Image pyramids and frequency domain

Hi, Dr. Elizabeth?
Yeah, uh... I accidentally took the Fourier transform of my cat...

Meow!
Course announcements

- Homework 1 posted on course website.
  - Due on February 5th at 23:59.
  - This homework is in Matlab.

- Office hours for the rest of the semester:
  Monday, 4-6 pm, Anand
  Tuesday, 4-6 pm, Prakhar
  Friday, 4-6 pm, Yannis

- For now, all office hours will be at the Smith Hall graphics lounge, but a permanent location will be announced later this week.
Overview of today’s lecture

- Image downsampling.
- Aliasing.
- Gaussian image pyramid.
- Laplacian image pyramid.
- Fourier series.
- Frequency domain.
- Fourier transform.
- Frequency-domain filtering.
- Revisiting sampling.
Slide credits

Most of these slides were adapted directly from:

• Kris Kitani (15-463, Fall 2016).

Some slides were inspired or taken from:

• Fredo Durand (MIT).
• Bernd Girod (Stanford University).
• James Hays (Georgia Tech).
• Steve Marschner (Cornell University).
• Steve Seitz (University of Washington).
Image downsampling
This image is too big to fit on the screen. How would you reduce it to half its size?
Naïve image downsampling

Throw away half the rows and columns

delete even rows
delete even columns

What is the problem with this approach?
Naïve image downsampling

What is the 1/8 image so pixelated (and do you know what this effect is called)?
Aliasing
Reminder

Images are a *discrete*, or *sampled*, representation of a *continuous* world
Sampling

Very simple example: a sine wave

How would you discretize this signal?
Sampling

Very simple example: a sine wave
Sampling

Very simple example: a sine wave

How many samples should I take?
Can I take as many samples as I want?
Sampling

Very simple example: a sine wave

How many samples should I take?
Can I take as few samples as I want?
Undersampling

Very simple example: a sine wave

Unsurprising effect: information is lost.
Undersampling

Very simple example: a sine wave

Unsurprising effect: information is lost.
Surprising effect: can confuse the signal with one of lower frequency.
Undersampling

Very simple example: a sine wave

Unsurprising effect: information is lost.
Surprising effect: can confuse the signal with one of lower frequency.
Note: we could always confuse the signal with one of higher frequency.
Aliasing

Fancy term for: *Undersampling can disguise a signal as one of a lower frequency*

Unsurprising effect: information is lost.
Surprising effect: can confuse the signal with one of *lower* frequency.
Note: we could always confuse the signal with one of *higher* frequency.
Aliasing in textures
Aliasing in photographs

This is also known as “moire”
Temporal aliasing

Imagine a spoked wheel moving to the right (rotating clockwise). Mark wheel with dot so we can see what’s happening.

If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):

Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)
Anti-aliasing

How would you deal with aliasing?
Anti-aliasing

How would you deal with aliasing?

Approach 1: Oversample the signal
Anti-aliasing in textures

aliasing artifacts  anti-aliasing by oversampling
Anti-aliasing

How would you deal with aliasing?

Approach 1: Oversample the signal

Approach 2: Smooth the signal
  • Remove some of the detail effects that cause aliasing.
  • Lose information, but better than aliasing artifacts.

How would you smooth a signal?
Better image downsampling

Apply a smoothing filter first, then throw away half the rows and columns.

Gaussian filter delete even rows delete even columns

1/2

Gaussian filter delete even rows delete even columns

1/4

Gaussian filter delete even rows delete even columns

1/8
Better image downsampling

1/2

1/4 (2x zoom)

1/8 (4x zoom)
Naïve image downsampling

1/2  1/4 (2x zoom)  1/8 (4x zoom)
Anti-aliasing

Question 1: How much smoothing do I need to do to avoid aliasing?

Question 2: How many samples do I need to take to avoid aliasing?

Answer to both: Enough to reach the Nyquist limit.

We’ll see what this means soon.
Gaussian image pyramid
Gaussian image pyramid

The name of this sequence of subsampled images
Constructing a Gaussian pyramid

Algorithm

repeat:
    filter
    subsample
until min resolution reached

Question: How much bigger than the original image is the whole pyramid?
Constructing a Gaussian pyramid

Algorithm

repeat:
  filter
  subsample
until min resolution reached

Question: How much bigger than the original image is the whole pyramid?
Answer: Just 4/3 times the size of the original image! (How did I come up with this number?)
Some properties of the Gaussian pyramid

What happens to the details of the image?
Some properties of the Gaussian pyramid

What happens to the details of the image?
• They get smoothed out as we move to higher levels.

