_blur5 ./ max(max(post_blur5));

```