CS 4495 Computer Vision

Frequency and Fourier Transforms

Aaron Bobick
School of Interactive Computing
Administrivia

• Project 1 is (still) on line – you should really get started now!

• Readings for this week: FP Chapter 4 (which includes reviewing 4.1 and 4.2)
Questions about PS1?

• Where should I put the origin?
  • It’s up to you – you get to define the geometry.

• Should $\theta$ go from $-\pi$ to $\pi$ or $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ or what?
  • It’s up to you – you get to define the geometry.

• How do I draw the line?
  • I’m guessing that any line in your image crosses approximately two edges in the image. So given an equation of the line, you could try $x=1$ or $x=256$ or $y=1$ or $y=256$ and see what values you get. Just a thought…
Salvador Dali

“Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln”, 1976
Decomposing an image

- A basis set is (edit from to Wikipedia):
  - A basis \( B \) of a vector space \( V \) is a linearly independent subset of \( V \) that spans \( V \).
  - In more detail: suppose that \( B = \{ v_1, \ldots, v_n \} \) is a finite subset of a vector space \( V \) over a field \( F \) (such as the real or complex numbers \( \mathbb{R} \) or \( \mathbb{C} \)). Then \( B \) is a basis if it satisfies the following conditions:
    - the linear independence property:
      - for all \( a_1, \ldots, a_n \in F \), if \( a_1v_1 + \cdots + a_nv_n = 0 \), then necessarily \( a_1 = \cdots = a_n = 0 \);
    - and the spanning property,
      - for every \( x \) in \( V \) it is possible to choose \( a_1, \ldots, a_n \in F \) such that \( x = a_1v_1 + \cdots + a_nv_n \).
  - Not necessarily orthogonal

- If we have a basis set for images, could perhaps be useful for analysis – especially for linear systems because we could consider each basis component independently. (Why?)
Images as points in a vector space

- Consider an image as a point in a NxN size space – can rasterize into a single vector
  \[
  \begin{bmatrix}
  x_{00} & x_{10} & x_{20} & \ldots & x_{(n-1)0} & x_{11} & \ldots & x_{(n-1)(n-1)}
  \end{bmatrix}^T
  \]

- The “normal” basis is just the vectors:
  \[
  \begin{bmatrix}
  0 & 0 & 0 & 0 & \ldots & 0 & 1 & 0 & 0 & 0 & \ldots & 0
  \end{bmatrix}^T
  \]
  - Independent
  - Can create any image

- But not very helpful to consider how each pixel contributes to computations.
A nice set of basis
Teases away fast vs. slow changes in the image.

This change of basis has a special name…
Jean Baptiste Joseph Fourier (1768-1830)

- Had crazy idea (1807):
  - *Any* periodic function can be rewritten as a weighted sum of sines and cosines of different frequencies.

- Don’t believe it?
  - Neither did Lagrange, Laplace, Poisson and other big wigs
  - Not translated into English until 1878!

- But it’s true!
  - Called Fourier *Series*
A sum of sines

• Our building block:
  \[ A \sin(\omega x + \phi) \]

• Add enough of them to get any signal \( f(x) \) you want!

• How many degrees of freedom?

• What does each control?

• Which one encodes the coarse vs. fine structure of the signal?

\[
f(\text{target}) = \sum_{n=1}^{\infty} f_n + f_{n+1} + f_{n+2} + \ldots + f_{n+m} + \ldots
\]
Time and Frequency

- example: $g(t) = \sin(2\pi f t) + (1/3)\sin(2\pi (3f) t)$
Time and Frequency

- **example**: \( g(t) = \sin(2\pi f t) + \frac{1}{3}\sin(2\pi (3f) t) \)
Frequency Spectra - Series

- example: \( g(t) = \sin(2pf \ t) + \left(\frac{1}{3}\right)\sin(2p(3f) \ t) \)

One form of spectrum – more in a bit
Frequency Spectra - Series
Frequency Spectra - Series

\[ \approx \]

\[ + \]

\[ = \]
Frequency Spectra - Series
Frequency Spectra - Series
Frequency Spectra - Series
Frequency Spectra - Series

$$F(t) = A \sum_{k=1}^{\infty} \frac{1}{k} \sin(2\pi kt)$$

Usually, frequency is more interesting than the phase for CV because we’re not reconstructing the image.
Fourier Transform