What is preserved at the higher levels?
Some properties of the Gaussian pyramid

What happens to the details of the image?
• They get smoothed out as we move to higher levels.

What is preserved at the higher levels?
• Mostly large uniform regions in the original image.

How would you reconstruct the original image from the image at the upper level?
Some properties of the Gaussian pyramid

What happens to the details of the image?
• They get smoothed out as we move to higher levels.

What is preserved at the higher levels?
• Mostly large uniform regions in the original image.

How would you reconstruct the original image from the image at the upper level?
• That’s not possible.
Blurring is lossy

level 0 - level 1 (before downsampling) = residual

What does the residual look like?
Blurring is lossy

level 0 - level 1 (before downsampling) = residual

Can we make a pyramid that is lossless?
Laplacian image pyramid
At each level, retain the residuals instead of the blurred images themselves.

Can we reconstruct the original image using the pyramid?
At each level, retain the residuals instead of the blurred images themselves.

Can we reconstruct the original image using the pyramid?
• Yes we can!

What do we need to store to be able to reconstruct the original image?
Let's start by looking at just one level

Does this mean we need to store both residuals and the blurred copies of the original?
Constructing a Laplacian pyramid

Algorithm
repeat:
  filter
  compute residual
  subsample
until min resolution reached
Constructing a Laplacian pyramid

Algorithm
repeat:
  filter
  compute residual
  subsample
until min resolution reached

What is this part?
Constructing a Laplacian pyramid

It’s a Gaussian pyramid.

Algorithm

repeat:
    filter
    compute residual
    subsample
until min resolution reached
What do we need to construct the original image?
What do we need to construct the original image?

(1) residuals
What do we need to construct the original image?

(2) smallest image

(1) residuals

$f_2$

$h_1$

$f_0$

$h_0$
Reconstructing the original image

Algorithm

repeat:
  upsample
  sum with residual
until orig resolution reached
Gaussian vs Laplacian Pyramid

Shown in opposite order for space.

Which one takes more space to store?
Why is it called a Laplacian pyramid?
Reminder: Laplacian of Gaussian (LoG) filter

As with derivative, we can combine Laplace filtering with Gaussian filtering.

**Input**

**Laplacian of Gaussian**

**Output**

“zero crossings” at edges
Why is it called a Laplacian pyramid?

Difference of Gaussians approximates the Laplacian.
Why Reagan?
Ronald Reagan was President when the Laplacian pyramid was invented.

The Laplacian Pyramid as a Compact Image Code (1983)

Peter J. Burt, Edward H. Adelson
Still used extensively
Still used extensively

foreground details enhanced, background details reduced

user-provided mask
Other types of pyramids

Steerable pyramid: At each level keep multiple versions, one for each direction.

Wavelets: Huge area in image processing (see 18-793).
What are image pyramids used for?

- Image compression
- Multi-scale texture mapping
- Image blending
- Focal stack compositing
- Denoising
- Multi-scale detection
- Multi-scale registration
Some history
Who is this guy?
What is he famous for?

Jean Baptiste Joseph Fourier
(1768-1830)
What is he famous for?

Jean Baptiste Joseph Fourier (1768-1830)

The Fourier series claim (1807):

‘Any univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.’

... and apparently also for the discovery of the greenhouse effect
Is this claim true?

The Fourier series claim (1807):
‘Any univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.’

Jean Baptiste Joseph Fourier (1768-1830)
Is this claim true?

The Fourier series claim (1807):
‘Any univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.’

Well, almost.
• The theorem requires additional conditions.
• Close enough to be named after him.
• Very surprising result at the time.

Jean Baptiste Joseph Fourier
(1768-1830)
Is this claim true?

The Fourier series claim (1807):
‘Any univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.’

Well, almost.
• The theorem requires additional conditions.
• Close enough to be named after him.
• Very surprising result at the time.

Jean Baptiste Joseph Fourier (1768-1830)

The committee examining his paper had expressed skepticism, in part due to not so rigorous proofs.
Amusing aside

Only known portrait of Adrien-Marie Legendre

1820 watercolor caricatures of French mathematicians Adrien-Marie Legendre (left) and Joseph Fourier (right) by French artist Julien-Leopold Boilly.

For two hundred years, people were misidentifying this portrait as him

Louis Legendre
(same last name, different person)
Fourier series
Basic building block

\[ A \sin(\omega x + \phi) \]

Fourier’s claim: Add enough of these to get any periodic signal you want!
Fourier’s claim: Add enough of these to get any periodic signal you want!
Examples

How would you generate this function?

