Visual Computing:

Fourier Transform

Prof. Marc Pollefeys
Last lecture

Fourier Transform

\[ F(g(x,y))(u,v) = \int\int_{\mathbb{R}^2} g(x,y) e^{-i2\pi(ux+vy)} \, dx \, dy \]

\[ e^{-i2\pi(ux+vy)} = \cos 2\pi(ux+vy) - i \sin 2\pi (ux+vy) \]
Various Fourier Transform Pairs

• Important facts
  – The Fourier transform is linear
  – There is an inverse FT \( f = U^{-1}F \)
  – scale function down \( \iff \) scale transform up
    i.e. high frequency = small details
  – The FT of a Gaussian is a Gaussian.

\[ f = U^{-1}F \]

compare to box function transform
Fourier Transform of important functions

Spatial domain

frequency domain

Sin wave

Impulse, or "delta" function

Comb

Boxcar

Sinc Function

Boxcar
Convolution theorem

- The convolution theorem
  - The Fourier transform of the convolution of two functions is the product of their Fourier transforms
    \[ F \cdot G = U(f \ast \ast g) \]  
    (cfr. filtering)
  - The Fourier transform of the product of two functions is the convolution of the Fourier transforms
    \[ F \ast \ast G = U(f \cdot g) \]  
    (cfr. sampling)
Sampling

• Go from continuous world to discrete world, from function to vector
• Samples are typically measured on regular grid

![Diagram of sampling process]
Sampling in 2D does the same thing, only in 2D. We’ll assume that these sample points are on a regular grid, and can place one at each integer point for convenience.
A continuous model for a sampled function

- We want to be able to approximate integrals sensibly
- Leads to
  - the delta function
  - model on right

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

\[ = f(x,y) \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x-i, y-j) \]
Delta function

- limit to infinity of constant area function:
A continuous model for a sampled function

• We want to be able to approximate integrals sensibly
• Leads to
  – the delta function
  – model on right

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

\[
= f(x,y) \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x-i, y-j)
\]
The Fourier transform of a sampled signal

\[
F(\text{Sample}_{2D}(f(x,y))) = F\left( f(x, y) \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x - i, y - j) \right)
\]

\[
= F(f(x, y)) \ast \ast F\left( \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x - i, y - j) \right)
\]

\[
= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} F(u - i, v - j)
\]
multiply

Sample

Sampled Signal

Fourier Transform

Magnitude Spectrum

multiply

Copy and Shift

convolve

Cut out by multiplication with box filter

Aliasing!
Proper sampling

- Original signal
- Low-pass filtered signal
- Sampling
- Reconstructed signal
Smoothing as low-pass filtering

• The message of the FT is that high frequencies lead to trouble with sampling.

• Solution: suppress high frequencies before sampling
  – multiply the FT of the signal with something that suppresses high frequencies
  – or convolve with a low-pass filter

• A filter whose FT is a box is bad, because the filter kernel has infinite support

• Common solution: use a Gaussian
  – multiplying FT by Gaussian is equivalent to convolving image with Gaussian.
Sampling without smoothing.
Top row shows the images, sampled at every second pixel to get the next; bottom row shows the magnitude spectrum of these images.
Sampling with smoothing. Top row shows the images. We get the next image by smoothing the image with a Gaussian with sigma 1 pixel, then sampling at every second pixel to get the next; bottom row shows the magnitude spectrum of these images.
Sampling with smoothing.
Top row shows the images. We get the next image by smoothing the image with a Gaussian with sigma 1.4 pixels, then sampling at every second pixel to get the next; bottom row shows the magnitude spectrum of these images.
Nyquist sampling theorem

- Nyquist theorem: The sampling frequency must be at least twice the highest frequency
  \[ \omega_s \geq 2\omega \]

- If this is not the case the signal needs to be bandlimited before sampling, e.g. with a low-pass filter
What went wrong?

color sampled at half-resolution!
Signal reconstruction
Image reconstruction: pixelization