We want to understand the frequency $\omega$ of our signal. So, let’s reparametrize the signal by $\omega$ instead of $x$:

$$f(x) \rightarrow \text{Fourier Transform} \rightarrow A\sin(\omega x + \phi) \rightarrow F(\omega)$$

For every $\omega$ from 0 to inf (actually $-\infty$ to $\infty$), $F(\omega)$ holds the amplitude $A$ and phase $\phi$ of the corresponding sine

- How can $F$ hold both? Complex number trick!

Recall: $e^{ik} = \cos k + i\sin k \quad i = \sqrt{-1}$ (or $j$)

Even    Odd

Matlab sinusoid demo...
Fourier Transform

We want to understand the frequency $\omega$ of our signal. So, let’s reparametrize the signal by $\omega$ instead of $x$:

$$f(x) \xrightarrow{\text{Fourier Transform}} A \sin(\omega x + \phi) \xrightarrow{\text{Inverse Fourier Transform}} F(\omega)$$

For every $\omega$ from 0 to inf, (actually $-\infty$ to $\infty$), $F(\omega)$ holds the amplitude $A$ and phase $\phi$ of the corresponding sine.

- How can $F$ hold both? Complex number trick!

$$A = \pm \sqrt{R(\omega)^2 + I(\omega)^2}$$

$$F(\omega) = R(\omega) + i I(\omega)$$

Even $\quad$ Odd

$$\phi = \tan^{-1} \frac{I(\omega)}{R(\omega)}$$

And we can go back:

$$F(\omega) \xrightarrow{\text{Inverse Fourier Transform}} f(x)$$
Computing FT: Just a basis

• The infinite integral of the product of two sinusoids of different frequency is zero. (Why?)

\[ \int_{-\infty}^{\infty} \sin(ax + \phi) \sin(bx + \varphi) \, dx = 0, \text{ if } a \neq b \]

• And the integral is infinite if equal (unless exactly out of phase):

\[ \int_{-\infty}^{\infty} \sin(ax + \phi) \sin(ax + \varphi) \, dx = \pm \infty \]

If \( \phi \) and \( \varphi \) not exactly \( \pi/2 \) out of phase (sin and cos).
Computing FT: Just a basis

- So, suppose $f(x)$ is a cosine wave of freq $\omega$:

$$f(x) = \cos(2\pi \omega x)$$

- Then:

$$C(u) = \int_{-\infty}^{\infty} f(x) \cos(2\pi u x) dx$$

Is infinite if $u$ is equal to $\omega$ (or $-\omega$) and zero otherwise:
Computing FT: Just a basis

- We can do that for all frequencies $u$.

- But we’d have to do that for all phases, don’t we???

- No! Any phase can be created by a weighted sum of cosine and sine. Only need each piece:

  $$C(u) = \int_{-\infty}^{\infty} f(x) \cos(2\pi u x) \, dx$$
  $$S(u) = \int_{-\infty}^{\infty} f(x) \sin(2\pi u x) \, dx$$

- Sinusoid demo?
- Or…
Fourier Transform – more formally

Represent the signal as an infinite weighted sum of an infinite number of sinusoids

\[ F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} \, dx \]

Again: \( e^{ik} = \cos k + i \sin k \quad i = \sqrt{-1} \)

Spatial Domain (x) \quad \rightarrow \quad Frequency Domain (u or s)

(Frequency Spectrum \( F(u) \))

Inverse Fourier Transform (IFT) – add up all the sinusoids at x:

\[ f(x) = \int_{-\infty}^{\infty} F(u) e^{i2\pi ux} \, du \]
Fourier Transform - limitations

- The integral \( \int_{-\infty}^{\infty} f(x)e^{-i2\pi ux} \, dx \) exists if the function \( f \) is integrable:

\[
\int_{-\infty}^{\infty} \left| f(x) \right| \, dx < \infty
\]

- If there is a bound of width \( T \) outside of which \( f \) is zero then obviously could integrate from just \(-T/2\) to \( T/2\)
Fourier Transform $\iff$ Fourier Series

