Visual Computing - Notes Week 4

Ruben Schenk, ruben.schenk@inf.ethz.ch

October 26, 2021

5. Fourier Transform

5.1 Introduction

The idea of the **Fourier Transform** is to represent functions in a new basis. We might think of functions as vectors, with many components, where we can apply a linear transformation to transform the basis (i.e. a dot product with each basis elements).

In the expressions, u and v select the basis elements, so a function of x and y becomes a function of u and v. The **basis elements** have the form $e^{-i2\pi(ux+vy)} = \cos 2\pi(ux+vy) - i \sin 2\pi(ux+vy)$. Or:

$$F(g(x, y))(u, v) = \int\int_{\mathbb{R}^2} g(x, y) e^{i2\pi(ux+vy)} \mathrm{d}x\mathrm{dy}$$

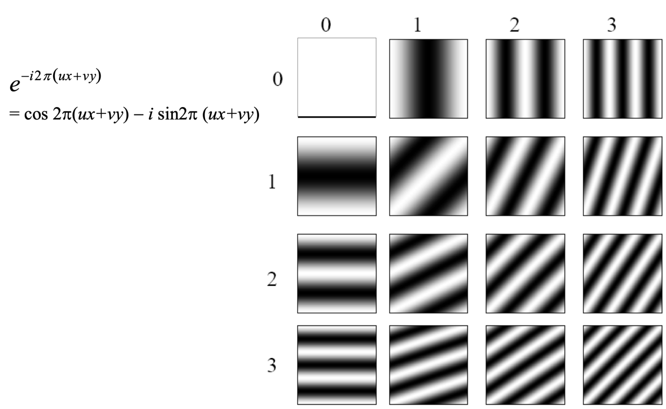
The **discrete Fourier transform** therefore is of the form:

$$F = Uf$$

where:

- F is the transformed image
- U is the Fourier transform base
- f is the vectorized image

5.2 Fourier Basis Functions



5.3 Phase and Magnitude

The Fourier transform of a real function is complex. It's difficult to plot and to visualize, and instead, we can think of the phase and magnitude of the transform.

The **phase** is the phase of the complex transform, and the **magnitude** is the magnitude of the complex transform. An interesting fact is that all natural images have about the *same magnitude transform*, hence the phase seems to matter much more than the magnitude does.

5.4 Convolution Theorem

The **Convolution Theorem** goes as follows:

> The Fourier transform of the convolution of two functions is the product of their Fourier transforms: $F.G = U(f**g)$.
>
> The Fourier transform of the product of two functions is the convolution of the Fourier transform: $F**G = U(f.g)$.

5.5 Sampling

The idea of **sampling** is to go from a continuous world to a discrete world, i.e. from a function to a vector. Samples are typically measured on a *regular grid.*

For example, we might want to be able to approximate integrals sensibly.

$$\text{Sample}_{2\text{D}}(f(x,\, y)) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(x,\, y)\delta(x-i,\, y-j) = f(x,\, y) \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x-i,\, y-j)$$

5.5.1 Fourier Transform of a Sampled Signal

The *Fourier transform of a sampled signal* is given by the following equalities:

$$\begin{aligned} F\left(\text{Sample}_{2\text{D}}(f(x,y))\right) &= F\left(f(x,y) \sum_{i=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} \delta(x-i,y-j)\right) \\ &= F\left(f(x,y)\right) ** F\left(\sum_{i=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} \delta(x-i,y-j)\right) \\ &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} F\left(u-i, v-j\right) \end{aligned}$$

5.5.2 Nyquist Sampling Theorem

The **Nyquist theorem** says that the sampling frequency must be at least twice the highest frequency, i.e. $\omega_s \geq 2\omega$. If this is not the case, the signal needs to be bandlimited before sampling, e.g. with a low-pass filter.

5.2 Image Restoration

5.2.1 Image Restoration Problem

If we have an image transformation of the form:

$$f(x) \to h(x) \to g(x) \to \tilde{h}(x) \to f(x)$$

Then, the **inverse kernel** $\tilde{h}(x)$ should compensate the effect of the *image degradation* $h(x)$, i.e.

$$(\tilde{h} * h)(x) = \delta(x)$$

$\tilde{h}$ may be determined more easily in the Fourier space, since:

$$\mathcal{F}[\tilde{h}](u,\, v) \cdot \mathcal{F}[h](u,\, v) = 1.$$

To determine $\mathcal{F}[\tilde{h}]$ we need to estimate:

1. The distortion model $h(x)$ or $\mathcal{F}[h](u, v)$
2. The parameters of $h(x)$, e.g. r for defocussing

5.2.2 Image Restoration: Motion Blur

The **kernel for motion blur** is given by: $h(x) = \frac{1}{2l}(\theta(x_1 + l) - \theta(x_1 - l))\delta(x_2)$, i.e. the transformation of a light dot into a small line in x_1 direction.

The Fourier transformation of this is given by:

$$\begin{aligned} \mathcal{F}[h](u, v) &= \frac{1}{2l} \int_{-l}^{+l} \exp(-\imath 2\pi u x_1) \underbrace{\int_{-\infty}^{+\infty} \delta(x_2) \exp(-\imath 2\pi v x_2) dx_2}_{=1} \, dx_1 \\ &= \frac{\sin(2\pi ul)}{2\pi ul} =: \mathrm{sinc}(2\pi ul) \end{aligned}$$

Which leads to:

- $\hat{h}(u) = \mathcal{F}[h](u) = \mathrm{sinc}(2\pi ul)$
- $\mathcal{F}[\tilde{h}](u) = \frac{1}{\hat{h}(u)}$

However, the following problems arise:

- The convolution with the kernel h completely cancels the frequencies $\frac{v}{2l}$ for $v \in \mathbb{Z}$. Vanishing frequencies cannot be recovered!
- There is a lot of noise amplification for $\mathcal{F}[h](u, v) << 1$.

5.2.3 Avoiding Noise Amplification

We can avoid **noise amplification** by a *regularized reconstruction filter* of the form:

$$\tilde{\mathcal{F}}[\tilde{h}](u, v) = \frac{\mathcal{F}[h]}{|\mathcal{F}[h]|^2 + \epsilon}$$

.

The size of ϵ implicitly determines an estimate of the noise level in the image, since we discard signal which are dampened below the size ϵ.