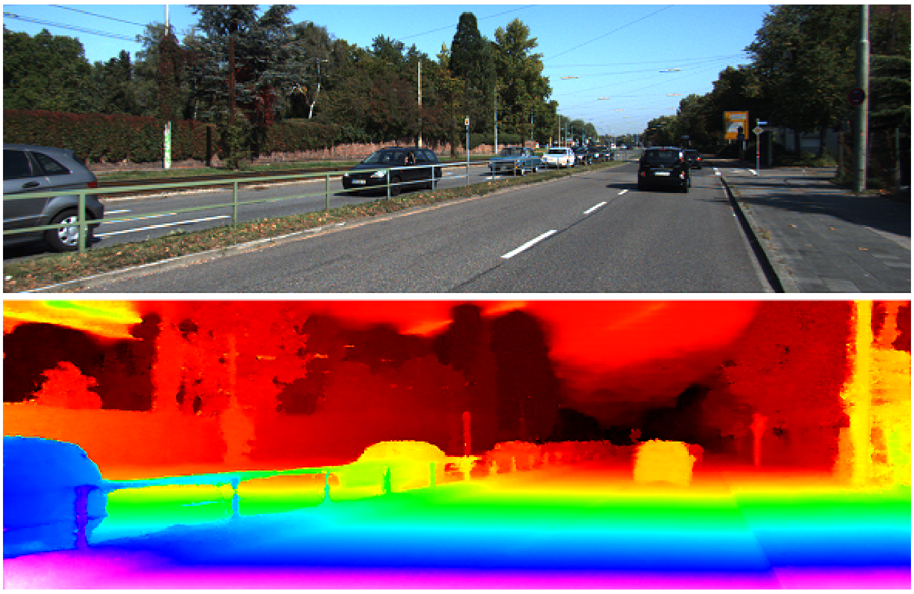
SHANDONG UNIVERSITY, SCHOOL OF MATHEMATICS AND STATISTICS

# Fast Fourier Transform with Image Processing

Shijie Huang 201900830110

June 19, 2021


## ABSTRACT



With the rising demand of autonomous driving, depth estimation algorithm stepped into our eyes. However, while the car is driving, the camera estimating depth infront of the car cannot always maintain a perfect shooting effect. Sometimes due to the interference of smog and mud, the captured image will have larger noise.

We had already published a paper[1] that showing a good method to estimating the depth in a single image, which is in a machine learning way. This paper is going to explore the possibility using FFT to reduce the bad infulence from noised images before we estimating the depth.

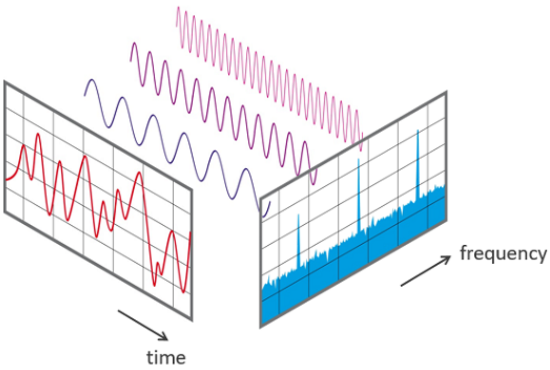



Figure 1.1: Time domain and Frequency domain

# 1 Brief Introdcution of FFT

## 1.1 What is FFT?

The "Fast Fourier Transform" (FFT) is an important measurement method in the science of audio and acoustics measurement. It converts a signal into individual spectral components and thereby provides frequency information about the signal. FFTs are used for fault analysis, quality control, and condition monitoring of machines or systems.
Strictly speaking, the FFT is an optimized algorithm for the implementation of the "Discrete Fourier Transformation" (DFT). A signal is sampled over a period of time and divided into its frequency components. These components are single sinusoidal oscillations at distinct frequencies each with their own amplitude and phase. [2]

# 2 FFT in Image

## 2.1 How it works in Image?

As we are only concerned with digital images, we will restrict this discussion to the *Discrete Fourier Transform* (DFT).
The DFT is the sampled Fourier Transform and therefore does not contain all frequencies forming an image, but only a set of samples which is large enough to fully describe the spatial domain image. The number of frequencies corresponds to the number of pixels in the spatial domain image, i.e. the image in the spatial and Fourier domain are of the same size.
For a square image of size NŒN, the two-dimensional DFT is given by:

$$F(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) e^{-\iota 2\pi \left( \frac{ki}{N} + \frac{lj}{N} \right)} \tag{2.1}$$

where $f(a,b)$ is the image in the spatial domain and the exponential term is the basis function corresponding to each point $F(k,l)$ in the Fourier space. The equation can be interpreted as: the value of each point $F(k,l)$ is obtained by multiplying the spatial image with the corresponding base function and summing the result.
The basis functions are sine and cosine waves with increasing frequencies, $i.e. F(0,0)$ represents the DC-component of the image which corresponds to the average brightness and $F(N-1, N-1)$ represents the highest frequency.