= ? + ?
How would you generate this function?

\[ \text{[Image]} = \sin(2\pi x) + ? \]
Examples

How would you generate this function?

\[ f(x) = \sin(2\pi x) + \frac{1}{3} \sin(2\pi 3x) \]
Examples

How would you generate this function?

\[
\text{square wave} = \text{?} + \text{?}
\]
Examples

How would you generate this function?

\[
\text{square wave} \approx \sin(2\pi f t) + \sin(4\pi f t) + \ldots
\]
How would you generate this function?

square wave \approx \sin(2\pi f t) + \sin(4\pi f t)

Examples
Examples

How would you generate this function?

\[ \approx + \]

\[ = \]
Examples

How would you generate this function?

\[ \text{square wave} \approx \text{ } + \text{ } = \]
Examples

How would you generate this function?

\[
\text{square wave} \approx + \]

How would you express this mathematically?
Examples

\[ \text{square wave} = A \sum_{k=1}^{\infty} \frac{1}{k} \sin(2\pi k x) \]

How would you visualize this in the frequency domain?
Examples

square wave = \[ A \sum_{k=1}^{\infty} \frac{1}{k} \sin(2\pi k x) \]

infinite sum of sine waves

magnitude

frequency
Frequency domain
Visualizing the frequency spectrum
Visualizing the frequency spectrum

Recall the temporal domain visualization

\[ f(x) = \sin(2\pi k x) + \frac{1}{3} \sin(2\pi 3k x) \]
Visualizing the frequency spectrum

Recall the temporal domain visualization

\[ f(x) = \sin(2\pi kx) + \frac{1}{3} \sin(2\pi 3kx) \]

How do we plot ...

\[ \sin(2\pi kx) \]
Visualizing the frequency spectrum

Recall the temporal domain visualization

\[ f(x) = \sin(2\pi kx) + \frac{1}{3} \sin(2\pi 3kx) \]
Visualizing the frequency spectrum

Recall the temporal domain visualization

\[ f(x) = \sin(2\pi k x) + \frac{1}{3} \sin(2\pi 3k x) \]
Visualizing the frequency spectrum

not visualizing the symmetric negative part

What is at zero frequency?

Recall the temporal domain visualization

\[ f(x) = \sin(2\pi k x) + \frac{1}{3} \sin(2\pi 3k x) \]

Need to understand this to understand the 2D version!
Visualizing the frequency spectrum

Recall the temporal domain visualization

\[ f(x) = \sin(2\pi kx) + \frac{1}{3} \sin(2\pi 3kx) \]

Signal average (zero for a sine wave with no offset)

Not visualizing the symmetric negative part

Need to understand this to understand the 2D version!
Examples

Spatial domain visualization

Frequency domain visualization

1D

2D

$|F(k)|$
Examples

Spatial domain visualization

1D

Frequency domain visualization

$|F(k)|$

$k$

2D

$k_x$

$k_y$

What do the three dots correspond to?
Examples

Spatial domain visualization          Frequency domain visualization

$\Delta_{y}$ $\Delta_{x}$
Examples

Spatial domain visualization

Frequency domain visualization

$k_x$

$k_y$
Examples

How would you generate this image with sine waves?
Examples

How would you generate this image with sine waves?

Has both an x and y components
Examples

\[ \text{[Image 1]} + \text{[Image 2]} = ? \]
Examples

\[ \begin{array}{c}
\text{[Image]}
\end{array} \quad + \quad \begin{array}{c}
\text{[Image]}
\end{array} \quad = \quad \begin{array}{c}
\text{[Image]}
\end{array} \]
Fourier’s claim: Add enough of these to get *any* periodic signal you want!

Basic building block

What about non-periodic signals?
Fourier transform
Recalling some basics

Complex numbers have two parts:

rectangular coordinates

\[ R + jI \]

what’s this?  what’s this?
Recalling some basics

Complex numbers have two parts:

\[ R + jI \]

rectangular coordinates

real \quad imaginary
Recalling some basics

Complex numbers have two parts:

**rectangular coordinates**

\[ R + jI \]

real \hspace{1cm} imaginary

Alternative reparameterization:

**polar coordinates**

\[ r(\cos \theta + j \sin \theta) \]

how do we compute these?

polar transform
Recalling some basics

Complex numbers have two parts:

**rectangular coordinates**

\[ R + jI \]

- real
- imaginary

Alternative reparameterization:

**polar coordinates**

\[ r(\cos \theta + j \sin \theta) \]

- polar transform

\[ \theta = \tan^{-1}(\frac{I}{R}) \]
\[ r = \sqrt{R^2 + I^2} \]
Recalling some basics