- The bounded integral give some relation between the Fourier transform and the series and the Discrete Fourier transform.
- The **Discrete FT**:

$$F(k) = \frac{1}{N} \sum_{x=0}^{x=N-1} f(x)e^{-i\frac{2\pi kx}{N}}$$

- $k$ is the number “cycles per period of the signal” or “cycles per image.
- Only makes sense $k = -N/2$ to $N/2$. Why? What’s the highest frequency you can unambiguously have in a discrete image?
- What is $F(k)$ when $k$ is zero?
2D Fourier Transforms

The two dimensional version:

\[ F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi(ux + vy)} \, dx \, dy \]

And the 2D Discrete FT:

\[ F(k_x, k_y) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) e^{-i \frac{2\pi}{N} (k_x x + k_y y)} \]

Works best when you put the origin of \( k \) in the middle....
Frequency Spectra – Even/Odd

Frequency actually goes from –inf to inf. Sinusoid example:

Even \((\cos)\)

Odd \((\sin)\)

Magnitude

Real

Imaginary

Power
Frequency Spectra

(a) 

(b) 

(c)
Extension to 2D
2D Examples – sinusoid magnitudes
2D Examples – sinusoid magnitudes
2D Examples – sinusoid magnitudes
Linearity of Sum
Extension to 2D – Complex plane

Both a Real and Im version
Examples
Man-made Scene

Where is this strong horizontal suggested by vertical center line?
Fourier Transform and Convolution

Let $g = f \ast h$

Then $G(u) = \int_{-\infty}^{\infty} g(x) e^{-i2\pi ux} \, dx$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\tau) h(x - \tau) e^{-i2\pi ux} \, d\tau \, dx$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ f(\tau) e^{-i2\pi u\tau} \, d\tau \right] h(x - \tau) e^{-i2\pi u(x - \tau)} \, dx$$

$$= \int_{-\infty}^{\infty} \left[ f(\tau) e^{-i2\pi u\tau} \, d\tau \right] \int_{-\infty}^{\infty} h(x') e^{-i2\pi u x'} \, dx'$$

$$= F(u)H(u)$$

Convolution in spatial domain

$\Leftrightarrow$ Multiplication in frequency domain
Fourier Transform and Convolution

Spatial Domain \((x)\) \hspace{1cm} Frequency Domain \((u)\)

\[
g = f * h \quad \leftrightarrow \quad G = FH
\]

\[
g = fh \quad \leftrightarrow \quad G = F * H
\]

So, we can find \(g(x)\) by Fourier transform

\[
g \quad = \quad f \quad * \quad h
\]

\[
G \quad = \quad F \quad \times \quad H
\]
Example use: Smoothing/Blurring

- We want a smoothed function of $f(x)$

\[ g(x) = f(x) \ast h(x) \]

- Let us use a Gaussian kernel

\[ h(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[ -\frac{1}{2} \frac{x^2}{\sigma^2} \right] \]

- The Fourier transform of a Gaussian is a Gaussian

\[ H(u) = \exp\left[ -\frac{1}{2} (2\pi u)^2 \sigma^2 \right] \]

Fat Gaussian in space is skinny Gaussian in frequency. Why?
Example use: Smoothing/Blurring

- We want a smoothed function of $f(x)$

$$g(x) = f(x) * h(x)$$

- Let us use a Gaussian kernel

$$h(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{1}{2} \frac{x^2}{\sigma^2}\right]$$

- Convolution in space is multiplication in freq:

$$G(u) = F(u)H(u)$$

$H(u)$ attenuates high frequencies in $F(u)$ (Low-pass Filter)!
2D convolution theorem example

\[ f(x,y) \]
\[ h(x,y) \]
\[ g(x,y) \]

\[ \text{\( |F(s_x, s_y)| \)} \]
\[ \text{(or \( |F(u,v)| \))} \]
\[ \text{\( |H(s_x, s_y)| \)} \]
\[ \text{\( |G(s_x, s_y)| \)} \]
Low and High Pass filtering