In a similar way, the Fourier image can be re-transformed to the spatial domain. The inverse Fourier transform is given by:

$$f(a,b) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l) e^{\imath 2\pi (\frac{k}{N} + \frac{1b}{N})} \tag{2.2}$$

Note the $\frac{1}{N^2}$ normalization term in the inverse transformation. This normalization is sometimes applied to the forward transform instead of the inverse transform, but it should not be used for both.
To obtain the result for the above equations, a double sum has to be calculated for each image point. However, because the Fourier Transform is separable, it can be written as

$$F(k,l) = \frac{1}{N} \sum_{b=0}^{N-1} P(k,b) e^{-\imath 2\pi \frac{1b}{N}} \tag{2.3}$$

where

$$P(k,b) = \frac{1}{N} \sum_{a=0}^{N-1} f(a,b) e^{-\imath 2\pi \frac{ka}{N}} \tag{2.4}$$

Using these two formulas, the spatial domain image is first transformed into an intermediate image using $N$ one-dimensional Fourier Transforms. This intermediate image is then transformed into the final image, again using $N$ one-dimensional Fourier Transforms. Expressing the two-dimensional Fourier Transform in terms of a series of $2N$ one-dimensional transforms decreases the number of required computations.
Even with these computational savings, the ordinary one-dimensional DFT has $N^2$ complexity. This can be reduced to $Nlog_2N$ if we employ the *Fast Fourier Transform* (FFT) to compute the one-dimensional DFTs. This is a significant improvement, in particular for large images. There are various forms of the FFT and most of them restrict the size of the input image that may be transformed, often to $N = 2^n$ where $n$ is an integer. The mathematical details are well described in the literature.
The Fourier Transform produces a complex number valued output image which can be displayed with two images, either with the $real$ and $imaginary$ part or with $magnitude$ and $phase$. In image processing, often only the magnitude of the Fourier Transform is displayed, as it contains most of the information of the geometric structure of the spatial domain image. However, if we want to re-transform the Fourier image into the correct spatial domain after some processing in the frequency domain, we must make sure to preserve both magnitude and phase of the Fourier image.
The Fourier domain image has a much greater range than the image in the spatial domain. Hence, to be sufficiently accurate, its values are usually calculated and stored in float values.[3]

## 2.2 Explanations and Limitations

In most FFT used cases, we process the signal by FFT, get the frequency domain information, we clean the higher part up, then use IFFT to form the signal denoised. But it's not always the case.
In fact, the Fourier transform is all about treating your data as a big vector and projecting it onto a set of (complex-valued) sinusoids that serve as basis functions. Each sinusoid is of a different frequency. The length of the vector in the direction of each basis function tells us something about the "frequency content" of the signal for that given frequency.

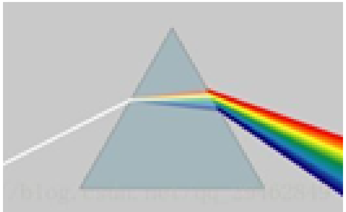


Figure 2.1: Refraction of white light

But what it tells us is not the actual frequency content. If you have a single cosine wave of the same frequency as a basis function, and you push it through the Fourier transform, you'll get what you'd expect: a single spike in the output at the appropriate frequency. But if that cosine wave has a frequency somewhere in between the basis functions, you don't get a spike any more. You get a smear of frequencies across the entire spectrum.
Even though each Fourier basis function consists of a single frequency, in the end they're still just basis functions. If you ascribe any interpretation to the transformed data, aside from "the inner product of my input data with each basis function", then you do so at your own risk. "Frequency Content of the signal" is just an interpretation, and it's a slightly misleading one.

LIMITATIONS When we low-pass filter in order to antialias, we are acting on this misleading interpretation: The original signal is made up of these sine waves, and we "extract" these elements through transformation. But this understanding is wrong, the Fourier transform is only a "interpretation" of the signal through a specific perspective, that is, these sine waves do not actually exist, the decomposition of sine waves is a result of an operation, not a part of the actual signal itself. So there is a information lose if we just cut the higher part off, but unfortunately there is no wonderful mathematical method that can perfectly explain the meaning "noise" in an image, so the defects are inevitable.

# 3 RESULT

## 3.1 Image noise reduction

In this project, I simply keep the 300 x 300 pixels in the center of the frequency domain to eliminate the noise, the reuslts are shown by the Figure 3.1.


Figure 3.1: FFT Denoise

As you can see in Table 3.1, the error of mean square compared to the original image has been decreased nearly tenflod.

| *Method* | **MSE** |
|---|---|
| Gaussian Noise | 20.4513 |
| FFT | 2.8495 |

Table 3.1: FFT MSE

### 3.2 Depth estimating

After we decreased the noise in the image ( which is supported to be a $O(n\log n)$ algorithm ), we can do the depth estimating part. The results are shown by Figure 3.2. (For more details of depth estimation, see [1])

Figure 3.2: Depth Estimation

## References


[1] Huachen Gao, Xiaoyu Liu, Meixia Qu, and Shijie Huang. Pdanet: Self-supervised monocular depth estimation using perceptual and data augmentation consistency. *Applied Sciences*, 11:5383, 6 2021. (document), 3.2

[2] Fast fourier transformation fft - basics. `https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft`. Accessed: 2021-06-19. 1.1

[3] Image transforms - fourier transform. `https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm`. Accessed: 2021-06-19. 2.1