Complex numbers have two parts:

**Rectangular coordinates**

\[ R + jI \]

real \hspace{0.5cm} \text{imaginary}

Alternative reparameterization:

**Polar coordinates**

\[ r(\cos \theta + j \sin \theta) \]

polar transform

\[ \theta = \tan^{-1}\left(\frac{I}{R}\right) \quad r = \sqrt{R^2 + I^2} \]

How do you write these in exponential form?
Recalling some basics

Complex numbers have two parts:

- **Rectangular coordinates**: $R + jI$
  - real
  - imaginary

Alternative reparameterization:

- **Polar coordinates**: $r(\cos \theta + j \sin \theta)$
  - polar transform
  - $\theta = \tan^{-1}\left(\frac{I}{R}\right)$
  - $r = \sqrt{R^2 + I^2}$

- **Exponential form**: $re^{j\theta}$
  - how did we get this?
Recalling some basics

Complex numbers have two parts:

rectangular coordinates

\[ R + jI \]

real \quad imaginary

Alternative reparameterization:

polar coordinates

\[ r (\cos \theta + j \sin \theta) \]

polar transform

\[ \theta = \tan^{-1}\left(\frac{I}{R}\right) \quad r = \sqrt{R^2 + I^2} \]

or equivalently

\[ re^{j\theta} \]

Euler’s formula

\[ e^{j\theta} = \cos \theta + j \sin \theta \]

This will help us understand the Fourier transform equations.
Fourier transform

<table>
<thead>
<tr>
<th>Continuous Fourier transform</th>
<th>Inverse Fourier transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ F(k) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi k x} , dx ]</td>
<td>[ f(x) = \int_{-\infty}^{\infty} F(k)e^{j2\pi k x} , dk ]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Discrete Fourier transform</th>
<th>Inverse Fourier transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ F(k) = \frac{1}{N} \sum_{x=0}^{N-1} f(x)e^{-j2\pi k x/N} ] for ( k = 0,1,2,\ldots,N-1 )</td>
<td>[ f(x) = \sum_{k=0}^{N-1} F(k)e^{j2\pi k x/N} ] for ( x = 0,1,2,\ldots,N-1 )</td>
</tr>
</tbody>
</table>

Where is the connection to the ‘summation of sine waves’ idea?
Fourier transform

**Continuous Fourier transform**

\[ F(k) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi k x} \, dx \]

**Inverse Fourier transform**

\[ f(x) = \int_{-\infty}^{\infty} F(k) e^{j2\pi k x} \, dk \]

**Discrete Fourier transform**

\[ F(k) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j2\pi k x / N} \]

\[ f(x) = \sum_{k=0}^{N-1} F(k) e^{j2\pi k x / N} \]

Where is the connection to the ‘summation of sine waves’ idea?
Fourier transform

Where is the connection to the ‘summation of sine waves’ idea?

\[ f(x) = \sum_{k=0}^{N-1} F(k) e^{i2\pi kx/N} \]

- Euler’s formula
  \[ e^{i\theta} = \cos \theta + j \sin \theta \]
- sum over frequencies
- scaling parameter
- wave components
Fourier transform pairs

spatial domain

frequency domain

Note the symmetry: duality property of Fourier transform
Computing the discrete Fourier transform (DFT)
Computing the discrete Fourier transform (DFT) is just a matrix multiplication:

\[ F(k) = \frac{1}{N} \sum_{x=0}^{N-1} f(x)e^{-j2\pi kx/N} \]

is just a matrix multiplication:

\[ F = Wf \]

\[
\begin{bmatrix}
F(0) \\
F(1) \\
F(2) \\
F(3) \\
\vdots \\
F(N-1)
\end{bmatrix} =
\begin{bmatrix}
W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\
W^0 & W^1 & W^2 & W^3 & \cdots & W^{N-1} \\
W^0 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\
W^0 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\
\vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\
W^0 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W^1
\end{bmatrix}
\begin{bmatrix}
f(0) \\
f(1) \\
f(2) \\
f(3) \\
\vdots \\
f(N-1)
\end{bmatrix}
\]

\[ W = e^{-j2\pi/N} \]

In practice this is implemented using the fast Fourier transform (FFT) algorithm.
Fourier transforms of natural images

original

amplitude

phase
Fourier transforms of natural images

Image phase matters!

cheetah phase with zebra amplitude

zebra phase with cheetah amplitude
Frequency-domain filtering
Why do we care about all this?
The convolution theorem