Ringing
Properties of Fourier Transform

<table>
<thead>
<tr>
<th>Spatial Domain (x)</th>
<th>Frequency Domain (u)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linearity</strong></td>
<td></td>
</tr>
<tr>
<td>$c_1f(x) + c_2g(x)$</td>
<td>$c_1F(u) + c_2G(u)$</td>
</tr>
<tr>
<td><strong>Scaling</strong></td>
<td></td>
</tr>
<tr>
<td>$f(ax)$</td>
<td>$\frac{1}{</td>
</tr>
<tr>
<td><strong>Shifting</strong></td>
<td></td>
</tr>
<tr>
<td>$f(x - x_0)$</td>
<td>$e^{-i2\pi ux_0}F(u)$</td>
</tr>
<tr>
<td><strong>Symmetry</strong></td>
<td></td>
</tr>
<tr>
<td>$F(x)$</td>
<td>$f(-u)$</td>
</tr>
<tr>
<td><strong>Conjugation</strong></td>
<td></td>
</tr>
<tr>
<td>$f^*(x)$</td>
<td>$F^*(-u)$</td>
</tr>
<tr>
<td><strong>Convolution</strong></td>
<td></td>
</tr>
<tr>
<td>$f(x) * g(x)$</td>
<td>$F(u)G(u)$</td>
</tr>
<tr>
<td><strong>Differentiation</strong></td>
<td></td>
</tr>
<tr>
<td>$\frac{d^n f(x)}{dx^n}$</td>
<td>$(i2\pi u)^n F(u)$</td>
</tr>
</tbody>
</table>
## Fourier Pairs (from Szeliski)

<table>
<thead>
<tr>
<th>Name</th>
<th>Signal</th>
<th>Transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>impulse</td>
<td>$\delta(x)$</td>
<td>$1$</td>
</tr>
<tr>
<td>shifted impulse</td>
<td>$\delta(x-u)$</td>
<td>$e^{-j\omega u}$</td>
</tr>
<tr>
<td>box filter</td>
<td>$\text{box}(x/a)$</td>
<td>$a \text{sinc}(a\omega)$</td>
</tr>
<tr>
<td>tent</td>
<td>$\text{tent}(x/a)$</td>
<td>$a \text{sinc}^2(a\omega)$</td>
</tr>
<tr>
<td>Gaussian</td>
<td>$G(x; \sigma)$</td>
<td>$\sqrt{\frac{2\pi}{\sigma}}G(\omega; \sigma^{-1})$</td>
</tr>
<tr>
<td>Laplacian of Gaussian</td>
<td>$(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2})G(x; \sigma)$</td>
<td>$-\sqrt{\frac{2\pi}{\sigma}}\omega^2G(\omega; \sigma^{-1})$</td>
</tr>
<tr>
<td>Gabor</td>
<td>$\cos(\omega_0 x)G(x; \sigma)$</td>
<td>$\sqrt{\frac{2\pi}{\sigma}}G(\omega \pm \omega_0; \sigma^{-1})$</td>
</tr>
<tr>
<td>unsharp mask</td>
<td>$(1+\gamma)\delta(x)$</td>
<td>$\frac{(1+\gamma)}{\sqrt{2\pi\sigma}}G(\omega; \sigma^{-1})$</td>
</tr>
<tr>
<td>windowed sinc</td>
<td>$\text{rcos}(x/(aW))$</td>
<td>(see Figure 3.29)</td>
</tr>
</tbody>
</table>
Fourier Transform smoothing pairs

\[ f(x) \]

**Spatial domain**

- box\( (x) \)
- gauss\( (x; \sigma) \)
- sinc\( (s) \)

**Frequency domain**

- \( F(s) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi sx} \, dx \)
- gauss\( (s; 1/\sigma) \)
- sinc\( (s) \)
- box\( (x) \)
Fourier Transform Sampling Pairs

FT of an “impulse train” is an impulse train

FT of an “impulse train” is an impulse train
Sampling and Aliasing
Sampling and Reconstruction
Sampled representations

- How to store and compute with continuous functions?
- Common scheme for representation: samples
  - write down the function’s values at many points
Reconstruction

- Making samples back into a continuous function
  - for output (need realizable method)
  - for analysis or processing (need mathematical method)
  - amounts to “guessing” what the function did in between
1D Example: Audio
Sampling in digital audio