• Who is this?
Reconstruction filters

Square pixels

Gaussian reconstruction filter

Bilinear interpolation

Perfect reconstruction filter
Convolution of Box with Box

https://commons.wikimedia.org/wiki/File:Convolution_of_box_signal_with_itself2.gif
Designing the ‘perfect’ low-pass filter
Filtering in Fourier domain
Defocus blurring
Motion blurring

Each light dot is transformed into a short line along the \( x_1 \)-axis:

\[
h(x_1, x_2) = \frac{1}{2l} \left[ \theta(x_1 + l) - \theta(x_1 - l) \right] \delta(x_2)
\]
The ‘inverse’ kernel $\tilde{h}(x)$ should compensate the effect of the image degradation $h(x)$, i.e.,

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

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

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

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

1. the distortion model $h(x)$ (point spread function) or $\mathcal{F}[h](u, v)$ (modulation transfer function)
2. the parameters of $h(x)$, e.g. $r$ for defocussing.
Image Restoration: Motion Blur

Kernel for motion blur

\[ h(x) = \frac{1}{2l} \left( \theta(x_1 + l) - \theta(x_1 - l) \right) \delta(x_2) \]

(a light dot is transformed into a small line in \( x_1 \) direction).

Fourier transformation:

\[
\mathcal{F}[h](u,v) = \frac{1}{2l} \int_{-l}^{+l} \exp(-i2\pi ux_1) \left( \int_{-\infty}^{+\infty} \delta(x_2) \exp(-i2\pi vx_2) dx_2 \right) dx_1
\]

\[
= \frac{\sin(2\pi ul)}{2\pi ul} =: \text{sinc}(2\pi ul)
\]
\[ \text{sinc}(u) = \frac{\sin(u)}{u} \]

\[ \hat{h}(u) = \mathcal{F}[h](u) = \text{sinc}(2\pi ul) \quad \mathcal{F}[\tilde{h}](u) = 1/\hat{h}(u) \]

**Problems:**

- Convolution with the kernel \( h \) completely cancels the frequencies \( \frac{\nu}{2l} \) for \( \nu \in \mathbb{Z} \). Vanishing frequencies cannot be recovered!
- Noise amplification for \( \mathcal{F}[h](u, v) \ll 1 \).
Avoiding noise amplification

**Regularized reconstruction filter:**

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

Singularities are avoided by the regularization \( \epsilon \).

The size of \( \epsilon \) implicitly determines an estimate of the noise level in the image, since we discard signals which are dampened below the size \( \epsilon \).
Coded Exposure Photography: Assisting Motion Deblurring using Fluttered Shutter

Raskar, Agrawal, Tumblin (Siggraph2006)

Short Exposure | Traditional | Coded

- **Image is dark and noisy**
- **Result has Banding Artifacts and some spatial frequencies are lost**
- **Decoded image is as good as image of a static scene**
Space-time super-resolution

Shechtman et al. PAMI05
Space-time super-resolution
Shechtman et al. PAMI05
Space-time super-resolution

Shechtman et al. PAMI05

time super-resolution works better than space
Spatial super-resolution

• lens+pixel=low-pass filter (desired to avoid aliasing)

• Low-res images = D*H*G*(desired high-res image)
  – D: decimate, H:lens+pixel, G: Geometric warp

• Simplified case for translation: LR=(D*G)*(H*HR)
  – G is shift-invariant and commutes with H
  – First compute H*HR, then deconvolve HR with H

• Super-resolution needs to restore attenuated frequencies
  – Many images improve S/N ratio (~sqrt(n)), which helps
  – Eventually Gaussian’s double exponential always dominates
Next week:
More Image Transformations

Eigenfaces

Wavelets
• Maybe a bit too short, explain better super-resolution with added noise, etc. Lena+gaussian noise slide is funny…
• Defocus bluring slide also a bit funny…
• Make slides to explain reconstruction kernels as convolution of boxes…