The Fourier transform of the convolution of two functions is the product of their Fourier transforms:

$$\mathcal{F}\{g \ast h\} = \mathcal{F}\{g\} \mathcal{F}\{h\}$$

The inverse Fourier transform of the product of two Fourier transforms is the convolution of the two inverse Fourier transforms:

$$\mathcal{F}^{-1}\{gh\} = \mathcal{F}^{-1}\{g\} \ast \mathcal{F}^{-1}\{h\}$$

Convolution in spatial domain is equivalent to multiplication in frequency domain!
What do we use convolution for?
Convolution for 1D continuous signals

Definition of linear shift-invariant filtering as convolution:

\[(f * g)(x) = \int_{-\infty}^{\infty} f(y)g(x - y)dy\]

filtered signal \[\rightarrow\] filter \[\rightarrow\] input signal

Using the convolution theorem, we can interpret and implement all types of linear shift-invariant filtering as multiplication in frequency domain.

Why implement convolution in frequency domain?
Frequency-domain filtering in Matlab

Filtering with `fft`:

```matlab
im = double(imread('...'))/255;
im = rgb2gray(im); % "im" should be a gray-scale floating point image
[imh, imw] = size(im);

hs = 50; % filter half-size
fil = fspecial('gaussian', hs*2+1, 10);

fftsize = 1024; % should be order of 2 (for speed) and include padding
im_fft = fft2(im, fftsize, fftsize); % 1) fft im with padding
fil_fft = fft2(fil, fftsize, fftsize); % 2) fft fil, pad to same size as image
im_fil_fft = im_fft .* fil_fft; % 3) multiply fft images
im_fil = ifft2(im_fil_fft); % 4) inverse fft2
im_fil = im_fil(1+hs:size(im,1)+hs, 1+hs:size(im, 2)+hs); % 5) remove padding
```

Displaying with `fft`:

```matlab
figure(1), imagesc(log(abs(fftshift(im_fft)))), axis image, colormap jet
```
Spatial domain filtering

Fourier transform

Frequency domain filtering

Fourier transform

inverse Fourier transform
Revisiting blurring

Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?
Gaussian blur
Box blur
More filtering examples

filters shown in frequency-domain
More filtering examples

filters shown in frequency-domain

low-pass

band-pass
More filtering examples
More filtering examples
More filtering examples

original image

frequency magnitude

low-pass filter

?
More filtering examples

original image

[Image]

low-pass filter

[Image]

frequency magnitude

[Image]
More filtering examples

original image

high-pass filter

frequency magnitude
More filtering examples

original image

frequency magnitude

high-pass filter
More filtering examples

original image

band-pass filter

frequency magnitude
More filtering examples

original image

frequency magnitude

band-pass filter
More filtering examples

original image

frequency magnitude

band-pass filter
More filtering examples

original image

[Image of a face]

frequency magnitude

[Image of a grid]

band-pass filter

[Image of a circular pattern]
Revisiting sampling
The Nyquist-Shannon sampling theorem

A continuous signal can be perfectly reconstructed from its discrete version using linear interpolation, if sampling occurred with frequency:

\[ f_s \geq 2f_{\text{max}} \]

This is called the Nyquist frequency.

Equivalent reformulation: When downsampling, aliasing does not occur if samples are taken at the Nyquist frequency or higher.
How does the Nyquist-Shannon theorem relate to the Gaussian pyramid?
Gaussian pyramid

How does the Nyquist-Shannon theorem relate to the Gaussian pyramid?

- Gaussian blurring is low-pass filtering.
- By blurring we try to sufficiently decrease the Nyquist frequency to avoid aliasing.

How large should the Gauss blur we use be?
Frequency-domain filtering in human vision

“Hybrid image”

Aude Oliva and Philippe Schyns
Frequency-domain filtering in human vision

Gala Contemplating the Mediterranean Sea Which at Twenty Meters Becomes the Portrait of Abraham Lincoln (Homage to Rothko)

Salvador Dali, 1976
Frequency-domain filtering in human vision

Low-pass filtered version
Frequency-domain filtering in human vision
Variable frequency sensitivity

Experiment: Where do you see the stripes?
Our eyes are sensitive to mid-range frequencies.

- Early processing in humans filters for various orientations and scales of frequency.
- Perceptual cues in the mid frequencies dominate perception.
Basic reading:
• Szeliski textbook, Sections 3.4, 3.5

Additional reading:
• Hubel and Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat's visual cortex,” The Journal of Physiology 1962
  A foundational paper describing information processing in the visual system, including the different types of filtering it performs; Hubel and Wiesel won the Nobel Prize in Medicine in 1981 for the discoveries described in this paper.