- Recording: sound to analog to samples to disc
- Playback: disc to samples to analog to sound again
  - how can we be sure we are filling in the gaps correctly?
Sampling and Reconstruction

- Simple example: a sign wave
Undersampling

- What if we “missed” things between the samples?
- Simple example: undersampling a sine wave
  - unsurprising result: information is lost
Undersampling

• What if we “missed” things between the samples?
• Simple example: undersampling a sine wave
  • unsurprising result: information is lost
  • surprising result: indistinguishable from lower frequency
Undersampling

- What if we “missed” things between the samples?
- Simple example: undersampling a sine wave
  - unsurprising result: information is lost
  - surprising result: indistinguishable from lower frequency
  - also was always indistinguishable from higher frequencies
  - **aliasing**: signals “traveling in disguise” as other frequencies
Aliasing in video

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)
Aliasing in images
What’s happening?

Input signal:

Plot as image:

```
x = 0:.05:5; imagesc(sin((2.^x).*x))
```

Alias!
Not enough samples
Antialiasing

• What can we do about aliasing?

• Sample more often
  • Join the Mega-Pixel craze of the photo industry
  • But this can’t go on forever

• Make the signal less “wiggly”
  • Get rid of some high frequencies
  • Will loose information
  • But it’s better than aliasing
Preventing aliasing

• Introduce lowpass filters:
  • remove high frequencies leaving only safe, low frequencies
  • choose lowest frequency in reconstruction (disambiguate)
(Anti)Aliasing in the Frequency Domain
Define a *comb* function (impulse train) in 1D as follows:

\[
comb_M[x] = \sum_{k=-\infty}^{\infty} \delta[x - kM]
\]

where \( M \) is an integer.
Impulse Train in 1D

\[ \text{comb}_2(x) \]

\[ \frac{1}{2} \text{comb}_{\frac{1}{2}}(u) \]

Remember:

**Scaling**

\[ f(ax) \]

\[ \frac{1}{|a|} F\left(\frac{u}{a}\right) \]
Impulse Train in 2D \textit{(bed of nails)}

\begin{equation*}
\text{comb}_{M,N}(x, y) \equiv \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \delta(x - kM, y - lN)
\end{equation*}

- Fourier Transform of an impulse train is also an impulse train:

\begin{equation*}
\sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \delta(x - kM, y - lN) \iff \frac{1}{MN} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \delta\left(u - \frac{k}{M}, v - \frac{l}{N}\right)
\end{equation*}

\begin{align*}
\text{comb}_{M,N}(x, y) & \qquad \text{comb}_{\frac{1}{M}, \frac{1}{N}}(u, v)
\end{align*}

\textit{As the comb samples get further apart, the spectrum samples get closer together!}

B.K. Gunturk
Impulse Train

\[ \text{comb}_2[n] \]

\[ \frac{1}{2} \text{comb}_\frac{1}{2}(u) \]

Remember:

Scaling \[ f(ax) \]

\[ \frac{1}{|a|} F\left(\frac{u}{a}\right) \]
Sampling low frequency signal

\[ f(x) \quad \leftrightarrow \quad F(u) \]

\[ f(x) \cdot \text{comb}_M(x) \quad \leftrightarrow \quad F(u) \cdot \text{comb}_M(u) \]

Multiply:

Convolve:
Sampling low frequency signal

\[ f(x) \quad \leftrightarrow \quad F(u) \]

\[ f(x) \text{comb}_M(x) \quad \leftrightarrow \quad F(u) \ast \text{comb}_{\frac{1}{M}}(u) \]

No “problem” if \( \frac{1}{M} > 2W \)

B.K. Gunturk
Sampling low frequency signal

\[ f(x) \cdot \text{comb}_M(x) \]

\[ F(u) \cdot \text{comb}_{\frac{1}{M}}(u) \]

If there is no overlap, the original signal can be recovered from its samples by low-pass filtering.
Sampling high frequency signal

Overlap: The high frequency energy is folded over into low frequency. It is “aliasing” as lower frequency energy. And you cannot fix it once it has happened.
Sampling high frequency signal

\[ f(x) \quad \longleftrightarrow \quad F(u) \]

\[ f(x) * h(x) \quad \longleftrightarrow \quad \left[ f(x) * h(x) \right] \text{comb}_M(x) \]

Anti-aliasing filter
Sampling high frequency signal

- Without anti-aliasing filter:

\[ f(x)\comb_M(x) \]

- With anti-aliasing filter:

\[ [f(x) * h(x)]\comb_M(x) \]
Aliasing in Images
Image half-sizing

This image is too big to fit on the screen. How can we reduce it?

