

# Frequency Domain Image Filtering

## Table of Contents

Frequency Domain Image Filtering.....	1
Table of Contents.....	1
Introduction.....	1
Fourier Analysis.....	1
Example Image Transform .....	1
Index .....	3

## Introduction

This paper documents software used to implement an frequency type of filtering for images. The inspiration is the FK (frequency/wavenumber) techniques used in seismic. That algorithm is designed to remove artifacts that have a slope that is different than the slope of the structure. In this way we can design filters based on the slope in the frequency domain, and so filter out artifacts that do not follow the general trend of the structure.

The algorithm is usually applied in the “I” (intensity) domain, not in the RGB domain. This is based on the assumption: The structure is revealed in the intensity domain, while various colors reflect changes in minerals that are not structural in nature. Obviously this can be a specious assumption, but it nonetheless useful.

The software used includes two libraries:

- (1) A Fourier Transform library called FFTW (Fastest Fourier Transform in the West), described at [fftw.org](http://fftw.org).
- (2) A C++ Image Processing Library, available at <http://home.nycap.rr.com/qzhao/src.zip>.

## Fourier Analysis

Given a series of samples from the region  $[0, M - 1] \times [0, N - 1]$  we can apply a 2D (two-dimensional) Fourier Transform. The theory of that transformation is available [here](#).

We wrote a program that will make an FFT of an image. We call that program [fftImage](#). That program, when given an image calculates the 2D FFT of the image.

## Example Image Transform

We consider the image 99\_0\_46 from the SAFL tank. Here is a picture:

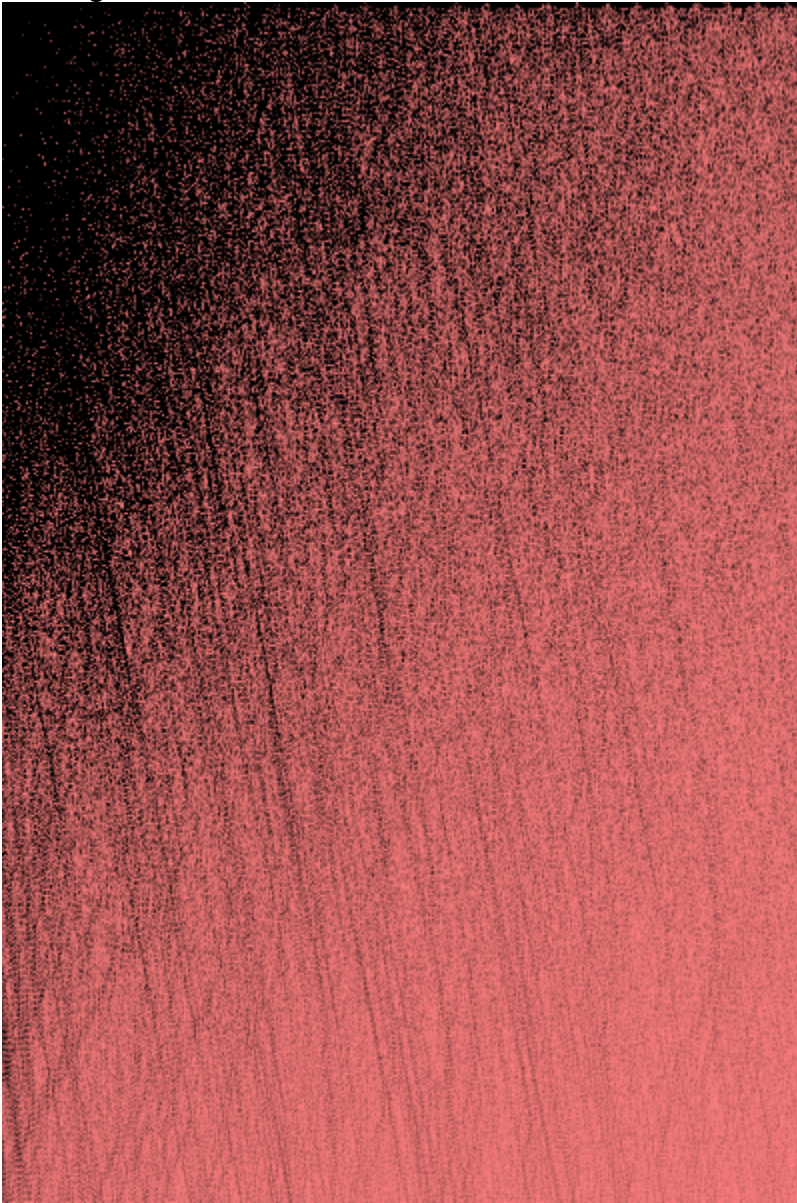


FftImage for SAFL

We can transform that image using the 2D FFT, and we see:



Not very interesting. You should note that the width is half as many pixels. This is due to a real to complex number conversion, that stores information in the amplitude and phase, and we use amplitude for intensity. The frequencies are laid out as: Low frequencies in the upper left corner, and higher to the right and down. The bottom half is basically a repeat of the top half of information.



You can now see the diagonal slashes representing energy at various slopes.

In general, the slope of an event determines the slashes you see.

For example, a completely horizontal event will appear as a slash along the top, and a completely vertical event will be a slash along the left hand side. We use such alignments in a dip filter.

If we so desire, we can apply such dip filters to the data to remove data identified as noise.

We can also simple make the lowest 2 or 3 frequencies equal between sections as a crude AGC.

## Index

AGC .....	1
-----------	---