How to generate a half-sized version?
Image sub-sampling

Throw away every other row and column to create a 1/2 size image - called *image sub-sampling*
Image sub-sampling

1/2

1/4 (2x zoom)

1/8 (4x zoom)

Aliasing! What do we do?
Gaussian (lowpass) pre-filtering

Solution: filter the image, \textit{then} subsample

- Filter size should double for each \( \frac{1}{2} \) size reduction. Why?
Subsampling with Gaussian pre-filtering

Gaussian 1/2  
G 1/4  
G 1/8
Compare with...

1/2

1/4 (2x zoom)

1/8 (4x zoom)
Campbell-Robson contrast sensitivity curve

The higher the frequency the less sensitive human visual system is...
Lossy Image Compression (JPEG)

Block-based Discrete Cosine Transform (DCT) on 8x8
Using DCT in JPEG

- The first coefficient $B(0,0)$ is the DC component, the average intensity
- The top-left coeffs represent low frequencies, the bottom right – high frequencies
Image compression using DCT

- DCT enables image compression by concentrating most image information in the low frequencies
- Lose unimportant image info (high frequencies) by cutting $B(u,v)$ at bottom right
- The decoder computes the inverse DCT – IDCT

- Quantization Table

<table>
<thead>
<tr>
<th>3</th>
<th>5</th>
<th>7</th>
<th>9</th>
<th>11</th>
<th>13</th>
<th>15</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>13</td>
<td>15</td>
<td>17</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>11</td>
<td>13</td>
<td>15</td>
<td>17</td>
<td>19</td>
<td>21</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>13</td>
<td>15</td>
<td>17</td>
<td>19</td>
<td>21</td>
<td>23</td>
</tr>
<tr>
<td>11</td>
<td>13</td>
<td>15</td>
<td>17</td>
<td>19</td>
<td>21</td>
<td>23</td>
<td>25</td>
</tr>
<tr>
<td>13</td>
<td>15</td>
<td>17</td>
<td>19</td>
<td>21</td>
<td>23</td>
<td>25</td>
<td>27</td>
</tr>
<tr>
<td>15</td>
<td>17</td>
<td>19</td>
<td>21</td>
<td>23</td>
<td>25</td>
<td>27</td>
<td>29</td>
</tr>
<tr>
<td>17</td>
<td>19</td>
<td>21</td>
<td>23</td>
<td>25</td>
<td>27</td>
<td>29</td>
<td>31</td>
</tr>
</tbody>
</table>
JPEG compression comparison

89k

12k
Maybe the end?
Or not!!!

• A teaser on pyramids…
Image Pyramids

Idea: Represent NxN image as a “pyramid” of 1x1, 2x2, 4x4, …, $2^k \times 2^k$ images (assuming $N=2^k$)

Known as a **Gaussian Pyramid** [Burt and Adelson, 1983]
- In computer graphics, a *mip map* [Williams, 1983]
- A precursor to *wavelet transform*
Band-pass filtering

Gaussian Pyramid (low-pass images)

These are “bandpass” images (almost).
Laplacian Pyramid

Original image

• How can we reconstruct (collapse) this pyramid into the original image?
Computing the Laplacian Pyramid

\[ I = G_0 \]

\[ L_0 \]

\[ G_0 = I \]

\[ g \]

\[ G_1 \]

\[ L_1 \]

\[ G_1 \]

\[ g \]

\[ G_2 \]

\[ G_2 \]

Reduce

Expand

Need \( G_k \) to reconstruct

Don’t worry about these details – YET! (PS4?)
What can you do with band limited imaged?
Apples and Oranges in bandpass

Fine \( L_0 \) (a)

L2 (d)

Coarse \( L_4 \) (g)

Reconstructed (j)
What can you do with band limited images?